GraceFall: Visualizer for Diverse Stress Test Degradation Data Spanning Multiple Time Scales

Matthew Tang and Ian Hill

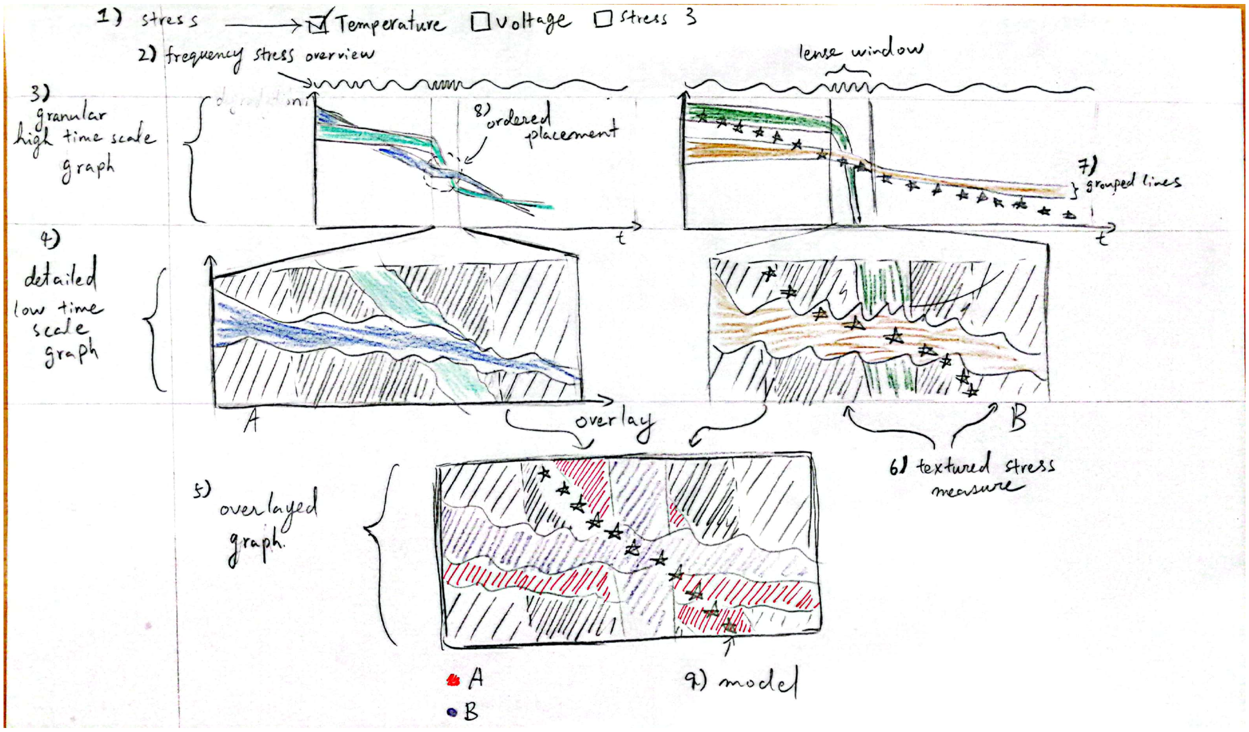

Fig. 1. Moc of proposed visualization. **1)** The stress induced per test. **2)** Frequency overview of the selected stress. **3)** The data in granular time scale. **4)** data in fine time scale. **5)** overlayed graph; this one maybe inaccurate. **6)** stress measure texture. **7)** grouped components. **8)** ordered line placement by arc length. **9)** Estimated model represented by stars.

Index Terms—Multiple time scale, many series summary, integrated circuit reliability.

1 Introduction

Long-term reliability is a niche but critical aspect of modern electronics design. Testing and planning for degradation and failures is incredibly challenging; nearly all product development cycles are much shorter than the required lifespans and so accelerated stress testing followed by extrapolation/prediction is leveraged in lieu of direct testing. These testing practices are the core focus of one author's (Ian's) doctoral thesis topic, and several shortcomings of existing reliability engineering practices were identified in a survey paper published in IEEE Transactions on Device and Materials Reliability [2]. A simulator has already been developed that enables stochastic temporal simulation of wear-out processes and Bayesian inference on wear-out models to address two of these shortcomings.

