Hamm-Lippstadt University of Applied Sciences

– Faculty of Electronic Engineering –

Simulation of PCBA robustness in alignment with the pre-development project titled ”Reliability Prediction”

Thesis for the attainment of the academic degree

Bachelor of Engineering (B.Eng.) in Electronic Engineering

submitted on 16 March 2024

Saikot Das Joy

Student ID: 2190340

First Supervisor	:	Prof. Dr.-Ing. Ali Hayek
Second Supervisor	:	Prof. Dr.-Ing. João Paulo J. da Costa

Non-Disclosure Notice

Acknowledgements

I would like to express my sincere gratitude to my supervisors, **Prof. Dr.-Ing. Ali Hayek** and **Prof. Dr.-Ing. João Paulo J. da Costa**, for their invaluable guidance and support throughout this research endeavor.

I also extend my deepest gratitude to **Dr. Rimma Zhytnytska** from Hella GmbH & Co. KGaA for providing me with the real test data and for directly guiding me during my time at Hella GmbH & Co. KGaA in this scientific research.

I am also very thankful to **Dr. Stephan Böckhorst** and **Peter Krismann** from Hella GmbH & Co. KGaA for their invaluable support throughout this journey.

Declaration

Deutsch

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst und nur unter Benutzung der angegebenen Literatur und Hilfsmittel angefertigt habe. Wörtlich übernommene Sätze und Satzteile sind als Zitate belegt, andere Anerkennungen hinsichtlich Aussage und Umfang unter Quellenangabe kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen und ist auch noch nicht veröffentlicht.

English

I hereby assure that I have independently written the present thesis and have only used the specified literature and resources. Directly quoted sentences and parts of sentences are cited as quotations, and other acknowledgments regarding content and scope are indicated with proper references. The thesis has not been submitted to any examination authority in the same or similar form and has not been published yet.

Lippstadt, 16 March 2024

Saikot Das Joy

Saikot Das Joy

Abstract

In the context of Electronics Reliability, the reliability of individual components plays a crucial role in determining the overall lifespan of electronics. Within the Automotive sector, especially in light electronics, Repeated changes in temperature cause fatigue in solder joints due to differences in expansion and contraction between components and the PCB. Over time, this causes the solder to develop cracks, which raises concerns about how dependable the final product will be.

This thesis investigates the possibility of predicting fatigue life of a specific type of solder through a simulation approach using a readily available simulation software. The simulation results obtained are then compared with actual laboratory observations. The theoretical framework supporting the simulation is also examined. Furthermore, a manual theoretical approach is used to predict fatigue life, considering specific geometric shapes of components, and the outcomes are then compared with real laboratory results.

Additionally, the impact of component modifications on solder fatigue reliability is evaluated through simulation, offering insights into variations in the overall lifespan.

Keywords: CTE (Coefficient of Thermal Expansion), Solder Fatigue, Solder Fatigue Simulation, SAC305 Solder, Innolot Solder, Aluminum IMS PCB, Copper IMS PCB, Stiffness, Total Axial Displacement, Maximum Strain Range, Mean TTF (Total Time to Failure), Theory Validation, Solder Fatigue Manual Calculation, Fatigue Life Comparison, LED Modifications, etc.

Contents

List of Figures

List of Tables

List of Acronyms & Abbreviations

Acronym / Abbreviation	**Description**
PCBA	Printed Circuit Board Assembly
LED	Light Emitting Diode
TTF	Total Time to Failure
TSF	Thermal Shock Factor
BTC	Bottom Terminated Component
CC	Ceramic Component
LCCC	Leadless Ceramic Chip carrier
SAC305	SnAgCu: Tin(Sn) 96.5%, silver(Ag) 3%, Copper(Cu) 0.5%
SnPb	Tin(Sn)-Lead(Pb) Solder alloy
PCB	Printed Circuit Board
IMS	Insulated Metal Substrate
CTE	Coefficient of Thermal Expansion (α)
Al	Aluminum
Cu	Copper
E	Elastic Modulus/Young's Modulus
G	Shear Modulus
t	Thickness
h / H	Height
A	Area
AF	Accelerated Factor
N_f **or N**	Mean Fatigue Life
α	Coefficient of Thermal Expansion
ΔW	Strain Energy
T_g	Glass transition Temperature
L_{comp}	Length of the component
L_d	Diagonal Length of the component
W_{comp}	Width of the component
L_{pad}	Length of the copper pad on the signal layer of PCB for soldering
W_{pad}	Width of the copper pad on the signal layer of PCB for soldering
L_{solder}	Length of the solder
W_{solder}	Width of the solder
SM	Surface Mount
TMM	Thermo-mechanical-microstructural
SI	International System of Units

