

MENTOR GRAPHICS CORPORATION

MECOP WRITTEN REPORT

FIRST INTERNSHIP

September 4, 2012

Intern

Nicholas Kelly

Oregon State University

Computer Engineering

Mentor

George Lippincott



Table of Contents

| | | |
|----------|--|-----------|
| 1 | Internship | 3 |
| 1.1 | Company | 3 |
| 1.2 | Department and Role | 3 |
| 2 | List of Projects | 3 |
| 3 | Executive Summary | 4 |
| 3.1 | Major Projects and Results | 4 |
| 3.2 | Specific Accomplishments | 5 |
| 4 | Projects | 6 |
| 4.1 | MTFlex GRID Monitor/gMultiflex | 6 |
| 4.2 | DevTerra/Terreno | 8 |
| 5 | Conclusion | 9 |
| 6 | Buzz Word List | 10 |

1 Internship

1.1 Company

Mentor Graphics is a publicly held company in the Electronic Design Automation (EDA) field. Their headquarters is in Wilsonville, Oregon, and they have other locations all throughout the world. The company was formed in 1981 by Tom Bruggere, Gerry Langelier, and Dave Moffenbeier (a group originally from Tektronix) in Beaverton, Oregon. They began development on computer aided engineering software (for electronics) which would run on Apollo hardware. Unlike some of their initial competitors, Mentor Graphics designed their software from scratch, closely listening to companies that would ultimately use the products. After only a couple years, Mentor Graphics began international operation why adding offices in Europe and Asia.

Currently Walden C. Rhines serves as the CEO and chairman of the board, and Gregory K. Hinckley serves as the President of Mentor Graphics and CFO. There about 4,400 employees located at over 70 offices worldwide, notably in France, Egypt, Russia, and Ireland. Last year, Mentor Graphics surpassed 1 billion dollars in revenue. Cadence Design Systems and Synopsys remain Mentor Graphics largest competitors, each share about a quarter of the EDA market.[1]

1.2 Department and Role

I worked in the Resolution Enhancement Technology (RET) division of Calibre, Mentor Graphics' product for IC Design/Verification/Manufacture. RET is comprised of many different techniques for increasing the resolution of a design on silicon. Since designs are becoming increasingly dense, there are issues that occur when transferring the mask (design) to the silicon. Using RET, these issues can be avoided which will cause the actual semiconductor to more closely match the design. My supervisor, Pradiptya Ghosh (located in Fremont, CA) is the Engineering Director for Semiconductor Manufacturing. My mentor, George Lippincott, is a Project Lead in Correction and SRAF (Mask Synthesis).

I was part of the Research and Development (R&D) part of RET, but I also worked on projects for Quality Assurance (QA). My role in the company involved developing various tools use mainly for testing the product (Calibre), visualizing test results, and displaying other information (usually using webpages or emails). I mainly worked with employees at the Wilsonville campus, but I also worked on some projects for the Fremont, CA office.

2 List of Projects

The list below is a comprehensive list of all the projects that I worked on throughout the internship.

- Defect Report Plot
- Workbench Polygon-Text
- Dev CFS
- ClearQuest Query
- Gather group files
- Devterra
- Terreno
- Testcase include search
- Coverity Web Query
- Status Reports
- User Status Reports
- Purify Reports
- Quality Initiative Emails
- gMultiflex
- MTFlex GRID Monitor
- Transcript/Log Archiving
- Forensic Regression Test

3 Executive Summary

3.1 Major Projects and Results

DR Plots/ClearQuest/Coverity/Dev CFS Query

I worked on various webpages which would display data from DR Plots (defect reports), ClearQuest (issue reporting), Coverity (error checking), and Dev CFS (test results). For Coverity and Dev CFS I implemented a caching system to improve performance. I added new features and improvements to all of the scripts including a commenting system, interactive graphs, CSV file output, and more. These webpages made it easier to gather information, visualize data, and review work done.

WORKbench Polygon-Text

The goal of this project was to create text in the Calibre WORKBench in polygons using only 90° and 45° angles. I created a system which would draw lines using polygons. Each letter would have various “move-to” or “line-to” commands which would then draw the letter/character. This system allowed for full paragraphs to be drawn in various sizes. This project added an extra way of debugging Calibre through drawing text in a design.