Due to the model-agnostic implementation and probabilistic focus of the developed simulator, visualizing the simulated stress, measurement data, and specified stochastic models in a way that supports the variety of user tasks enabled by the simulator presents a significant challenge. This project develops a visualization tool to aid reliability engineers carry out multiple user tasks that can be tackled using the simulator in an effort to overcome the conceptual challenges of non-constant stress and probabilistic models that engineers are not likely to be accustomed to. Key tasks include iterative stress test design, test data failure analysis, and model quality comparisons.

Our preliminary investigation found that in all three use cases engineers will need to compare large quantities of scalar data series under multiple stress configurations through time. The proposed solution, GraceFall, handles diverse line graphs with dynamic time scales to aid in these exploratory analysis tasks. The proposed visualization will integrate with the existing simulator and thus be primarily developed with Python.

2 Related Work

There are two types of related works to discuss in order to place this project in context, those within the integrated circuit wear-out reliability space, and then those within the field of info visualization.

2.1 Integrated Circuit Reliability Visualizations

Integrated circuit reliability is not well known for effective visualization tools, as the field is reasonably niche and driven by individual companies within the semiconductor industries. Reliability reporting is mostly opaque to end users of integrated circuit products, thus visual representations of reliability and reliability tests are often neglected. Typically, the visualizations that do get constructed are of poor quality. Common visual representations of reliability test data include Weibull plots of cumulative failures against time, or box plots showing degradation distributions for different devices. Some example visualizations published by one of the most recognizable researchers in the field are shown in Figure 2, highlighting the prevalence of poor visual design choices [4]. Some additional examples can be found in [5, 8]; most within the field follow very similar best practices, or lack thereof, when designing visual representations of their data. For Ian's research, effective visual representations are needed to eventually help engineers explore test design spaces and compare different probabilistic models, thus the current standard visualization practices in IC reliability are insufficient.

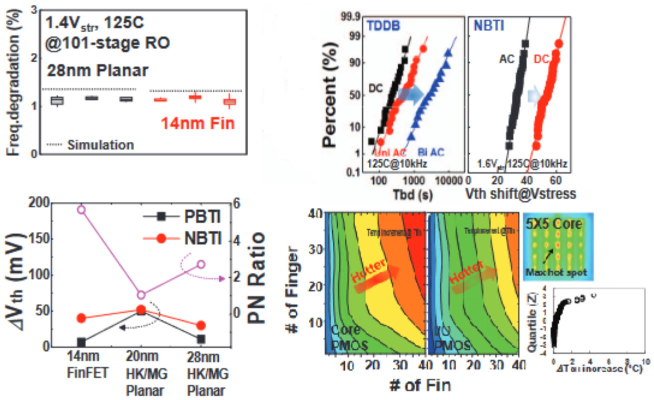

Fig. 2. Examples of accepted visualizations within the IC reliability research community from a well-regarded paper [4]

2.2 Relevant Information Visualization Design Studies and Techniques

Moving on to how this project fits within data visualization research, we are primarily focused on the application of visualization techniques as opposed to the development of novel ones. With this in mind, our project is suitably described as a design study, and can be positioned relative to existing similar design projects of larger scope, namely KD-Box [10] and Vismon [1]. KD-box is useful as a comparative study because it addresses similar problems of visualizing time-series data from sensors where many independent data series need to be summarized or reduced to extract useful information about the collective, and while Vismon does not consider time series plots, its treatment of model comparison across multiple views and incorporation of information uncertainty is comparable to the design components encountered further into the proposed project. Additionally, specific design challenges within our project have been explored by previous works. Although our project allocates significant time for more detailed literature review, we have already looked at some initial solutions for overcoming the following anticipated design challenges:

1. Multiple time scale plots introduce challenges when visualization users are interested in both long-term summaries of temporal data and in the more detailed structure of the same data when considered over a short interval. Techniques for displaying these types of data series at multiple timescales have been addressed in a paper on "Chronolensing" [9], providing a starting point for designing a tool that can support analysis at these multiple time scales.

2. When many individual data series are plotted in a single view, visualizations can quickly become cluttered and difficult to analyze for overall trends. An initial solution being considered is braiding/weaving [6], a highly adaptable and appealing technique that should be well suited for the project problem. The main challenge with braiding is graphical complexity due to careful use of shadowing and obscuring that may complicate implementation efforts.

3. Users of the visualization are likely to want to compare different tests or data sets in relation to one another, introducing a significant challenge to effectively lay out multiple test plots within a single view frame and navigate them. Vismon [1] can inform some initial design decisions for this project aspect.