Chapter 1

Introduction

1.1 Motivation

Within the realm of automotive technology, the term "Reliability" signifies the consistent and dependable performance of a vehicle, emphasizing its ability to operate without unexpected failures over time. Various subsystems contribute to the overall reliability of a vehicle, with electronics reliability standing out as a critical component. This reliability spans from the microscopic component level to the comprehensive electronics system level.

Ensuring the reliability of electronic components, particularly in car lights like LED (Light Emitting Diodes), is paramount for optimal vehicle performance and safety. As the automotive industry advances technologically, understanding and enhancing the reliability, including PCB robustness, of these components becomes increasingly vital.

The reliability of automotive lighting heavily relies on the solder connections between components and the printed circuit board (PCB). This significance is particularly evident due to the thermal environmental changes encountered by the delicate electronic components used in automotive lighting systems.

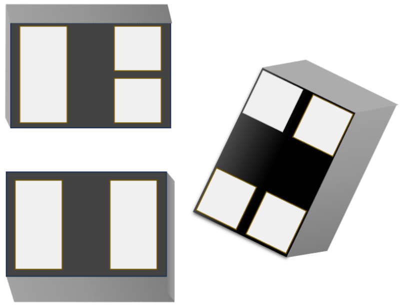
Successful prediction of these solder joint reliability with a simulation can significantly save time and costs, eliminating the need for extensive laboratory testing if the simulation approach proves acceptable through investigation. Current testing in the laboratory is time-consuming and entails considerable energy and effort expenditures. Moreover, if simulation results are deemed acceptable, parameter studies can be conducted through simulation to enhance the reliability of the components. This approach contributes to the development of highly reliable electronic design solutions, involving careful component selection and modifications.

The primary objective of this Thesis work is to investigate the applicability of a simulation software named ANSYS Sherlock to estimate the longevity of solder joints in Bottom Terminated Components (BTC : refer to the section 1.2). The focus centers on examining car light LED that utilize BTC solder models. The thesis also aims to elucidate the methodology employed by ANSYS Sherlock in conducting these simulations.

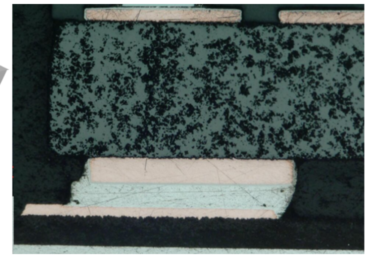
1.2 Bottom Terminated Component

BTC(Bottom-terminated components) are a class of leadless components with terminations protectively plated on the underside of the package. Varieties include QFN (quad flat no lead), DFN(dual flat no lead), LGA (land grid array), and MLF (micro lead-frame), each featuring different sizes, lead counts, and designs[17]. Suppliers provide unique parts with specific pad designs.

These components come in various sizes and are well-suited for applications requiring lightweight and thin profiles, making them ideal for handheld devices and consumer products. BTC components offer cost-effectiveness and are well-suited for large-scale applications. In this thesis work, LED will be utilized as an example of BTC components. A simple showcase of different pad designs and a cross-sectional view of BTC component's solder is illustrated in Figure 1.1.



