DEVELOPMENT OF PARALLEL TDDB PROGRAM AND MAXIMUM LIKELY HOOD METHOD SCRIPT

NOBLE PAUL KANJOOKARAN

STUDENT NUMBER: 00040491

COMPANY SUPERVISOR: MR. LEO VAN MARWIJK

SCHOOL SUPERVISOR: MR. D O HEINEKE

HOGESCHOOL ZEELAND

2006/07

Created by Noble Paul Kanjookaran
DEVELOPMENT OF PARALLEL TDDB PROGRAM AND MAXIMUM LIKELYHOOD METHOD SCRIPT

BY

NOBLE PAUL KANJOOKARAN

STUDENT NUMBER: 00040491

COMPANY SUPERVISOR: MR. LEO VAN MARWIJK

SCHOOL SUPERVISOR: MR. D O HEINEKE

A Student Internship Project report submitted to Hogeschool Zeeland and NXP Semiconductors, in partial fulfillment of the requirements for the Bachelors in Mechatronics

HOGESCHOOL ZEELAND

2006/07

Created by Noble Paul Kanjookaran
ACKNOWLEDGEMENTS

I wish to thank my company project supervisors, Mr. Leo Van Marwijk who was very supportive and gave positive feedback to make this project a big success. His appreciation and great wealth of knowledge certainly made this project a great success. I also wish to thank Mr. D.O Heineke, the school supervisor for his guidance and spontaneous way of thinking. Mr. Ramun Kho was very helpful in giving ideas and explaining various models for calculations. I also wish to thank all the technical support staff and teachers who have been supportive in their guidance, faith and trust in me throughout this learning experience and the time in school before the project as part of preparation for this Internship. I would like to express my deepest gratitude to the Internship co-coordinators (who patiently helped out in all legal aspects of this internship) and gave me an opportunity for this wonderful Internship.
**ABSTRACT**

The students in Hogeschool Zeeland have to do an internship for the last semester of their studies to apply their theoretical knowledge in a practical environment to make best use of their education. I was selected to do an internship in NXP semiconductors (Founded by Philips).

The project involved the application of hardware, software and programming to make the software for various machines and for a script to analyze the data from them. Hence the project involved all the disciplines of Mechatronics.

The project was completed successfully and programs were tried and tested under various conditions. The machines were functioning as expected and the script was giving desired output at the end of internship.
INDEX

Chapter 1 Introduction
  1.1 Company Background 1
  1.2 Rationale of the Project 2
  1.3 Objectives of the Project 3
  1.4 Scope and Limitations of the Project 4

Chapter 2 Theoretical Background 5
  2.1 Hardware 5
    2.1.1 HP 5250 Switching Matrix 6
    2.1.2 Keithley SCS 4200 6
    2.1.3 HP 4083 8
    2.1.4 Alessi Prober 11
    2.1.5 ECX-56 Expansion Module and KL1500 Light Intensity Controller 12
    2.1.6 Probe Cards 15
    2.1.7 Hot Chuck Controller 16
  2.2 Software 18
    2.2.1 KITE and KULT 18
    2.2.2 C++ 19
    2.2.3 Visual Basic Programming Language 20
    2.2.4 Cascade Microtech Galaxy 21

Chapter 3 Project Development 25
  3.1 Research 25
  3.2 Design Considerations 26
    3.2.1 Ease of Use 26
    3.2.2 Functionalities 26
    3.2.3 Future Considerations 27
  3.3 Making the Software 27
    3.3.1 Parallel TDDB Software 27
    3.3.2 Maximum Likelyhood Method Script 29
  3.4 How to Run the Software 31
    3.4.1 Parallel Program for switching Matrix 31
    3.4.2 Program for 4083A 32
    3.4.3 Program for Parallel TDDB Measurement 33
    3.4.4 Maximum Likelyhood Method Script 34

Created by Noble Paul Kanjookaran
Chapter 4 Results and Discussions

4.1 Parallel TDDB Measurement 36
4.2 Maximum Likelihood Method 37
4.3 Lessons learnt and Reflections 38
4.4 Environmental Considerations 39

Chapter 5 Conclusion and Recommendations

5.1 Conclusion: Parallel TDDB 40
  5.1.1 Objectives Met 40
  5.1.2 Objectives Not Met 41

5.1.3 Recommendations for Parallel TDDB Program 41
  5.2 Conclusion: Maximum Likelihood Method Script 41
    5.2.1 Objectives Met 41
    5.2.2 Objectives Not Met 42
    5.2.3 Recommendations for Maximum Likelihood Method 42

References 43
List of Figures 44

Appendix A User guide for the programs 45
Appendix B Detailed Information on Maximum Likelihood Method 55
Appendix C Problems Faced and Solutions Found
CHAPTER 1

INTRODUCTION

1.1 COMPANY BACKGROUND

NXP is a newly independent semiconductor company (founded by Philips) with a fifty-year history of providing engineers and designers with semiconductors and software that deliver better sensory experiences for mobile communications, consumer electronics, security applications, contact less payment and connectivity, and in-car entertainment and networking.

**Multimarket Semiconductors**

NXP has one of the largest portfolios of multimarket semiconductors in the industry, from basic building blocks like timers and amplifiers to sophisticated ICs that improve media processing, wireless connectivity, and broadband communications. These are designed to save space, extend battery life, enable customized solutions tailored to customers' needs, and make it easy to implement last-minute changes.

#1 in PC (A protocol for providing communications links between integrated circuits)
#1 in industrial UARTs (Universal Asynchronous Receiver Transmitter)
#1 in interface products
#1 in RF products for CATV and satellite tuners (Cable Television)
#1 in 2 laptops uses NXP GreenChip power supply controller
#World's first ARM-based 32-bit microcontroller with 0.18- and 0.14-μm embedded Flash memory
1.2 RATIONALE FOR THE PROJECT

The Student Internship program is conducted by Hogeschool Zeeland to give students a familiarity of working environment and for the companies to select potential candidates. It last around 5 months and can be in a company that is related to the course of study. This gives the students practical applications of what they studied and how to use it.

I was assigned to NXP semiconductors. My job was to make soft wares and test them for two main parts. Earlier, the measurements of TDDB(Time dependant Di-electric Break down) and EM(Electro Migration) were done with one per SMU(Source Monitor Unit ) with a maximum of 4 SMU’s in parallel. This happened since the old software was written in Delphi and so could not be ported to Windows environment in a working way. Some of the softwares were written in HP BASIC(Beginners’ All-purpose Symbolic Instruction Code) language, which was too long difficult to trouble shoot and with almost no possibility to be integrated to current modern environment. My job was to make a new program in C++ which could do the same job as it was done by the earlier programs for the current environment.

A method called OLS (Ordinary Least Squares) method was used in life time predictions. It was rendered cumbersome and is a less reliable engineering method for lifetime calculations. My job was to make a new script which can make this much more reliable and easier. I had to introduce a new method called Maximum Likelihood Method. It is based on statistical analysis and hence provides a better accuracy for a lifetime predictions. This was to be made in VBA (Visual Basic).
1.3 OBJECTIVES OF THE PROJECT

The aim of the project is to make a script that can give more accurate lifetime predictions in visual basic and software in C++ which makes it possible to make measurements in parallel with a HP5250 switching matrix. This will greatly speed up the measurements and hence makes an efficient use of the machines.