4. Visualizing mathematical models is an interesting challenge as they don't fit as cleanly within standard visualization conceptual frameworks. The data object being shown is the output of an expression that is variable, making the data dynamic when dependent on other visualized data. A 2008 paper on plotting models based on parameter selections is considered as a useful starting point for this component of the project [3].

3 Data and Task Abstraction

A core problem reliability engineers encounter in analyzing a set of test result is to understand component degradation with respect to induced stresses. For clarity, an engineer may seek to investigate which stress is causing the most degradation. In the current paradigm, engineers frequently need to compare multiple test results in the form of pure numerical tables, a notoriously difficult data form to analyze from. As visualization are lacking, engineers frequently resort to using component conditions only at the beginning and end of the test, ignoring the information rich degradation process in between. We introduce GraceFall to aid engineer to exploit those rich temporal information in their task. This section will first break down the goal into concrete tasks, identify the data abstraction and propose our solution.

3.1 Data Set

Prior to dissecting the overarching visualization objective, it is necessary to first present the specific data available. There are three major data sets relevant to our visualization: **1)** table data listing the measurements conducted during the test, **2)** table data summarizing the conducted test in terms of applied stresses and durations, and **3)** hypothesized mathematical models that can be fit to the measurement data set. The scalar measured degradation data tables can be abstracted to 3 attributes time, component ID, and measure, which they are quantitative, categorical, and quantitative data types respectively. The "measure" attribute refers to the performance measure used to quantify the amount of degradation. Nuances of the data include the multi-field measurement IDs (i.e. the component ID can be broken down into multiple non-unique fields with only the particular combination being unique), and the potential for different quantities being measured, resulting in multiple categories within the measured values. Broadly speaking, the stress data set is also represented with three attributes: stress phase ID, stress measures, and time, with attribute types of categorical, quantitative, and quantitative, respectively. For clarity, an example of a stress is : (Burn In Stress, 60°, 13 hours). Finally, the hypothesis models are mathematical expressions attempting to explain the test results as fitted functions of the stress data. To keep the project in a manageable scope, this project will restrict the data to deterministic models as opposed to the eventual desire to additionally support probabilistic ones. As this data type does not fit into the framework introduced in the current course, this will be simply represented as sets of quantitative points.

3.2 Task Specification

From analysis, the overarching goal of exploring multiple test data sets can be broken down into three tasks. **T1** - Visually inspect the plotted line graphs to understand the effects of different stressors over different time resolutions. **T2** - Comparing degradation trends across many line graphs and the collective trends. **T3** - Proposing different hypothesis models and comparing model fits.

Early mock-ups identified several design challenges that will form the key feature requirements of GraceFall. **R1** - Handling of clutter