(a) A BTC Component



(b) Cross Sectional View of a BTC Component (LED) Solder

Figure 1.1: (a) A Bottom Terminated Component (LED) with different pad designs (b) A BTC Component Cross-sectional Solder View from [16]

Transitioning to BTC components with a BTC Solder Model enhances electrical performance and offers excellent thermal dissipation. This approach provides a leadless device with low resistance and capacitance. The thermal pad directly connects to the printed circuit board (PCB), facilitating efficient heat transfer.

1.3 Solder Fatigue

In a broad mechanical context, fatigue refers to the cumulative damage and structural deterioration of a material subjected to repeated loading and unloading cycles. This phenomenon holds particular significance in materials science and engineering, where cyclic thermo-mechanical stresses (Stress: Details in Chapter 2, subsection 2.1.3) can lead to the initiation and propagation of cracks or fractures over time. Fatigue is often classified into various types, and Solder joint fatigue is a specific subtype relevant to materials exposed to varying temperature conditions.

Repeated stretching and contracting induce fatigue in materials. Likewise, when a component undergoes cycles of transitioning from low to high temperatures, it experiences fatigue. In the automotive sector, light-emitting diodes (LED) have become a common lighting component. Due to the frequent warming up and cooling down of these LED, they undergo a repetitive thermal cycle. Unfortunately, this continuous rhythm eventually fatigues the LED solder joint, leading to crack propagation and the failure of the component.

The Coefficient of Thermal Expansion (CTE)(CTE: Details in chapter 2, in subsection 2.1.1) plays a crucial role in this scenario. It measures how much a material expands or contracts when subjected to heating or cooling, quantifying the fractional change in size in response to temperature variations.

Differences in CTE between the printed circuit board (PCB) and the LED parts at varying temperatures contribute to the solder becoming fatigued and weak over time, ultimately leading to its breakage. Figure 1.2 provides a clear illustration of how changes in the Coefficient of Thermal Expansion (CTE) cause cracks in the solder, particularly focusing on the longevity of the solder in BTC models of these components.

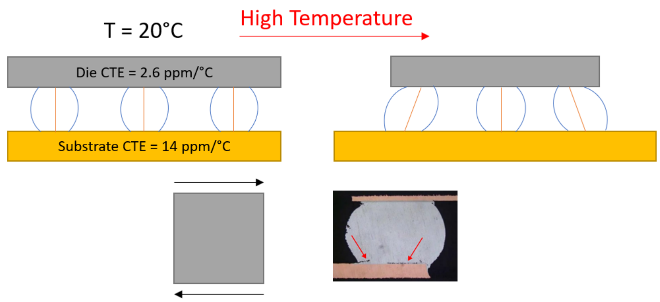


Figure 1.2: Solder fatigue caused by CTE differences.
The original image was sourced from [1] and has been recreated.

The present study is underpinned by the pre-development project titled "Reliability Prediction."

1.4 Pre development Project

The pre-development project, titled "Reliability Prediction," at HELLA GmbH & Co. KGaA, Lippstadt, Germany, was initiated with the goal of reducing time, effort, and energy expenditure on laboratory testing during pre-development as well as for product qualification. Predicting solder fatigue in the Bottom Terminated Components (BTC) solder model is a important part of this project.

Various methods can be used for this, including FEA simulation and ANSYS Sherlock solder fatigue simulation. However, this thesis specifically aims to explore how accurately ANSYS Sherlock can predict the solder fatigue life of BTC component solder. In simpler terms, it investigates how ANSYS Sherlock can help HELLA GmbH & Co. KGaA to predict the reliability of solder joints under thermo-mechanical stresses. The thesis also aims to scrutinize the underlying theory of the simulation and, if required based on the geometry, proposes slight modifications to refine accuracy.

1.5 Methodological Framework of the Thesis Work

The current work begins after the introduction in Chapter 1. The following chapters delve into explaining mechanical terms, reviewing existing literature, and modifying simulation outcome graphs for better comparison with real results in Chapter 2.