The parallel connection program was to make connections for up to 8 pins per SMU. The specific objectives for this software including parallel TDDB measurement were to:

- Develop a reliable program.
- Possibility to connect up to 8 pins per SMU for parallel measurement.
- Ensure that the timings for the breakdown is accurate.
- Ensure that the program gets suitable inputs and its corresponding outputs.
- Design a program that can get easy to use outputs in standard programs.

The Visual Basic script was to be used to estimate the lifetimes of various wafers. For this project, my specific objectives were to:

- Make a script that can get the lifetime better than that currently used script.
- Easy-to-use
- Ensure that the script gives consistent outputs
- Ensure that the scripting is able to accommodate future modifications
- Ensure that the output is easily understandable with graphical results
- Use Maximum Likelihood Method
1.4 SCOPE AND LIMITATIONS OF THE PROJECT

For this project it was necessary to make the software with the objectives mentioned and above. But considering the time that is limited it was decided to make a rough program with minimum functionalities and then to improve on it. It could make sure that in case something happens there was a working program that could be used.

One of the main limitations of the project is that the time is limited. Also since the school is far away the supervisor from the school is unable to monitor the day-to-day proceedings. Since the program is used on three different machines it might be possible that there could be some errors on some of the machines. This could make it harder to troubleshoot the problem and to repair it.

Chapter 2 will be about the background of the project and hence will explain the various hardware and software that were used in the project. Chapter 3 deals with the development of the project and has a general overview of the programs. Chapter 4 will be the result which shows what was achieved during the work placement and the explanation of it in detail. Chapter 5 will be the conclusion of the project and various recommendations that are suggested after the work placement.
CHAPTER 2

THEORETICAL BACKGROUND

2.1 HARDWARE

2.1.1 HP 5250 Switching Matrix

![Figure 2.1.1 HP 5250](image)

The HP5250 is a switching matrix. It can connect various inputs to various outputs depending on the parameters given to the machine\(^2\). It is widely used to make various measurements in parallel. However due to the current constraints of the software it is only possible to connect one pin per SMU.

The machine was designed to give maximum flexibility for the user to change the settings in the middle of the program while running it. This matrix was also programmed from the...
software so multiple points can be measured one after other. The greatest advantage of this machine was that it was completely compatible with their SCS 4200 that was used to control all other equipments. It used a GPIB (General Purpose Interface Bus)\(^3\) interface, which made it possible to connect machines at the same time. The lab was completely networked and all the machines could be controlled from a single machine. All the machines had a different address. So it is possible to control them without any physical contact. This was very much helpful when large measurements were to be done a single site. For example using the switching matrix it is possible to do a measurement on the site with one machine and then to disconnect it at the specified time and to connect a different machine, which could do a different measurement.

2.1.2 KEITHLEY SCS 4200

![Figure 2.1.2(a) SCS 4200](image-url)
Keithley Model 4200 Semiconductor Characterization System (SCS) is an automated instrument designed to provide IV (Current vs Voltage) and It (Current vs Time) characterization of semiconductor devices and test structures. Its advanced digital sweep parameter analyzer combines speed and accuracy for deep sub-micron characterization.

The SMU’s are connected to the switching matrix, which in turn are connected to the pins. This machine also has its own graphical output, which makes it possible to be used as a stand-alone machine. This feature is greatly helpful where there are space constraints. The software called KITE can send the data to various machines and also get their outputs and export it to an Excel file.

The key features are listed below:

- Intuitive, point-and-click Windows®-based environment
- Unique Remote PreAmpps extend the resolution of SMUs to 0.1fA
- New pulse and pulse I-V capabilities for advanced semiconductor testing
- New scope card provides integrated scope and pulse measure functionality
- Self-contained PC provides fast test setup, powerful data analysis, graphing and printing, and on-board mass storage of test results
- Unique browser-style Project Navigator organizes tests by device type, allows access to multiple tests, and provides test sequencing and looping control
- Built-in stress/measure, looping, and data analysis for point-and-click reliability testing.
• Integrated support for a variety of LCR (Impedance Capacitance Resistance) meters, Keithley switch matrix configurations, and both Keithley Series 3400 and Agilent 81110 pulse generators

Figure 2.1.2(b) Close view of SMU in DSCS 4200 showing GPIB connection

2.1.3 HP 4083
Figure 2.1.3(a) HP 4083

This is a simple switching matrix. It only has an on or off connection. The SCS 4200 also controls this machine. The capabilities of this equipment are listed below:

- 8 bit Data Input/Output: Data Input/Output to control peripherals
  - Data Input: 8bit data and 2 bit interrupt signals, TTL level ; outputted via HP-IB
  - Data Output: 8 bit data, relay connection, controllable manually or by HP-IB.
  - I/O connector: 36-pin receptacle. Mating Connector HP P/N:1251-0084

  - Interface Functions: SH1 AH1 T6 L4 SR1 RL1 PP0 DC1 DT0 C0 E1
Remotely Controllable Functions: 16057A Switching, setting of 8 bit data output

Data Output: Status of 8 bit data input.

Remote Program Codes for the 4083A

<table>
<thead>
<tr>
<th>Name</th>
<th>Control</th>
<th>Program Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>CLOSE</td>
<td>C</td>
<td>Example 1: C1 (Close switch 1)</td>
</tr>
<tr>
<td></td>
<td>OPEN</td>
<td>O</td>
<td>Example 2: O1 (Open switch 1)</td>
</tr>
<tr>
<td></td>
<td>CLOSE ALL</td>
<td>CA</td>
<td>Example 3: C123O45 (Closes switch 1, 2, 3 and opens 4, 5)</td>
</tr>
<tr>
<td></td>
<td>OPEN ALL</td>
<td>OA</td>
<td></td>
</tr>
<tr>
<td>Switching Module:</td>
<td>Disable</td>
<td>R0</td>
<td>If SWITCHING MODULE READY is set to Enable, SRQ signal is outputted when the 16057A Switching Module is completely changed.</td>
</tr>
<tr>
<td></td>
<td>Enable</td>
<td>R1</td>
<td></td>
</tr>
<tr>
<td>General Interrupt</td>
<td>Disable</td>
<td>G0</td>
<td>If GENERAL INTERRUPT is set to enable, SRQ signal is outputted when the 4083A receives C1 or C2.</td>
</tr>
<tr>
<td></td>
<td>Enable</td>
<td>G1</td>
<td></td>
</tr>
</tbody>
</table>
2.1.4  Alessi Prober

Figure 2.1.4(a) Alessi Prober in setup
The REL-6100 Semiautomatic Analytical Probe Station features submicron resolution, stability and precise motion control. The REL-6100 combines precision sub-micron probing capability with the superior measurement performance necessary for DC/CV analysis. The stations feature sub-micron resolution, precise motion control, and 8-inch travel. A new semiautomatic, guarded, thermal, triaxial chuck allows probing from 0°C to 200°C with noise levels below 200 fA in a 75-pF capacitive measurement environment. The Windows-based Galaxy software facilitates automatic station control, step and repeat operations, and wafer-map creation. Test results can be overlaid on the map, saved, printed or transferred to file. Manual control is via a joystick or IEEE 488 commands. 

**FEATURES**

Wafer Size : 3" to 8" of Silicon, GaAs, etc.