Devterra/Terreno

I worked on Devterra and Terreno which are both used to submit tests to various computers. Both were Graphical user interfaces (GUIs) which were used to setup a group of tests for submittal to an array of computers on the network. I also added a feature to copy files from Fremont, CA to Wilsonville into to speed up tests. These additions made tests faster for employees off the Wilsonville campus, and made tests easier to submit in general.

Purify Testing

There was a need for nightly Purify testing/debugging, with result pages. I created a system

which would run Purify tests, and then make HTML reports for groups of tests (extracting Purify errors).

Quality Initiative Emails

The RET QA department was working on a Quality Initiative where certain pieces of information needs to be filled out for issues. My project was to create a system which would look at defect reports that met a criteria and email the owners if not enough information was filled out. Managers would also be emailed with a summary of different statistics. The script saved managers the need to remind employees to fill out information and provided an insight into a group of issues on a weekly basis.

gMultiflex/MTFlex GRID Monitor

There was a need to easily add extra “worker” processes to a Calibre run for jobs which may be taking too long. I created a program which could add/remove “remotes” to/from a Calibre run. I also created a GUI which simplified adding and removing, as well as providing other statistics. For Calibre runs that would take too long to run, this script added an easy way to add remotes to speed up the task. It also added a way to easily visualize how multiple hosts were performing.

Transcript/Log Archiving

In order to track test statistics, there was a need to store the logs from each test (run nightly). My project was to create a system which gathered and compressed the logs for each (daily) test run. Certain values could be extracted from these archive and stored in another archive used by a webpage to view results and trends. These logs would provide a new archive which would allow developers to see how certain values have changed over a long interval.

Forensic Regression Test

In order to get more insight into problems, multiple tests would often be run on failed testcases. To simplify this process, I created a custom regression test which combined a few tests into one and integrated into existing programs. This script saved employees the need to manually re-run certain tests.

3.2 Specific Accomplishments

- Learned how to use Perl and Tcl.
- Thought of a unique way to store/draw polygon characters by using a grid of points along with “move-to” and “line-to” commands.
- QI EMailer started sending emails.
- Created my first Tcl tk GUI with Terreno.

4 Projects

Below are a couple of major projects that I worked on for a significant portion of my internship and that I believe describe my internship well.

4.1 MTFlex GRID Monitor/gMultiflex

On the Wilsonville campus, nearly all Calibre testcases are run through a network of computers called GRID (using the Oracle GRID system). Some testcases are run in MTFlex or Hyperflex mode where there is one (MTFlex) or more (Hyperflex) controllers, and multiple remotes or “worker” processes. These remotes can be on any host in GRID, though they are usually assigned in groups of 4 (Quad-core CPUs). GRID has a queue system where employees submit their jobs and they are served based on the load and priority. Thus, in some cases it can be difficult to get a job on GRID sometimes. For testcases that require a high amount of remotes, it can be increasingly difficult since the amount of remotes requested has to be available at that time. There was one existing solution: CalCM (Calibre Cluster Manager), which can dynamically add remotes to a Calibre job depending on the job’s demand. However, CalCM was separate from GRID and had a smaller dedicated cluster. The idea was for me to create a script which could add remotes to an existing Calibre job on GRID.

In the first version of gMultiflex (my script), it could only start Calibre jobs and add remotes to them. In order to get remotes from GRID, I called another script called “gridflexr”, which would wait till it received an amount of remotes, and would then return the hosts. I would simply rsh (remote-shell) a Calibre command to connect to the Calibre job’s remote socket, which was obtained in the log/transcript for the job. In addition to this, CalCM had to be disabled because it would try to remove remotes that were added (if there was a lower demand). Also, in order to guarantee that there were enough Calibre licenses for each of the remotes added, the “MAXCOUNT” of remotes had to be set in the beginning. This was done with a multiplier (with respect to the initial amount of remotes) or a max CPU count.