Moving on to Chapter 3, the focus shifts to understanding the ANSYS Sherlock Solder Fatigue Simulation theory for the solder model in CC (Ceramic Component). For example, a resistor is illustrated in Figure 1.3, and its solder fatigue simulation theory is also validated in this chapter. In the same chapter, the theoretical analysis of the BTC component solder model is explored taking input from CC Model theory and doing some analysis. The BTC theory is also validated by manually calculating the fatigue life based on it's theory and comparing it with the simulation results for a available BTC component from the Part library of ANSYS Sherlock.

(a) A Resistor (CC Component)

(b) CC Component Solder Model

Figure 1.3: (a) Example of a CC Component [16] (b) CC component's solder

Configuration setting for solder fatigue simulation in ANSYS Sherlock of a specific LED model, Samsung V Series (1 chip, 2 chip & 3 chip LEDs), are explained in Chapter 4. In Chapter 5, the predicted solder fatigue life outcome is compared with the lab-observed solder fatigue life of these components. Taking into account the specific geometrical pad design of these LED, a manual approach with a slight modification of the theory is followed in Chapter 6 with a comparison with real results again.

After that, as an additional work, the effect of the component modification concept of another LEDs model named OSRAM and NICHIA is investigated in Chapter 7 following the ANSYS Sherlock solder fatigue simulation. A summary & outlook of this work is provided in Chapter 8. All the python codes for data visualization and calculation of fatigue life manually are provided in the Appendix.

Chapter 2

Background and Review

2.1 Mechanical Terms

2.1.1 Coefficient of Thermal Expansion (CTE)

The Coefficient of Thermal Expansion (CTE) is a material property that quantifies the fractional change in size of a material for a unit change in temperature. It characterizes how a material's dimensions (length, area, or volume) change in response to a temperature variation. The CTE is denoted by the symbol α and is expressed in units of reciprocal temperature (e.g., $\frac{1}{^{\circ}C}$ or $\frac{1}{K}$).

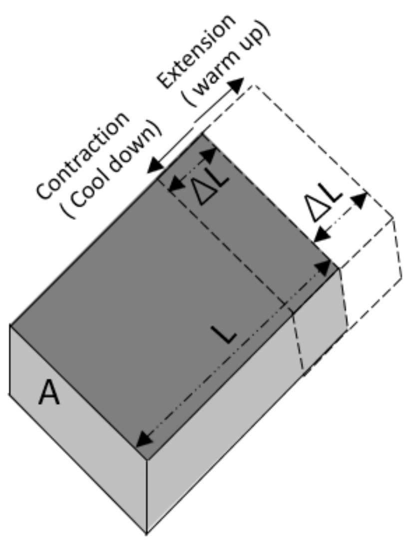


Figure 2.1: Coefficient of Thermal Expansion

The mathematical definition of CTE for linear expansion in Figure 2.1 is given by:

$$\alpha = \frac{1}{L} \cdot \frac{\Delta L}{\Delta T}$$

where:

- α is the Coefficient of Thermal Expansion,
- L is the original length of the material,

- ΔL is the change in length,
- ΔT is the change in temperature.

For area expansion or volumetric expansion, similar expressions can be derived. The CTE is a crucial parameter in various engineering applications, especially in the design of structures and components exposed to temperature variations.

The positive or negative sign of the CTE indicates the direction of the expansion or contraction concerning the temperature change. Positive CTE values signify expansion, while negative values indicate contraction. Different materials exhibit different CTE values, and engineers must consider these properties to prevent unwanted deformations, stresses, or material failures in response to temperature fluctuations.

2.1.2 Strain

Strain represents the deformation of a material under the influence of external forces and is defined as the ratio of the change in dimension to the original dimension. Strain is denoted by the symbol ε and can be expressed mathematically as:

$$\varepsilon = \frac{\Delta L}{L_0}$$

Where:

- ε is the strain,
- ΔL is the change in length, and
- L_0 is the original length.

2.1.3 Stress

Stress is a measure of the internal forces within a material, resulting from applied external forces or deformations. It is defined as the force per unit area and is denoted by the symbol σ. Mathematically, stress (σ) is expressed as:

$$\sigma = \frac{F}{A}$$

Where:

- σ is the stress,
- F is the applied force, and
- A is the cross-sectional area showed in Figure 2.1 over which the force is applied.