Travel Resolution : less than 0.1um

Chuck leakage plus noise : less than 20fA at non-thermal by patented FemtoGuarded chuck

Thermal chuck Temperature : -55°C to +200°C by patented MicroChamber

**Software:** Windows 95, Galaxy, Wafer Mapping Control, GP-IB, RS-232C

**Applications**

General electrical DC parametric test

fA level low current measurement

Low temperature Measurement
2.1.5  ECX-56 Expansion Module and KL 1500 Light Intensity Controller

Figure 2.1.5(a)  ECX-56 Expansion module and KL 1500 Light Intensity Controller
The ECX-56 is equipment with various microcontrollers inside which controls the steeper motors which aligns the wafer. This equipment is in turn controlled by the dedicated PC, which sends signals to this equipment deciding on the location to be measured. This is highly precise and is made even more precise by the usage of stepper motors.

**Figure 2.1.5(b)** Various stepper motors used for correct positioning of wafer

KL 1500 light intensity controller is used to control the intensity of the light on the wafer. This is important depending on the types and properties of the wafers. Since this can decide the quality of the video output is highly necessary to be used in conjunction with the dedicated PC.
Figure 2.1.5(c) Highlighting importance of lighting in a setup

2.1.6 Probe cards

Figure 2.1.6 (a) Various probe cards

Probe cards are one of the most important parts in measuring environment. It takes the connection from the output of various HP 5250 controllers and makes it to microscopic
pins for accurate measurement. Different probe cards are used for different wafers and it is required as the wafers have different components in them.

Figure 2.1.6 (b) Probe cards in setup

2.1.7 Hot Chuck Controller

Figure 2.1.7 Hot chuck controller
This is the controller, which controls the temperature at the stressing site of the wafer. It can control the temperature from 0 to 150 degrees. Higher temperature is necessary for stressing and hence obtaining a faster break down time, which gives an idea of normal lifetime. It is highly accurate and maintains the temperature as given to it. The major features are listed below:

**Outstanding Features:**

- Wide Temperature range: 0° to +200°C
- DC Control System minimizes electrical noise
- Temperature Stability :±0.1°C
- Temperature Accuracy: ±0.5°C calibrated against transfer standard
- Advanced ThermoChuck Design:
  - Low stray capacitance and high electrical resistance to ground: Surface Electrical Isolation: >10⁹ ohms at 500 VDC between surface and ground at +25°C; higher isolation configurations are available
  - Superior chuck temperature uniformity: ±0.5°C or ±0.5% of set temperature
  - Guarded (low leakage) chuck configurations are optional

**Standard Features:**

- IEEE-488 or RS232 remote interface
- Proven high reliability
- ThermoChuck interfaces to most major standard and automatic wafer probing stations, laser trimmers and inspection stations
• No Liquid Nitrogen or CO2 is required for cooling.
• HCFC-free and CFC-free
• Certified compliant

Applications:
• Thermal cycling and steady-state thermal testing of wafers, hybrids and other flat devices from 0°C to +200°C
• Moisture-free cold testing of wafers to 0°C with Controlled Environment Enclosure
• Low noise test applications
• High isolation and guarded (low leakage) applications
• Thermal characterization of standard and high power devices

2.2 SOFTWARE

2.2.1 KITE and KULT
KITE is an application program designed and developed specifically for characterizing semiconductor devices and materials. Source and measurement functions for a test are provided by up to eight Source-Measure Units (electronic instruments that source and measure DC voltages and currents). Test capabilities are extended by support of a variety of external components. For less precise measurement and testing applications, other Keithley instruments may be employed to configure the system more closely to the requirement. The major benefits of verifying the design within the computer are speed, cost, and flexibility. The KITE features include more data plotting functions, the ability to append multiple data runs to a given test and plot those data on the same graph, the addition of test condition information to a graph, and easier analysis with new cursor functions. All test data sheets can be saved in comma-delimited ASCII format in addition to spreadsheet format.\(^8\)

Various parameters like the voltage to be stressed; the connections for the SMU etc can be setup from this software. The various options like sweep, fast slow measurement etc can also be chosen from this software.
2.2.2 C++

C++ maintains aspects of the C programming language, yet has features which simplify memory management. Additionally, some of the features of C++ allow low-level access to memory but also contain high level features. So C++ is a modified version of C. It has more functions and is an OOP (Object Oriented Programming Language). This has a lot of advantages. One of the main reasons for popularity for C++ is that it is very modular which means that the program can be divided into many small parts. This is very much useful.

2.2.3 Visual Basic Programming language
Visual Basic for Applications (VBA) is an implementation of Microsoft's Visual Basic, an event driven programming language and associated integrated development environment (IDE) which is built into most Microsoft Office applications. *Visual Basic for Applications (VBA)* is a macro language available in most *Microsoft Office* applications and in other non-Microsoft software (e.g., *ArcInfo*). It can be used to automate most tasks, write scripts and hence is very popular to manipulate data\(^9\).

### 2.2.4 Cascade Microtech Galaxy

---

**Figure 2.2.3** Screen Shot of Visual Basic for Excel
This is an all in one controller which commands the site to be measured and shows the site in the screen.

It has mainly 3 parts as seen below:

**Figure 2.2.4(a)** Cascade Microtech Galaxy

**Figure 2.2.4(b)** Location measured in real time
This shows the current site that is being measured upon. The magnification can be changed to get a closer view or wider view depending upon the situation. There is also a possibility to get the picture of the shown to be saved into the disc.

**Figure 2.2.4(c)**  Wafer Map in Cascade Microtech Galaxy

This part shows the part of wafer that has been measured and that is yet to be measured. The light blue cells are the parts that have already been measured. The brown cells are the locations that are to be measure and the gray cells are the locations that will not be measured. During the setup the locations to be measured can be setup by the program and so it will measure them one after another till they are finished.
**Figure 2.2.4(d)** Controlling Parameters in Cascade Microtech Galaxy

These are the various controllers to control the stepper motor, which in turn controls the site. There are switches to move it to home position, speed of travel and choose if the motion is to be incremented by the distance or by cells.

Overall, this is very handy to get an idea of locations to be measured and to setup areas to be measured automatically. Usually the setup is made to measure the data one after other for relevant locations and left to measure overnight where the machine will automatically guide them to the next location and so on. Since this will be done automatically it saves a lot of time and also can be made to use in weekends where there is no possibility to be changed manually.
CHAPTER 3

PROJECT DEVELOPMENT

3.1 RESEARCH

The research part of the project started with studies of previous project reports. The programs that are written by the previous students were analyzed and studied to get a clear idea of what they had done and what they had not. After that the manuals of the machines were read and programming structure understood. This gave an idea of what is possible and what is not possible. It also provides the structured and the formats in which the program is to be written.

The research was done in two main areas: Hardware and Software. Research on the hardware mainly dealt with the understanding of various machines. Searching the Internet collected the details of most of them. The old project reports where referred for the possible hardware problems. It is necessary to understand how the machines communicate with each other to ensure that the data that is transmitted and received is accurate and no packets are lost in communication. It is also necessary to make sure that the machines must able to receive the commands as they were intended to. Some machines had to be reset before they could accept the data. Also it is necessary to make sure that the communication lines were free at the time of communication. This is especially the case with parallel TDDB measurement. Two machines were to be simultaneously controlled from a single program with the data transmitted in both ways.
Research on software was mainly done on the Internet. Samples of programme were studied that came with the original equipment. This gave an insight to the limitations and capabilities. Since some of the options of the programs could not be accepted by the running machines it is necessary to make a list of program and functions that could be recognized by the machines before a test program could be written. Especially KITE had limitations and restrictions in data conversions and type of arguments that could be passed. This posed serious challenges at times where some alternatives had to be found. It was tiring and time-consuming at times trying to find a work around with the various limitations posed by the software.