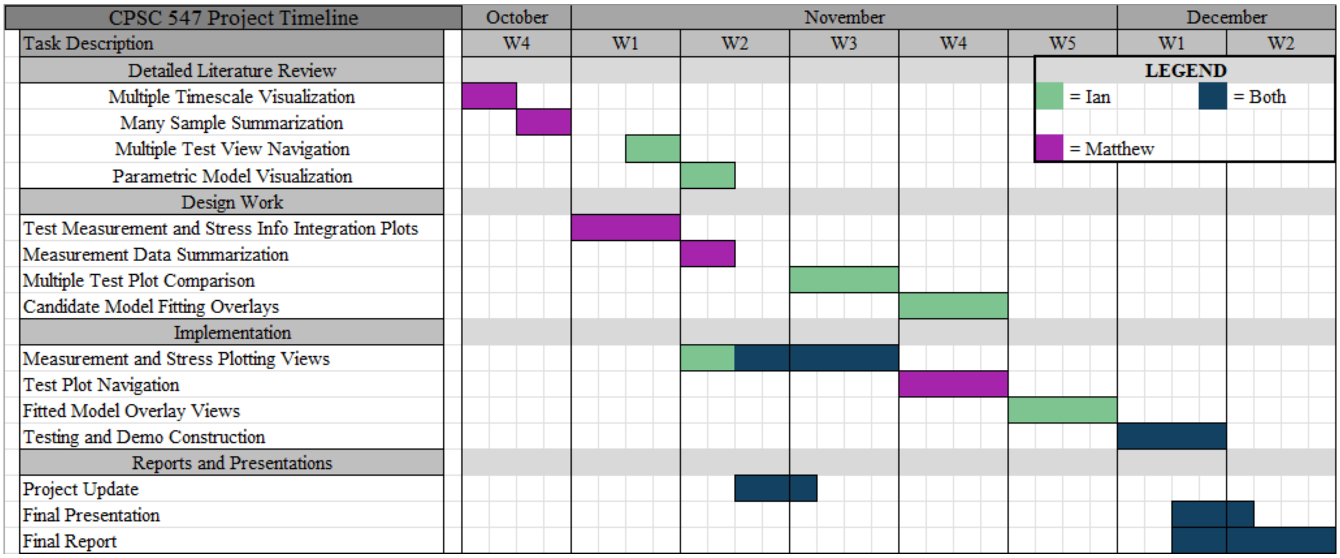

Fig. 3. Project work breakdown by task type and design component. Time shown in approximate dates based on expected weekly work cycle.

caused by many lines per graph due to large quantities of data series on multiple time scales of interest. **R2** Easy comparison of stress parameters within and across test results. **R3** - Comparison between multiple graphs using different test specifications or models. **R4** - Observing and comparing model data fits. The proposed visualization will satisfy each of the identified requirements in aiding the engineer to complete the aforementioned tasks.

4 Proposed Solution

The preliminary GraceFall design will fulfill **R1** using a combination of filtering, sequential line placement and gradient clustering to reduce the line cluttering. Inspired the foot step of Chronolense [9], lensing will be used to enable engineers to navigate easily across the different time scales. To achieve easy comparison described in **R2**, the texture channel is reserved to represent the stress of interest, and other potential channels can be considered for displaying additional stress parameters. Fulfilling **R3**, the Chronolense design patterns will be adapted to inform views of multiple test plots simultaneously, allowing for easy comparison. For **R4**, the model will initially be represented as a simple single dotted line as others channels are populated. To emphasize that the line is a model, the dots will be replaced with an alternative symbol, such as stars.

R1 The number of data series that must be represented within a single plot is potentially unbounded. Inspired by Line Weaver [6], the rendering process of each line will be sorted by arc length, with the longest in the back and shortest in the front, reducing overall occlusion as a result. Further, series from the same device groups will use similar colours to direct the engineer to components of interest. Based on this, engineers can filter groups of interest by decreasing the transparency of groups requiring further analysis. To achieve multi-time-scale data comparison, TVR uses lensing and overlays inspired by ChronoLenses [9] and Timenotes [7]. Lensing will allow the engineer to zoom in any time window desired to view short-term data series behaviours.

R2 The texture channel is reserved for communicating the stress measures in finer time scale graphs. Specifically, based on the value of the stress measure, the density of the texture will change accordingly. The texture will become the background of the period of time when it is applied, enabling the engineer to tell when the stress is applied. Large and differing time scales will clutter the texture channel, destroying the capacity to communicate measurement information. As an alternative, a sinusoidal function of diverse frequency will be put above the graph to inform the applied stress' effective area with the higher frequency corresponding to higher stress value. These frequency graphs will serve as overview to what is happening in the underlying granular time scale.

R3 For multiple test comparisons, the visualization user can select any two windows and overlay the plots on top of each other to observe if there is any pattern. By this construction, any number of tests can be overlayed on top of each other where their respective lines have various degree of transparency, enabling detailed test comparison.

R4 The design for this requirement will be further developed further into the project timeline. As a rough plan, the model will be represented with a simple dotted line where the dots are replaced with stars for emphasis. This channel is chosen to be orthogonal to other channels occupied thus far.

5 Milestones

Our visualization can conveniently be divided into a few conceptual components that are reasonably separate in terms of design, easing the process of breaking down the project into tasks. In terms of milestones, we define three internally: **1)** Literature Review Completed, **2)** Design Plan Work Completed, and **3)** Implementation and Testing Completed. Reports are not included in these internal milestone descriptions. A summary of the project execution plan, vertically divided by milestone and course deliverable tasks (tasks defined by feature requirements), is provided as a Gantt chart in Figure 3.

Some key elements within the provided Gantt chart are the milestone and deliverable deadlines and dependency ordering of literature review to design plan down to implementation for the different design components. The project update deadline is on November 15^{th}, or just into week 3 of November, the project presentation is December 14^{th}, or middle of week 2, and the final report is due on December 16^{th}, the end of that same week. Based on the work breakdown within the chart, each filled cell can be mapped to approximately 4 hours of expected work to obtain an estimate of project effort that slightly overshoots the course recommendation for total project work hours. The group has no issue with this light increase in workload. The Milestone 1 deadline is set for middle of week 2 in November (32 hours of expected work), Milestone 2 by end of November week 4 (56 hours), and 3 by end of week 1 December (80 hours).