2.1.4 Stiffness

Stiffness is a measure of a material's resistance to deformation under the influence of external forces. It is characterized by the ability of a material to withstand deformation and maintain its shape. Stiffness is often represented by the letter k.

2.1.4.1 Axial Stiffness:

Axial stiffness represents a material's resistance to deformation along its axial direction. Mathematically, axial stiffness (k_{axial}) can be defined as the ratio of the applied axial force (F_{axial}) to the resulting axial displacement (δ_{axial}):

$$k_{\text{axial}} = \frac{F_{\text{axial}}}{\delta_{\text{axial}}}$$

Where:

- k_{axial} is the axial stiffness,
- F_{axial} is the applied axial force perpendicular to the area A in Figure 2.1, and
- δ_{axial} is the resulting axial displacement which is ΔL in Figure 2.1.

2.1.4.2 Shear Stiffness:

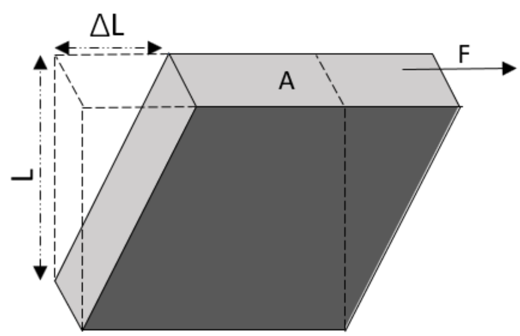


Figure 2.2: Shear Deformation

Shear stiffness is a measure of a material's resistance to deformation due to shear forces. Mathematically, shear stiffness (k_{shear}) can be defined as the ratio of the applied shear force (F_{shear}) to the resulting shear displacement (δ_{shear}) in FIgure 2.2:

$$k_{\text{shear}} = \frac{F_{\text{shear}}}{\delta_{\text{shear}}}$$

Where:

- k_{shear} is the shear stiffness,
- F_{shear} is the applied shear force, and
- δ_{shear} is the resulting shear displacement ΔL in Figure 2.2 .

2.1.5 Equilateral Hysteresis Loop

An equilateral hysteresis loop is a graphical representation of the relationship between two variables during cyclic loading and unloading of a material. It is commonly used in materials science and mechanics to depict the behavior of a material subjected to repeated mechanical loading and unloading cycles.

In the context of temperature and deformation, an equilateral hysteresis loop would typically be plotted with temperature on the y-axis and deformation (strain or displacement) on the x-axis. The loop in Figure 2.3 illustrates how these two parameters change as the material undergoes cycles of loading and unloading. The loop is "equilateral" because the loading and unloading paths coincide, forming a closed loop.

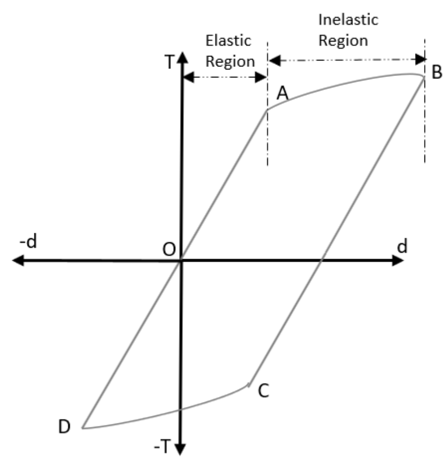


Figure 2.3: Equilateral Hysteresis Loop

Key features of an equilateral hysteresis loop include:

Elastic Region(O to A):

- considering the normal temperature at point O, as the temperature increase, The initial portion of the loop represents the material's response to temperature increase.
- As the temperature rises, the material undergoes reversible deformation.

In-Elastic Region(A to B):

- The material enters a thermal plasticity phase where a small changes in the temperatures causes more deformation in the material.This deformation in this region is not reversible fully..