3.2 DESIGN CONSIDERATIONS

3.2.1 Ease of Use

One of the major design considerations was to make the program easy to use and understand without any prior training. In future some one who had no familiarity with this set up before might use the program. So considering this possibility the program was to be made easy to use and simple. Most of the work was to be done inside after the necessary inputs were received and errors were to be shown with reason of errors so the user without any other interference could correct them.

3.2.2 Functionalities

The program was to give enough options and choices for the user to make various adaptations or manipulations before running. Almost all the programs were started as a simple running frame without any other functionalities. This was made to make sure that there was a base which was reliable. After the base was tried and tested and the code made to work new functions were added namely for the script. It included a graph,
internal error correction, multiple checking, dynamic values using input boxes etc. From the request of the user many suggestions were implemented. After a working and functional program was ready it was optimized for speed and portability.

3.2.3 Future Considerations

Even though the program was used in the way as it was for a specific machine, this could change in the future. The machines might be different or the data output might be different too. The programs so were made modular so some parts could be retained and others not and could be changed depending on the need.

3.3 Making the software

3.3.1 Parallel TDDB Software

This software is quite complex because it deals with control of two simultaneous equipments. So as a step it was necessary to make software that could connect and disconnect multiple channels. This basic software could also be used in other programs for parallel stressing. The basic program was first conceived and the program that came with Keithley was analyzed. After getting a general impression the parallel program was designed and tested. After the basic parallel program was working it was necessary to add in check and self disconnect capacities based on real time calculations. So getting data back, analyzing them and making decisions based on it became a priority.

It was noticed that there were many conditions that made this hard. There was noise in the system, data could not be interpreted as required and misreading was also seen. After these considerations, new software was made by adding this with previously designed one. This software though buggy gave a general impression of the limitations of this
setup. Two files communicating with each other had to be ruled out, as the KITE architecture couldn’t allow this in a simple way. So a large program that could do the connection and disconnection for reliable measurement was the only solution. After this program was made, it was tested with various wafers to see the errors. There was also the need to add in some error corrections and exit conditions in case there was no connection or lack of contact when the software was running.

Finally after going through the problems mentioned above the new made program was subjected to a new face-lift. Since the existing method was cumbersome many self-checking small parts returning values were added. All SMU’s were made more or less independent of each other. This made the program more modular, easy to modify and robust. The new program was also tested in various conditions and adapted to the needs of the user. Finally after suggestions and ideas from the supervisor a new final program was made which included the self-check, display graph, parallel measurement etc.

A handy tool, which was greatly useful in the time of debugging, was a piece of software called MSGCON (Message Console) from Keithley. This could show the data passed between the GPID devices and test values could be returned to MSGCON making it easier to know the condition of various variables, which reflected the internal condition of the machines. For example this made it easier to find if the machine had received the commands in the way it was send and also gave a low level communication display which made error finding much more efficient.
3.3.2 Maximum Likelyhood Method Script

This was a completely new project and method. So there was no reference that could be relied upon. The earlier methods were studied and attention was also given to the necessity of the tests and the results that were given by them. It was noted that the method that was currently used was not very efficient and the chances to make mistakes were quite high. The new method was expected to make the probability of errors much lower and in much. In accelerated life tests, devices are exposed to stressed conditions (e.g. higher current and higher temperature regarding Electro Migration and higher voltage regarding Hot Carrier degradation and Gate Oxide Time Dependent Dielectric Breakdown (TDDDB)). At each stress condition the times to failure of several devices are measured. These failure times (also called lifetimes) are considered to say something about the reliability of the devices at normal (non-stressed) conditions.
The present model for the distribution of the lifetimes (e.g. Hot Carrier degradation: log-normal distribution; Gate Oxide TDDB: Weibull distribution). Another possibility is a “physical” model relating one or more parameters of the lifetime distribution to conditions (i.e. current, temperature and/or voltage) and to dimension of the device (e.g. effective length of the transistor or area of the capacitor). Given the coefficients of the models, the lifetime distribution is determined and the time at which at most 0.1% of the devices fail at use conditions can be predicted. Normally, this time should be at least 10 years. Frequently in accelerated life tests, times are “censored”. There are two reasons for the occurrence of censored times:

Devices do not fail within the planned time span. Because results must be reported, the test is stopped and data have to be analyzed.

Devices fail as a result of another failure mechanism that has nothing to do with the failure mechanism in consideration (e.g. mechanical damage by the probe as a result of vibration, damage because the power supply was not uniform, etc.)

In these cases data must be analyzed with two types of time:
lifetimes of devices that failed because of the failure mode in consideration, and
times of devices that survived a certain time and did not fail because of the failure mode in consideration. So, from these devices we do not know the lifetime, but we know that the lifetime is at least a certain value (w.r.t. the failure mechanism in consideration). We call these times (right) censored (since the lifetimes to the right are missing).

Both types of time contain information about the reliability of the device (w.r.t. the failure mechanism in consideration), but of course they should be treated differently.
Note that the best devices (i.e. with high lifetimes) have the highest probability of censoring, so we would make a systematic error (bias) by simply deleting censored times.

After understanding the new method a script that could analyze the raw data in Maximum Likelihood Method was made. It also had to include the filtering, clearing out unnecessary values etc. A rough program was made to compare the results to currently used method and it was found to be reliable. So more functionalities and features like real time updating, di-electric constant, thickness, area etc was added which made it easy to be used and also more accurate.

3.4 HOW TO RUN THE SOFTWARE?

3.4.1 Parallel Program for Switching Matrix

![Interface for parallel switching matrix]

Figure 3.4.1 Interface for parallel switching matrix
The software for the HP5250 AND 4083A runs from within the KITE. The GPIB address is the address of the machine. For the HP5250 parallel program the user has to input pin number to the corresponding SMU number. The values are to be separated by “;”. It can take up to eight connections per SMU. This makes it possible to have eight connections in parallel measurement. This greatly improves that time for measurement and gets the output values much faster than the previous program.

3.4.2 Program for 4083A

This program is relative simple to use. The GPIB address is the address of the machine. The “1” means its on and “0” means its not selected. For the I/C connection, “1” will choose I(CURRENT) and “0” will chose (CAPACITANCE).

Figure 3.4.2: Interface for 4083 switching controller
### 3.4.3 Program for Parallel TDDB measurement

As in first case the GPIB address is the address of the HP5250. The values like required voltages, Extrapolation, array size, abort current etc are to be filled in according to the wafer. The pinning is similar to that of parallel connection program. This program as mentioned above is the most hardest of all. It was partially due to the fact that it communicated with two machines and also had to analyze data in the mean time. The analyzed data was ported to KITE in real time. This was displayed as values and also as graph.