In the second version of gMultiflex, my mentor wanted the functionality of removing remotes. The only way of (safely) doing this was to connect to the Controller socket for the job and “REMOVE” certain remotes. Thus, the program was re-designed with a CSH script as the main script, a Perl script to connect to the socket, and shell script to launch remotes. Adding remotes now was done by sending “TRIGGER hosts” to the socket, which call the launch script (similar to version 1). Removing remotes was similar, but “REPORT REMOTES” was called to get the most recently added remotes (returned in XML), and remove them. Similarly, each “add” process monitored the remotes added, and quit when the remotes were removed or finished. However, the remove process was not ideal, since “REMOVE” supported only a list of hosts or a number, there would often be one remote (busy) that would be removed. Since the add processes kept all their GRID remotes until they quit, this meant remotes were not being freed to GRID.

In the final version, an enhancement was made to Calibre by an employee so that each “TRIGGER” call formed a group of remotes. Then, these groups could be removed by calling “REMOVE GROUP” with the group ID, which guaranteed removal of the remotes in the group, freeing up resources on GRID. Because of the need to know what group to remove, a GUI was created using Tcl tk which interfaced to gMultiflex and the (Perl) socket script.

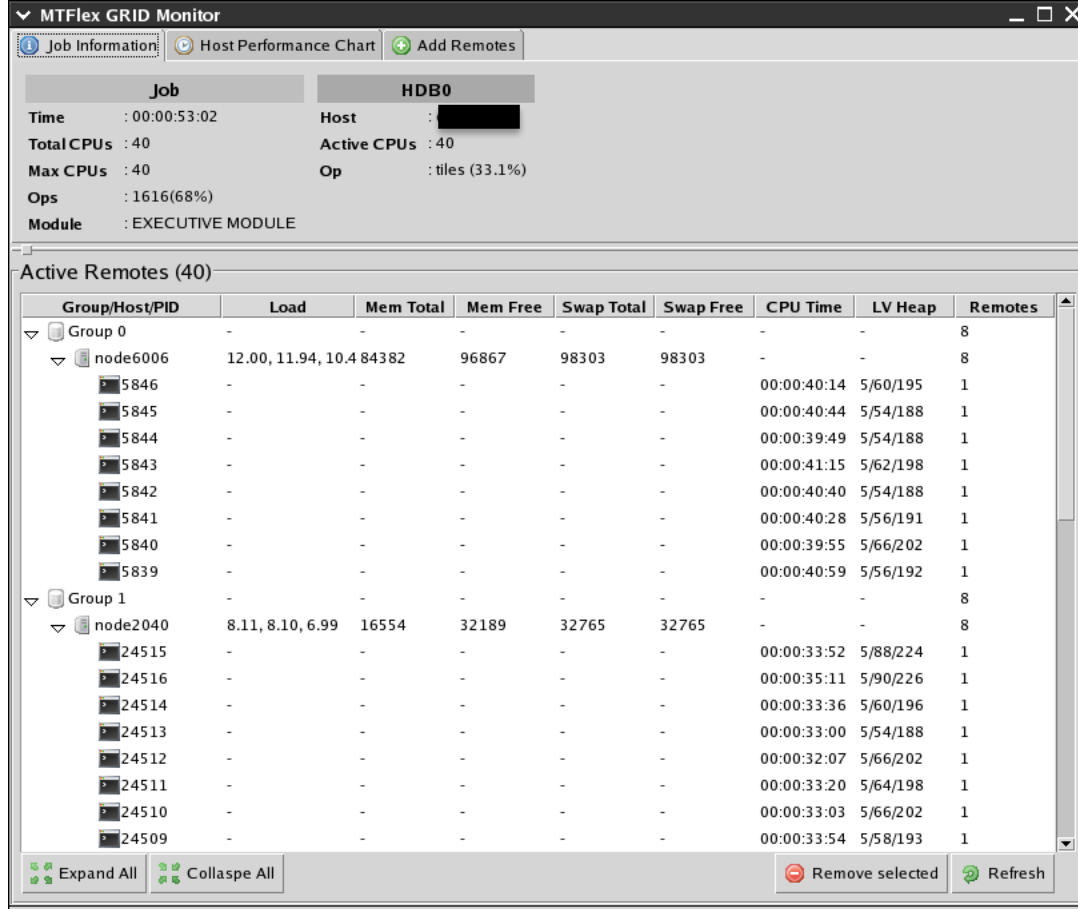


Figure 1: MTFlex GRID Monitor