Elastic Region (B to C):

- This part corresponds to a temperature drop. As the temperature decreases, the material undergoes a reversible elastic phase deformation.

Inelastic Region (C to D):

- Moving from point C to D represents an irreversible inelastic/plastic phase.

Temperature Rise to Normal (D to O):

- From point D to O, the material experiences a temperature rise, returning to the normal temperature at point O.

2.2 Literature Review

For years, SnPb has been the go-to solder alloy for traditional electronic devices, significantly influencing the reliability of solder joints. However, the shift towards non-toxic solders, prompted by regulations such as the EU's WEEE and RoHS directives since 2006, has sparked increased interest in alternatives. Consequently, the industry must align with environmental standards, given its reliance on electronic materials. Lead-free options, such as SnAgCu, are gaining popularity for their competitive pricing, mechanical properties, and lower melting points. Nevertheless, variations in silver and copper concentrations in these alloys can impact reliability. Thermal fatigue poses a significant threat to solder joint reliability, emphasizing the importance of understanding creep behavior for designing durable joints [26][36].

Solder interconnects play crucial roles, providing electrical connection, mechanical binding, and facilitating heat diffusion between the package and substrate. Due to varying coefficients of thermal expansion (CTE) among materials, powering on the device induces cyclic thermo-mechanical fatigue, necessitating simulation for reliability assessment.

In solder operating above half of its melting point with moderate thermo-mechanical stress, creep processes dominate. Nonlinear numerical simulation of creep strain accumulation employs a constitutive model, with equations 2.1 and 2.2 representing steady-state creep strain rates [38]:

$$\dot{\varepsilon} = A\left(\sigma^{n}\right)\exp\left(-\frac{Q}{RT}\right) \tag{2.1}$$

$$\dot{\varepsilon} = A\left[\sinh(\alpha\sigma)\right]^{n}\cdot\exp\left(-\frac{Q}{RT}\right) \tag{2.2}$$

where A is a material constant, σ is stress, n is the constant exponent, Q is an apparent activation energy and R is the universal gas constant.

Accumulated creep is obtained by integrating the constitutive relation. Mechanical responses and creep properties vary with specimen size. Constitutive models from solder joint measurements, especially for surface-mount (SM) assemblies, are crucial for engineering applications. Equations (2.1) and (2.2) describe the creep behavior of eutectic Sn/Pb and SnAgCu alloys [38], offering valuable insights for predicting solder deformation in electronic assemblies.

Darveaux and Banerji 1992 [6] and **Darveaux et al. 1995 [7]** have conducted extensive tests on actual soldered assemblies, considering the effects of grain and intermetallic compound distribution. Tensile and shear loading were applied in the strain rate range between 10^{-8} to 10^{-1} s^{-1}, and the temperature range between 25°C and 130°C.

They suggested using a sinh law, as proposed by Frost and Ashby 1982 [10], where, at intermediate stresses, the strain rate depends on stress to the power of n. At high stresses, the strain rate is an exponential function of stress. This power-law breakdown region can be described by a single expression:

$$\dot{\varepsilon} = C_1\left(\frac{G}{T}\right)\left[\sinh\left(\alpha\frac{\sigma}{G}\right)\right]^{n}\cdot\exp\left(-\frac{Q}{RT}\right) \tag{2.3}$$

where C_1 is a constant, G is the temperature-dependent shear modulus, T is the temperature in absolute scale, R is the universal gas constant, α prescribes the stress level at which the power-law dependence breaks down, and Q is the activation energy.

To obtain the true activation energy, the temperature dependence of the shear modulus must be incorporated:

$$G = G_1 - G_0 T'\,(°\mathrm{C}) \tag{2.4}$$

Table 2.1: Creep model constants by Darveaux [6]

Solder Alloy Deformation Constant Elastic Steady State Creep						
Solder	G_0 (10^6 lbf/in^2)	G_1 (10^3 lbf/in^2)	C_2 (k/s/lbf/in^2)	α	n	Q (eV)
60Sn40Pb						
Shear	1.9	8.1	0.198	1300.0	3.3	0.548
Tensile	5.0	22.0	0.114	751	3.3	0.548
1 eV = 96489 J/mol						

The shear modulus at 0 °C, denoted as G_0, represents material stiffness. G_1 indicates the temperature dependence, and T' denotes temperature in degrees Celsius. The necessary constants by Darveux are presented in Table 2.1.