The program had basically started by connecting all pins in parallel to the SMU for parallel measurement. After each measurement the returned value was analyzed. If it was seen to be out side the pre given range the pins were disconnected and the machine was reset. After that the first pin was connected to the SMU and the value measured again to check if it was in the range. If it conformed to the standard it was give the status “ok”. Then it was disconnected and the next pin connected to the SMU and status checked. This went on until the pin which was not conforming to the standards was found. The status of such pin(s) was given “not ok”. After each pin was tested the pins with status “ok” was connected in parallel again and stressed in parallel. This continued on until all the pins had broken down. From the pre defined settings it was possible to see if there was a short or open with the pins. For short, a value of “1e+21” and for open “1e-21” was given. This made very distinct lines in graph making analyzing graphs easier. After all the pins connected to an SMU had broken down a value of “1e-51” was assigned to the SMU.

Created by Noble Paul Kanjookaran
Figure 3.4.3: Interface for Parallel TDDB program

3.4.4 Maximum Likelyhood Method Script

Figure 3.4.4: Starting up of Maximum Likelyhood Method
The Maximum Likelihood Method script was loaded as an ‘xla’ file into excel. It was shown with a menu which made it easier to run. This program doesn’t need any inputs like the previous ones. As seen from the picture the program takes the input as the data in sheet. The program was activated when the ‘Filter’ button was pressed. The file basically analyzed the values in the opened file and filtered them according to the data given by the user. The result was stored in a new sheet. Subsequent pages were made after each other with modifications as necessary. The processed values were always copied to a new sheet in case the user wanted to analyze the data which was processed. Only few user inputs were necessary namely in filtering values. Suspicious values were highlighted and presented to user for action.

After all the values were checked the final part of the program could be run by pressing the ‘Solver’ button. The second part starts where the first part ended and does the Maximum Likelihood Method and shows the result in a graph.
CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Parallel TDDB measurement

After the program is executed it does various steps like communicating with the switching matrix, making necessary connections etc. After the locations are measured the data is shown as a graph. In the graph shown below it means that SMU1 broke down at 1 seconds, SMU2 at 35 seconds, SMU3 at 52 seconds and there is no contact for SMU4. It gives the data in an easy to use format and clear view of breakdown times. The picture shown below is an example for single connection per SMU. Multiple pinning per SMU would be unsuitable for demonstration in here.

![Graph showing TDDB program output](image)

**Figure 4.1:** Sample TDDB program output
4.2 Maximum Likelyhood Method script

The program as mentioned before takes the data available as output from the machine and processes the data to get easy to understand outputs. As shown in the figure below the final page is shown with a graph that gives the lifetime in a graphical manner. The graph, which shows the lifetime, is specifically shown below. As seen from the picture the result page has all data that it used to get the result and various constants that can be changed to get a “what would have happened if “?” results in real time.

Figure 4.2(a): Maximum Likelyhood Method result page
As seen from the picture above this is the final result after executing the script. In this case it means that for gate oxide of area $0.1\,\text{cm}^2$, a chance of defect of 0.1\% at voltages greater than 2 Volts is about $1\times10^7$ seconds. The graph is simple and shows only relevant information. This makes predictions easier and less tiring than old method.

### 4.3 LESSONS LEARNT AND REFLECTIONS

I have derived many benefits from the work that I have done on this project. Some of the benefits are as follows:

- I learnt how to do research on a given subject and plan my tasks according to the schedule.
- I learned several concepts in the field of programming, communication and new softwares.
• I learnt how to operate several industrial machines and learned their functions.

• I learnt to use some softwares like KITE, KULT, VBA etc

4.4 ENVIRONMENTAL CONSIDERATIONS

During the development of the project, many processes were involved which required the usage of different types of materials and resources. Printers weren’t used much as this needed papers, which in turn mean using trees. So until the final part of the project where everything was confirmed and tested the program was only as softcopy. Unnecessary use of resources was avoided in testing of software. It was tried to use non-destructive testing for most part of the project unless where it was necessary. Also the data and other communication were also saved which could make troubleshooting easier. More over all the printed papers were used for future reference and double-sided printing was employed at all times. The computer was shutdown after use and screen turned off wherever unnecessary.
CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 CONCLUSION: Parallel TDDB

5.1.1 Objectives Met

The program had to connect various pins in parallel. For the parallel TDDB measurement the switches had to be connected and disconnected as the measurement progressed and results shown in the KITE. The specific objectives for this robot were to:

- Develop a reliable program.
- Possibility to connect up to 8 pins per SMU.
- Ensure that the timings for the breakdown is accurate.
- Ensure that the program gets suitable inputs and its corresponding outputs.
- Design a program that can get easy to use outputs in standard programs.

The new program could give better use of existing machines without any additional hardware. This saved money and also gave higher efficiency for each machine. Measurements could be done in parallel and this reduced the time needed to measure an entire wafer by about 1 to 8 times. Since the software for parallel TDDB also included the switching matrix it was portable to another machine without any modifications.
5.1.2 Objectives Not Met

There are currently only 4 SMUS in the SCS 4200. The current program can accommodate all of that. But it can’t be used for more than 4 SMU’s. But there is room for further addition of SMU in the program

5.1.3 Recommendations for the Parallel TDDB program

- The software should have more interaction with the KITE software
- Various colors should automatically be given to the various breakdown times.
- Unnecessary lines of codes could be removed if the KITE could accept many standard C++ program’s functions
- Kite’s properties should be available as functions.
- Kult should be improved for a more versatile code editor.

5.2 CONCLUSION: Maximum Likelihood Method Script

5.2.1 Objectives Met

The objective was to use a new method called Maximum Likelihood Method for better results. The program was to be faster and accurate than the previously used method

As the developer of this project, my specific objectives were to:

- Make a script that can get the lifetime better than that currently used script.
- Easy-to-use
- Ensure that the script gives consistent outputs
- Ensure that the scripting is able to accommodate future modifications
• Ensure that the output is easily understandable with graph

• Use maximum likelihood method

The script was made and tested successfully without any problems. The results were compared to previously used method. The time for the script to run was lesser and the result was shown as an easy to understand graph without any unnecessary details. Some other functionality were added to it in the mean time which eliminated some error values that were caused by machine errors, non contact etc. Doubtful values were highlighted for the user to change before the final part of the result was calculated. The project was also done in such a way that the old method could also take the input from the first part of output of this program. This made it also easier to use the earlier script in case if it was necessary to compare them both.

5.2.2 Objectives Not Met

All most all the objectives were met in the final version. This was the result after modifications to previous ones. In the end the program was reliable and easy to use and had many features that were necessary.

5.2.3 Recommendations For Maximum Likelihood Method

• Faster execution time might be avoided if only necessary parts of the file are copied.

• Instead of comparing each value in a cell other method could make it faster for error correction etc.