Some final points to note in the chart: **1)** Ian does not have tasks assigned for the first planned week and a half due to a research paper deadline, **2)** some existing basic plotting functions were already developed for the simulator that the visualization tool is built on top of and thus the first portion of implementation is dedicated to having Ian translate these functions into the selected library to provide a basic foundation to build on.

Due to the large planned scope of the project, there is a significant risk that the available time will be insufficient to design and implement all desired features. To mitigate this risk the model fitting design component has been designated as lower importance; if necessary this aspect of the project can be cut. Doing so would save 40 hours of allocated time, based on the chart timeline breakdown, that can then be reallocated to the test data integration plots and test plot navigation portions of the project as needed.

6 CONCLUSIONS

The GraceFall tool will tackle a challenging visualization problem, cleanly displaying highly varied scalar data series to avoid the large potential for visual clutter and obscured properties of interest. If successful, the tool will greatly aid engineers in developing accelerated reliability stress test procedures and tuning models for explaining physical degradation phenomenon. The large scope of this project necessitates a clear execution plan which has been developed to maximize the potential of constructing an effective solution.

REFERENCES

[1] M. Booshehrian, T. Möller, R. M. Peterman, and T. Munzner. Vismon: Facilitating analysis of trade-offs, uncertainty, and sensitivity in fisheries management decision making. *Computer Graphics Forum*, 31(3pt3):1235–1244, 2012. doi: 10.1111/j.1467-8659.2012.03116.x

[2] I. Hill, P. Chanawala, R. Singh, S. A. Sheikholeslam, and A. Ivanov. Cmos reliability from past to future: A survey of requirements, trends, and prediction methods. *IEEE Transactions on Device and Materials Reliability*, 22(1):1–18, 2022. doi: 10.1109/TDMR.2021.3131345

[3] K. Matkovic, D. Gracanin, M. Jelovic, and H. Hauser. Interactive visual steering - rapid visual prototyping of a common rail injection system. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1699–1706, 2008. doi: 10.1109/TVCG.2008.145

[4] S. W. Pae, H. C. Sagong, C. Liu, M. J. Jin, Y. H. Kim, S. J. Choo, J. J. Kim, H. J. Kim, S. Y. Yoon, H. W. Nam, H. W. Shim, S. M. Park, J. K. Park, S. C. Shin, and J. W. Park. Considering physical mechanisms and geometry dependencies in 14nm finfet circuit aging and product validations. In *2015 IEEE International Electron Devices Meeting (IEDM)*, pp. 20.6.1–20.6.4, 2015. doi: 10.1109/IEDM.2015.7409744

[5] G. Park, H. Yu, M. Kim, and C. H. Kim. An All BTI (N-PBTI, N-NBTI, P-PBTI, P-NBTI) Odometer based on a Dual Power Rail Ring Oscillator Array. In *2021 IEEE International Reliability Physics Symposium (IRPS)*, pp. 1–5. IEEE, Monterey, CA, USA, Mar. 2021. doi: 10.1109/IRPS46558.2021.9405181

[6] T. Trautner and S. Bruckner. Line weaver: Importance-driven order enhanced rendering of dense line charts. In *Computer Graphics Forum*, vol. 40-3, pp. 399–410. Wiley Online Library, 2021.

[7] J. Walker, R. Borgo, and M. W. Jones. Timenotes: a study on effective chart visualization and interaction techniques for time-series data. *IEEE transactions on visualization and computer graphics*, 22(1):549–558, 2015.

[8] C. Yang and F. Feng. Multi-Step-Ahead Prediction for a CMOS Low Noise Amplifier Aging Due to NBTI and HCI Using Neural Networks. *J Electron Test*, 35(6):797–808, Dec. 2019. doi: 10.1007/s10836-019-05843-7

[9] J. Zhao, F. Chevalier, E. Pietriga, and R. Balakrishnan. Exploratory analysis of time-series with chronolenses. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2422–2431, 2011.

[10] Y. Zhao, Y. Wang, J. Zhang, C.-W. Fu, M. Xu, and D. Moritz. Kd-box: Line-segment-based kd-tree for interactive exploration of large-scale time-series data. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):890–900, 2022. doi: 10.1109/TVCG.2021.3114865