To derive inelastic tensile constants from shear constants, the authors used the following relations, assuming Von Mises yield criteria apply [21]:

$$\sigma = \tau\sqrt{3} \tag{2.5}$$

$$\epsilon = \frac{1}{\sqrt{3}}\gamma \tag{2.6}$$

Where σ is the shear stress constant and τ is the inelastic tensile stress constant. Similarly, ϵ is the shear strain constant, and γ represents the tensile strain constant.

Pao et al. 1994 [3] conducted thermal cyclic shear stress/strain tests on 63Sn-37Pb solder joints using a double beam specimen. The temperature cycle had a 40-minute period with extreme temperatures of 40 °C and 140 °C. Steady-state creep properties were determined, and Norton's law was implemented in a finite element program to simulate the experiment. The authors discussed the fatigue life and failure mechanism of these solder joints. The corresponding steady-state creep properties and mechanisms for Sn-37Pb eutectic are as follows:

$$\dot{\gamma} = B \cdot (\tau^n) \cdot \exp\left(-\frac{\Delta H}{kT}\right) \tag{2.7}$$

The required constants are given in Table 2.2.

Table 2.2: Creep model constants by Pao [3].

Alloy	Deformation Mechanisms	B $\text{MPa}^{-n}\text{S}^n$	ΔH eV	n 140°C	n 140°C
Sn37Pb	Dislocation glid/climb	0.205	0.49	5.25	5.25

Osterman and Dasgupta (2007) [28] provide crucial constants for Garofalo's sinh-type model (Equation 2.2). These constants capture transitions in creep mechanisms within the relevant stress ranges, evident as changes in the slope of the log-log creep-rate versus stress curves. Refer to Table 2.3 for the detailed creep parameters.

Table 2.3: Creep model constants by Osterman [28].

Solder Alloy	A	σ	n	Q
Sn37Pb	1.15E4	0.20	2.2	5.93E4
Sn3.9Ag0.6Cu	1.50E3	0.19	4.0	7.13E4

In their physical measurements, it was observed that Pb-free SAC solder exhibits significantly greater creep resistance than Sn37Pb at low stress levels (1 MPa), where diffusion-driven mechanisms predominate. However, at higher stress levels (10 MPa), dislocation-driven creep mechanisms become predominant, and both solders show comparable creep behavior. This finding helps explain the observed longer life of Pb-free SAC solder in direct temperature cycling test comparisons with Sn37Pb. Additionally, it clarifies why acceleration factors for accelerated thermal cycling are higher for SAC solders compared to SnPb solder.

Zhang et al. (2003) [46] utilized a one-dimensional incremental model to simulate constant-load creep in a test setup. They conducted monotonic and isothermal cyclic mechanical tests across various temperatures, strain rates, and stresses using a thermo-mechanical-microstructural (TMM) test system. The study achieved primary creep, secondary creep, and plastic models for both Sn3.9Ag0.6Cu and Sn37Pb solders by fitting simulation results to constant-load creep and monotonic testing data.

Steady-state shear creep strain rates were reported using a sinh law (Equation 2.2), and the model constants for both solders are summarized in Table 2.4.

Constants of different lead-free solder alloys have been determined by **Schubert et al. 2003 [33]** in Table 2.5, using 108 data points from their measurement and literature.

Table 2.4: Creep model parameters by Zhang [46].

Solder Alloy	A	σ	n	Q
Sn37Pb	1999.4	0.2	2.1083	54137.2
Sn3.9Ag0.6Cu	248.4	0.188	3.7884	62916.7

Table 2.5: Creep model parameters by Schubert [33].