• The output of the method is highly dependent on the inputs. This could give inconsistent results. A combination of previous and new method could be used.
REFERENCES

   http://www.nxp.com/profile/

   c=US&lc=eng

   [Online]. Available: 
   http://www.ines.de/gpib-info.html

   [Online]. Available: 
   www.keithley.com/data?asset=50400


   Available: 
   www.cascademicrotech.com/go/engineering-products-division/service-and-support

   http://www.tempronic.com/tp03010_a_b.htm

[8] Zyvex Instruments (2006), *Four Point Probe I-V Electrical Measurements Using the 
   Zyvex Test System Employing a Keithley 4200 by By Jeff Hochberg* [Online]. Available: 
   http://www.tempronic.com/tp03010_a_b.htm

   http://en.wikipedia.org/wiki/Visual_Basic_for_Applications
List of Figures

Figure 2.1.1 HP 5250
Figure 2.1.2(a) HP 4083
Figure 2.1.2(b) Commands for 4083A
Figure 2.1.3(a) Alessi Prober in setup
Figure 2.1.3(b) Joy Stick to Control the prober
Figure 2.1.4(a) SCS 4200
Figure 2.1.4(b) Close view of SMU in DSCS 4200 showing GPIB connection
Figure 2.1.5(a) ECX-56 Expansion Module and KL 1500 Light Intensity Controller
Figure 2.1.5(b) Various Stepper motors used for correct positioning of wafer
Figure 2.1.5(c) Highlighting importance of lighting in a setup
Figure 2.1.6(a) Various probe cards
Figure 2.1.6(b) Probe cards in setup
Figure 2.1.7 Hot Chuck Controller
Figure 2.2.1(a) KITE in action
Figure 2.2.1(b) Configuring Parameters in KITE
Figure 2.2.3 Screen Shot of Visual Basic for Excel
Figure 2.2.4(a) Cascade Microtech Galaxy
Figure 2.2.4(b) Location measured in real time
Figure 2.2.4(c) Wafer map in Cascade Microtech Galaxy
Figure 2.2.4(d) Controlling Parameters in Cascade Microtech Galaxy
Figure 3.3.1 MSGCONV during GPIB communication
Figure 3.4.1 Interface for Parallel switching matrix
Figure 3.4.2 Interface for 4083 switching controller
Figure 3.4.3 Interface for parallel TDBB program
Figure 3.4.4 Starting up of Maximum Likelihood Method
Figure 4.1 Sample TDBB output
Figure 4.2(a) Maximum Likelihood Result Page
Figure 4.2(b) Lifetime prediction graph
APPENDIX A

User guide for the Programs
Parallel TDBD Program

Figure: Screen shot of KITE

As seen from the picture this is the user library from KITE. There are many libraries for various uses. The library for Parallel TDDB measurement is TDDB_LIB. After selecting it the following files in the library appears as below.
Figure: The list of files in TDDB_LIB

As seen from the picture above the folder shows the list of files in the TDDB_LIB. The relevant file to be used is lastdone. It is a fully built in with control of HP5250. So no other files to control the switching matrix are needed. If its necessary, Keithley message Console can be activated by start->run->and typing in “msgcon” (without quotes). This would bring up a screen, which shows the communication between the two machines in a low level without any changes.
**Figure**: Physical Location of source files

The following picture illustrates the physical location of the source files. It is in the folder named “src”. It is the same for all the libraries. In this case the “src” folder will contain the files of HP5250 user library. However it is to be noted that the amendments done to the file will only be noted after recompiling them in KULT.
Maximum Likelyhood Method Script

Figure: The starting point of Maximum Likelyhood method script.

The most important thing to be noted is that the tab has to be renamed to “raw_data” as shown in the figure. Pressing the “New_Met” button in the toolbar can activate the interface. This would give rise to the screen as shown below.
Figure: User Interface ready for Maximum Likelihood Method

As shown in the picture the highlighted button “Filter” filters and makes necessary changes to the existing file and sorts it with necessary information into a sheet called “Sorted”. A new sheet called “Filtered Result” is made from sorted sheet after necessary user defined filtering. After it is done the script comes to a temporary halt. The user is given time to review the work done and to make any necessary changes as needed.
After necessary changes are done it is ready for the final analysis. The button called “Solver” is to be pressed. This copy over the data from the “Filtered Result” sheet into a new sheet called “Maximum_Like” sheet. All the necessary calculations are done here and the final values are copied to the last sheet namely “Maxlike” as seen from the figure below.
**Figure:** Maximum Likelihood Method Script after full analysis

After all the necessary calculations are done this is the final sheet. The values in the cells are linked to each other. Any change made to a cell will definitely affect all other values.

The graph with result is shown on the right. An enlarged view is shown below.
Figure: The enlarged view of graph from Maximum Likelyhood Method Script part

This is the graph after necessary calculations which show the predicted life time. The blue dots are the data.
**Figure:** Physical Location of ‘xla’ files in Excel start up folder.

This picture shows the physical location of the ‘xla’ files. It is normally in C:\Program File\Microsoft Office\Office\XLstart. Any ‘xla’ file copied to this directly will be loaded as the Excel starts.
APPENDIX B

Detailed information on Maximum Likelihood Method

By Ramun Kho
Temperature cycling (e.g. due to power on/off) may cause mechanical fatigue damage. To investigate this, stress tests are used with more extreme temperature swings (ranges). The Coffin-Manson model is a simple and often used model to translate the number of cycles to failure under stress conditions, to that under use conditions:

\[ N_f = A \left( \frac{1}{\Delta T} \right)^m \]  

[1]

where \( N_f \) is the number of cycles to failure, \( A \) and \( m \) are material “constants” and \( \Delta T \) is the temperature swing. In this model [1] the frequency of the cycles is assumed to be constant or to have an effect that is negligible.

In general, for the same material and temperature swing, varying numbers of cycles to failure will be realized. So, the number of cycles follows a probability (lifetime) distribution. In this report a (two parameter) Weibull distribution\(^1\) is assumed (each particular temperature swing has its own Weibull distribution):

\(^1\) The Weibull distribution can be expected when the weakest link principle applies: the cycling affects many similar spots and the first failing spot causes the product to fail. If at each cycle a random amount of degradation accumulates (i.e. due to the central limit theorem the total amount is normally distributed) until a certain critical value is reached beyond which failure occurs, another distribution that might be considered in this context is the Birnbaum-Saunders fatigue life distribution:

\[ F(t) = \Phi \left[ \frac{1}{\sigma} \left( \frac{t}{\beta} - \sqrt{\frac{t}{\beta}} \right) \right], \]

where \( \Phi \) denotes the standard normal CDF and where \( \beta \) is the scale parameter proportional to the threshold critical value.
where $F(t)$ is the cumulative fraction of units that fails before time (or number of cycles) $t$, $t$ is failure time (or number of cycles to failure), $\alpha$ is the characteristic life (or number of cycles where 63.2% has been failed) and $\beta$ is the shape parameter of the distribution. The assumption is made that the acceleration by a higher temperature swing is true and linear. In this case the lifetime distribution remains Weibull with constant shape parameter ($\beta$). The characteristic life ($\alpha$) is dependent of the temperature swing and is described by [1]. In that case, parameter $A$ can be interpreted as the characteristic life at a temperature swing of one.

If models [1] and [2] are valid and if parameters $A$, $m$ and $\beta$ are known, any percentile at any temperature swing can be calculated: $A$ and $m$ are used in model [1] to calculate the characteristic life $\alpha$ at the given temperature swing; this is substituted together with $\beta$ into [2] to calculate the number of cycles to failure ($t$) for a given cumulative fraction ($F(t)$).

In general for a certain product/package/solder material, parameters $A$, $m$ and $\beta$ need to be estimated by experiments. Because of the power relations, small deviations in these estimates may lead to very large deviations (orders of magnitude) in the predicted number of cycles when extrapolating (to another $\Delta T$ and/or cumulative fraction $F(t)$); see appendix C. So, it is important that an estimation method for these parameters is used that
is statistically optimal and that uses all information available in the data. Such a method is described in this report. Excel tools with an implementation of the method are given in the appendices.