Every process in the GUI for receiving information was forked from the GUI in order to keep the GUI responsive. The job statistics and performance information was refreshed every 10 seconds, with information about hosts and processes. The socket script parsed the XML for each request/response and put it in a more readable format for Tcl. Groups could now be removed by simply selecting the groups in the treeview and clicking “Remove Selected.” Since the socket was serial, this would delay updating in the GUI for about 5 minutes. Remotes could also be added where the “add status” would be displayed by reading the output of the process via a non-blocking pipe. I also added a performance graph which would display the performance (CPU load, memory usage, swap usage) for each host.

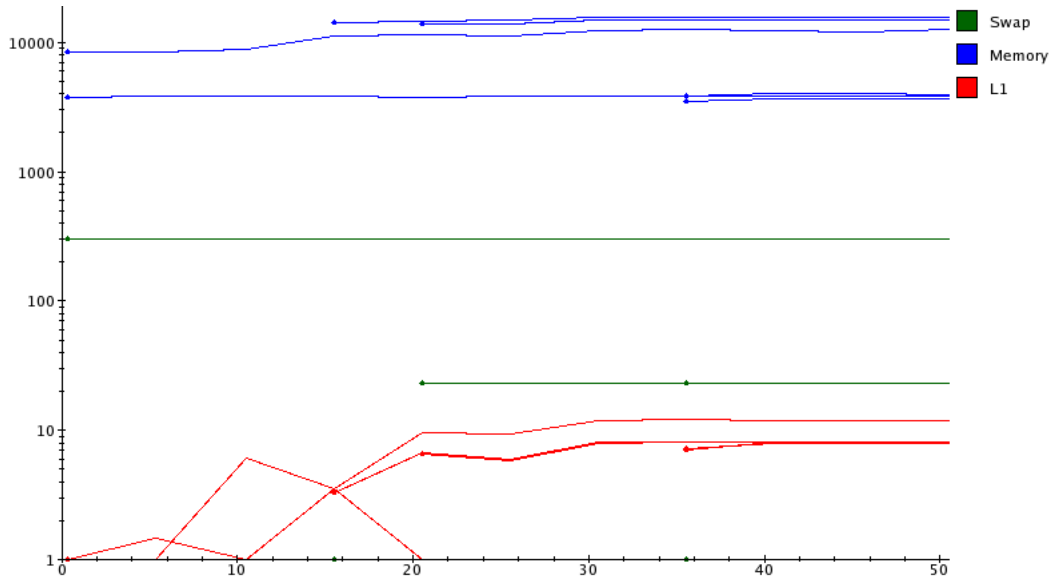


Figure 2: Performance Graph

4.2 DevTerra/Terreno

Devterra is Tck tk GUI which was created by another MECOP intern. It is used to submit testcases to GRID using another tool called Luna. While GRID is Wilsonville campus, some employees use Devterra in Fremont, CA. These testcases can take longer since the DWA (Developer Work Area) and/or IWA (Integration Work Area) is located in Fremont, meaning data has to go over the network. In order to fix this, if a DWA was located in somewhere other than WV (Wilsonville), it would be relocated to an WV directory.

The process of copying the DWA or IWA to a WV directory was done by the “createNanotreeFromDwa” script which was called right before the testcases were submitted to Luna. Based off of the DWA’s path and Results directory, a path would be generated for where the DWA/IWA would be relocated to. “Copy” and “Workbench copy” checkboxes were added to the GUI to indicate what to relocate. If a DWA was already located in WV, it would not be relocated. Also, if there was a WV equivalent IWA, that would be used instead of copying the IWA over as well. Other features such as overwriting the relocated DWA or renaming it were also added.