**Weibull fits for each temperature swing**

**Experimental design considerations**

To estimate parameters $A$ and $m$ of the Coffin-Manson model [1], experiments must be done with at least two different temperature swings. With *two* different temperature swings the parameters can be estimated, but the validity of the model [1] cannot be checked. With *three* (or more) different temperature swings, the parameters can be estimated and the validity of the model [1] can be checked as well.

At each temperature swing several products (30 – 40) are put on test and for each product the number of cycles to failure is registered.

**Preliminary analysis**

The first step of the analysis is to make for each temperature swing separately a probability plot where $\ln(-\ln(1-F(t_i)))$ is plotted against $\ln(t_i)$. This is possible with e.g. the Excel tool *WeibullFit.xls* (Appendix A). Points of attention should be:
Observations must be on a straight line. If they are not on a straight line, the distribution might not be Weibull. Especially systematic deviations from this straight line indicate that the distribution is not Weibull.

Individual points that clearly deviate from the rest should be checked: they may be outliers.

Points that are far at the left side of the line (having a time which is much “too short”) might be censored: failure occurred because of another mechanism than the mechanism under investigation. If there is evidence that this is the case, it can be marked in the Excel tool by placing a “1” in the column next to the time (number of cycles).

The Excel tool WeibullFit.xls fits the data by the least squares method and by the Maximum Likelihood method (after pressing the “Fit” button). The Maximum Likelihood method gives as estimates for the parameters $\alpha$ and $\beta$, those values that maximize the loglikelihood:

$$
\ln L(\alpha, \beta) = \sum_{i=1}^{r} \left( \ln(\beta) - \ln(t_i) + \beta \cdot (\ln(t_i) - \ln(\alpha)) - e^{\beta \cdot (\ln(t_i) - \ln(\alpha))} \right) + \sum_{j=1}^{n-r} \left( - e^{\beta \cdot (\ln(t_j) - \ln(\alpha))} \right) \tag{3}
$$

where $t_i \ (i=1...r)$ are failure times and $t_j \ (j=1...n-r)$ are censored times. Especially when censored data are present, these estimates are more accurate (fig 1).
Fig. 1 Least squares fit (left) and maximum likelihood fit (right) of the same data. The first two observations failed due to another mechanism and were marked as censored.

The Excel tool in Appendix A also fits the three-parameter Weibull distribution, but this is not used further.

Check on the assumption true and linear acceleration

Each temperature swing gives estimates for scale parameter $\alpha$ (the higher the temperature swing, the smaller $\alpha$) and shape parameter $\beta$. The estimates for $\beta$ should be more or less the same (i.e. the probability plots must have more or less the same slope).
Initial estimates of Coffin-Manson parameters

By taking the logarithm of [1], the Coffin-Manson model becomes:

\[
\ln(N_f) = \ln(A) - m \cdot \ln(\Delta T)
\]

For the 63.2th percentile, this is:

\[
\ln(\alpha) = \ln(A_{63}) - m \cdot \ln(\Delta T)
\]

[4]

In the former section for each temperature swing the Weibull distribution was fitted, giving for each \( \Delta T \) estimates of \( \alpha \) and \( \beta \). When calculating the logarithms of \( \alpha \) and \( \Delta T \) and plotting \( \ln(\alpha) \) against \( \ln(\Delta T) \), according to equation [4] we expect a straight line.

Fitting equation [4] by linear regression, gives as an initial estimate of \( A \):

\[
A = e^{\text{intercept}}
\]

and an initial estimate of \( m \) is:
The linear regression and these calculations are done by the “Initial” sheet in the Excel file MaxLike Coffin Manson.xls (appendix B).

Check on Coffin-Manson model

If experiments were done at three or more temperature swings, all points in the plot of $\ln(\alpha)$ against $\ln(\Delta T)$ should be on the straight line. If the points are clearly not on a straight line, this indicates that the Coffin-Manson model is not valid.

Maximum Likelihood estimates of Coffin-Manson parameters

The Coffin Manson model (equation 4) can be substituted into the loglikelihood of the Weibull distribution (equation 3). This gives:

$$\ln L(A, m, \beta) = \sum_{i=1}^{r} \left( \ln(\beta) - \ln(t_i) + \beta \cdot (\ln(t_i) - \ln(A) + m \cdot \ln(\Delta T_i)) - e^\beta (\ln(t_i) - \ln(A) + m \ln(\Delta T_i)) \right) +$$

$$\sum_{j=1}^{n-r} \left( -e^\beta (\ln(t_j) - \ln(A) + m \ln(\Delta T_j)) \right)$$

[5]

In this equation:

for each failed product $i$ ($i=1…r$), the accompanying temperature swing $\Delta T_i$ and the failure time $t_i$ are known,
for each censored product \( j \) (\( j=1\ldots n-r \)), the accompanying temperature swing \( \Delta T_j \) and the
censored time \( t_j \) are known, and

the Coffin Manson parameters \( A \) and \( m \) and the Weibull shape parameter \( \beta \) are the only
unknowns.

The parameters that maximize the loglikelihood [5] are the maximum likelihood
estimates. They must be found numerically. To avoid convergence to a local maximum,
the initial estimates of section 3 can be used as starting values for \( A \) and \( m \). The initial
estimate of shape parameter \( \beta \) can be the average of the \( \beta \)’s for each temperature swing
(section 2).

The necessary calculations are done in the MaxLike sheet of the Excel file MaxLike
Coffin Manson.xls (appendix B).

Conclusions

Temperature cycling experiments investigating mechanical fatigue damage are often
analyzed with the Coffin-Manson model in combination with the Weibull lifetime
distribution. Probability plots (section 2 and appendix A) give insight in the data and a
visual check on model assumptions. Therefore they are valuable for a first analysis. The
Coffin-Manson model parameters could be estimated by a regression method (section 3
and appendix B).
Small deviations in the estimations of the model parameters may cause large deviations in predicted (extrapolated) lifetimes (appendix C). Therefore it is important to use the most efficient analysis method that uses all information available in the data. Such a method (maximum likelihood) is described in this report and implemented in Excel tools (section 4 and appendix A and B). The maximum likelihood estimates should be used for further calculations and extrapolations.
Appendix A  Excel tool WeibullFit.xls

This tool fits the Weibull distribution by the least squares method and by the maximum likelihood method (see section 2).

Fill in the data (failure times or censored times) in column A of the "data" sheet (in cell A12:A303).

Indicate in column B whether the time should be interpreted as a censored time (cens=1) or a failure time (cens<>1)

Press the **Fit** button. This will result in:

The least squares estimations are used as starting values. I.e. copied to cell H9:H10

The loglikelihood (cell I7) of the 2 parameter Weibull is maximized with Solver by changing cell H9:H10
The least squares estimations are used as starting values. I.e. copied to cell J9:J10. Cell J8 (initial failure free time) is set to zero.

The loglikelihood (cell K7) of the 3 parameter Weibull is maximized with Solver by changing cell J8:J10.

Before usage of the Fit button, establish a reference to the Solver add-in:

Go to Tools → Macro → Visual Basic Editor

In the Visual Basic Editor go to:

Tools → References

check SOLVER

If SOLVER is not present

Press Browse

And open SOLVER in

C:\Program Files\Microsoft Office\Office\Library\Solver\SOLVER.XLA
Appendix B    Excel tool MaxLike Coffin Manson.xls

This tool estimates the parameters of the Coffin-Manson model as well as the Weibull shape parameter by the regression method (section 3) and by the maximum likelihood method (section 4).

Estimate for each temperature swing separately the Weibull parameters $\alpha$ and $\beta$ with WeibullFit.xls (Appendix A).
Enter these estimates in the "Initial" sheet (in cell B7:D25). Initial estimates for $A$, $m$ and $\beta$ are calculated (in cell J17:J19) by the regression method (see section 3).

Fill in the data in the "MaxLike" sheet (in cell B8:D507)
Press the **Fit** button. The estimates of the Initial sheet (J17:J19) are copied to the MaxLike sheet (to cell B3:B5) and the maximum likelihood estimations are calculated (see section 4).
Appendix C  Simulated example

Two samples from 30 units each were simulated from a Weibull distribution with shape parameter $\beta=2$ and scale parameter $\alpha$ that is determined by:

$$N_f = 5000000 \left( \frac{1}{\Delta T} \right)^{1.4}$$

So, the scale parameter was exactly determined by the Coffin Manson model with $A=5000000$ and $m=1.4$. One sample was drawn with $\Delta T=165$ and the other with $\Delta T=215$.

The simulated number of cycles at each temperature swing are:

<table>
<thead>
<tr>
<th>$\Delta T=165$</th>
<th>$\Delta T=215$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1121</td>
<td>99</td>
</tr>
<tr>
<td>1248</td>
<td>443</td>
</tr>
<tr>
<td>1452</td>
<td>521</td>
</tr>
<tr>
<td>1741</td>
<td>534</td>
</tr>
<tr>
<td>1812</td>
<td>1256</td>
</tr>
<tr>
<td>1830</td>
<td>1335</td>
</tr>
<tr>
<td>2215</td>
<td>1357</td>
</tr>
<tr>
<td>2296</td>
<td>1790</td>
</tr>
<tr>
<td>2585</td>
<td>1884</td>
</tr>
<tr>
<td>2805</td>
<td>1899</td>
</tr>
<tr>
<td>3046</td>
<td>1959</td>
</tr>
<tr>
<td>3193</td>
<td>2007</td>
</tr>
</tbody>
</table>
The objective is to determine the characteristic number of cycles (63th percentile) at a temperature swing of 20. In reality, according to the model, this number of cycles is:

\[ N_f = 5000000 \left( \frac{1}{20} \right)^{1.4} = 75427 \]

but we simulate that we do not know the model parameters. The data were used to estimate this number by the regression method (described in section 3) and by the maximum likelihood method (described in section 4).
For each temperature swing, the estimated Weibull parameters (section 2) were:

<table>
<thead>
<tr>
<th>Method</th>
<th>$\Delta T = 165$</th>
<th>$\Delta T = 215$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Least squares</td>
<td>$\alpha$</td>
<td>4432</td>
</tr>
<tr>
<td></td>
<td>$\beta$</td>
<td>2.164</td>
</tr>
<tr>
<td>Maximum likelihood</td>
<td>$\alpha$</td>
<td>4445</td>
</tr>
<tr>
<td></td>
<td>$\beta$</td>
<td>2.172</td>
</tr>
<tr>
<td>True values</td>
<td>$\alpha$</td>
<td>3931</td>
</tr>
<tr>
<td></td>
<td>$\beta$</td>
<td>2</td>
</tr>
</tbody>
</table>

Due to sampling fluctuations, the estimates deviate from the true values. For the sample at $\Delta T = 165$, the difference between the least squares and the maximum likelihood methods is small; for the sample at $\Delta T = 215$, the difference is larger, with the maximum likelihood estimates closer to the true values.

The estimates of the Coffin Manson parameters and of the predicted number of cycles at a temperature swing of 20 are:

<table>
<thead>
<tr>
<th>Method</th>
<th>$A$</th>
<th>$N_f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>24203127</td>
<td>155327</td>
</tr>
<tr>
<td>(section 3)</td>
<td>1.6853</td>
<td></td>
</tr>
<tr>
<td>Maximum likelihood</td>
<td>15000000</td>
<td>126546</td>
</tr>
<tr>
<td>(section 4)</td>
<td>1.594</td>
<td></td>
</tr>
<tr>
<td>True values</td>
<td>5000000</td>
<td>75427</td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td></td>
</tr>
</tbody>
</table>
The estimates are larger than the true value (this is due to random sample fluctuations). The regression method gives estimates that deviate more from the true value than the maximum likelihood estimates.

Conclusions
Due to sampling fluctuations, the estimates of the Weibull parameters may deviate with tenths of percents.

Due to sampling fluctuations, the estimates of the Coffin Manson parameters and the extrapolated value under use conditions may deviate with factors.

The Maximum likelihood estimates deviated less than the Least squares estimates. This was the case with only one simulated case and may be different in another case. However when these simulations would be repeated thousands of times, we would see that in average the maximum likelihood estimates are closer to the true values than the least squares estimates.
APPENDIX C

Problems Faced and Solutions Found
PROBLEMS FACED AND SOLUTIONS

Parallel TDDB

**Problem 1:** Channels were not closed at all times

The HP 5250 didn’t close its channels even if the same program was run at different times.

**Solution 1:** After some investigation it was found that the machine needed to be reset for the channels to change again.

The connections could be changed only after resetting the machine. Though the machine would work without resetting at times, there was a high possibility that it would not work as intended if it was not reset.

**Problem 2:** Erratic results were observed from measurement at times in parallel TDDB.

Some times there was no output from machines for no reason. The data was sent successfully from the computer. But it did not get much back.

**Solution 2:** Checking the communication Lines

Firstly, the program was tested part by part. After ruling out the possibility that the parts had errors in them, they were assembled back. Since the same problem existed after
assembling the manual of the book was read. It was noted that the communication lines were not always free. So this was corrected and the program was running again.

**Problem 3:** The machine returned some values that seemed like there was no contact between the pins at times.

Randomly it appeared that there was no connection between various pins. It was strange since the same program was used each time.

**Solution 3:** Delay after connections

The pins were checked by making deliberate shorts and opens to make sure that the program was working perfectly. Since the program was responding to artificial errors like it was supposed to it seemed that there was no problem. After some more investigation, it seemed that the problems came from switching matrix. And after running some testing programs it seemed that the connections could not be re-established after the switching was made. So a delay of 200 ms was given after each switching to complete the operation.

**Maximum Likelihood Method**

**Problem 1:** The solver would not run from VBA and it showed errors when called.

As a part of maximum likelihood method it was necessary to run the solver from VBA.

However there were some errors, which made it impossible to be run.

**Solution 1:** Use an add-in.

After reading through many forums and discussing it with other students, it was found that before solver was called from VBA it needed a reference to excel to be run. A reference is a small file which points to the solver in excel. After adding this reference the program was working well.
**Problem 2:** Solver couldn’t run with values inserted from VBA. It was found that for the solver to find the best solution, it needed all the values to be dynamically linked to each other. But since the values were inserted by loops in VBA, they weren’t linked to each other and so the solver could run only once.

**Solution 2:** As a solution to this it was found out that a file could be pre created with dynamically linked values. When the program was run, it made a copy of the file and used it. The values were then changed to reflect new values. But they were dynamically linked as the file was before.