There was also a need for a simple version of Devterra, which launched testcases in a different way. I created a Tcl tk GUI called Terreno which would create a state file (for Terra) and a script which would modify the state file and launch Terra. The GUI was similar to Devterra, but more information was taken from environment variables, and features were more automatic. For instance, the baseline (IWA) for the DWA selected was automatically found. If it was a non-WV DWA it would automatically suggest relocating it (experimental). Filter files (containing which testcases to run), could also be merged, allowing multiple “products” to be tested at once. Their respective variable files (environment variables) would also be merged, with conflicts being resolved by the user.

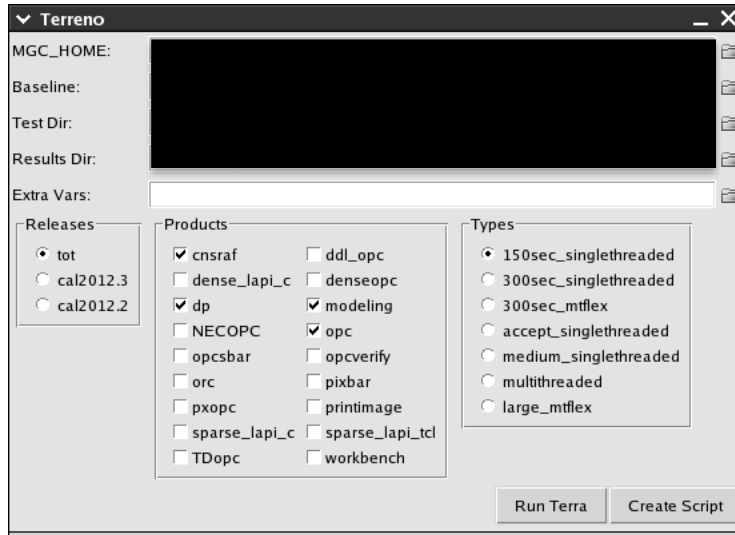


Figure 3: Terreno

5 Conclusion

During my internship, I learned more about scripting languages such as Perl, Tcl, Csh, Bash, and Python. I used Perl and Tcl most often, Perl proved to be very useful for regular expressions and ease of development. Within scripting, I learned more about GUI design, specifically using Tcl/tk. I also learned a bit about C++ programming mostly involving creating classes. Going into Mentor Graphics, I knew little about the EDA industry. Now, I understand a bit more about EDA in general and how important the software they make is. At Mentor Graphics, I also experienced how the software development process works including tasks in the R&D side and QA side. Along with this, I learned more about different testing processes. Mentor Graphics had various types of tests they used, and many different testcases.

At first I was a bit cautious when being accepted to Mentor Graphics. But, just after a week, I could tell that it was going to challenge me, and give me insight into a different industry. Just about everything that I learned at my internship are traits that are required by most employers in EE, CpE, and CS. It was good to get experience in a work environment that will be close to where I would work after graduation. I also met many people at Mentor Graphics who are in the CS/EE industry, whom will be useful once I am looking for a job in a similar or related field. I feel that if I decide to go into CS, Mentor Graphics will definitely have high priority because of its unique mix of CS and EE knowledge.

The Mentor Graphics Calibre RET team mostly benefited from my work. I worked on pages which made information (DRs, test results, etc) easier to obtain and use. It could be useful for meetings, reports, or to just check progress. I also worked on scripts to help with testing (QA), which would minimize the amount of manual labor done. Some of the tests would be automatic, with online reports, so that developers can see how their product(s) are changing each week/day. Along with this, I made it easier for employees to be reminded about completing information for

certain DRs.

6 Buzz Word List

| | |
|------------|--|
| MGC | Mentor Graphics Corporation |
| EDA | Electronic Design Automation |
| D2S | Design to Silicon |
| RET | Resolution Enhancement Technology |
| OPC | Optical Proximity Correction |
| QA | Quality Assurance |
| FTO | Flexible Time Off |
| WFH | Working From Home |
| DR | Defect Report |
| DWA | Developer Work Area |
| IWA | Integration Work Area |
| VCO | Vendor, CPU architecture, Operating System |

References

- [1] Mentor Graphics Corporation History, Various pages on MGC's internal website