Solder Alloy	A S^{-1}	α MPa^{-1}	n	Q eV
Sn95.5Ag3.8Cu0.7 Sn95.75Ag3.5Cu0.75 Sn96.5Ag3.5Cu0.5	277984	0.02447	6.41	0.56

The Kariya model, as described by **Kariya et al. (2001) [18]**, fits well with Sn-3.0Ag-0.5Cu and Sn-3.8Ag-0.7Cu data, providing steady-state strain rates (ϵ) as a function of stress (σ) and absolute temperature (T) using the equation:

$$\dot{\varepsilon} = A\left(\frac{\sigma}{E}\right)^n \exp\left(-\frac{Q}{RT}\right) \tag{2.8}$$

Where $A = 1.37 \times 10^{46}$, $n = 13.2$, $Q = 61000\,$J/mol, and E(MPa) $= 76087 - 109 \times T$. The model aligns with Kariya and Neu et al.'s 2001 datasets, deviating slightly at $-55°$C and strain rates above 10^{-3}/sec, and does not fit Schubert's data [22].

A comprehensive review of available SAC data by the **National Institute of Standards and Technology in 2004 [23]** considered datasheets from Schubert et al. [33], Kariya et al. [18], and Neu et al. [24]. Regression analysis for a power-law breakdown creep model covered temperatures from $-55°$C to $150°$C, stresses from $2\ MPa$ to $100\ MPa$, and strain rates from 3.8x$10^{-9}/sec$ to 1x$10^{-3}/sec$. The constants for the sinh law derived from this analysis are presented in Table 2.6

Table 2.6: Creep model parameters by NIST [23].

Solder Alloy	A (S^{-1})	α MPa^{-1}	n	Q J/mole
SAC	7.925×10^5	0.0356	6	67900

Pang et al. [30] conducted constant load creep tests at temperatures of 25°C, 75°C, and 125°C, with varying stress levels (2 to 40 MPa). They plotted the steady-state creep strain rate against stress and fitted a curve to the data using a hyperbolic-sine function (Eq. 2.2). The constants for the creep model of the 95.5Sn-3.8Ag-0.7Cu alloy are provided in Table 2.7 [30].

Table 2.7: Creep model parameters by Pang [30].

Solder Alloy	A (S^{-1})	α MPa^{-1}	n	Q eV
Sn95.5Ag3.8Cu0.7	501.3	0.0316	4.96	0.47

Lau et al. [19] employed the same equation as Pang et al., and their constants are: $A = 4.41 \times 10^5$, $n = 4.2$, and $Q = 45000\,$J/mol [19].

In 2003, **Wiese et al. [45]** presented constitutive models for eutectic SnAgCu solders, examining flip-chip solder joints, pin-through-hole solder joints, and standard bulk solder specimens. Constant-load creep tests were conducted at temperatures from 5°C to 70°C, covering strain rates between 10^{-10} and 10^{-3}. Microstructural analysis involved metallographic sectioning, optical microscopy, and SEM microprobe analysis. The proposed creep model for SnAg4Cu0.5 is a power-law equation given by

$$\dot{\varepsilon} = A(\sigma)^n e^{-\frac{Q}{RT}} \tag{2.9}$$

with constants $A = 2 \times 10^{-21}$, $n = 18$, and $Q = 83.1$ kJ/mole [23].

Syed 2001[38] and **2004[39]** proposed a life prediction law for SnAgCu solder joints using a step-by-step approach and a "sinh" type constitutive relation. The "Monkmann-Grant" equation (Equation 2.10) relates time to rupture (t_r) to steady-state creep strain rate ($\dot{\varepsilon}_{cr}$) during a test, assuming constant stress:

$$t_r = \frac{\varepsilon_f}{\dot{\varepsilon}_{cr}} \tag{2.10}$$

The constant ε_f represents the 'creep ductility' or the strain at the onset of failure.

For cyclic stresses, the time-fraction rule (Equation 2.11) estimates rupture time (N_f) using the sum of time fractions at different stress levels:

$$N_f \sum_{i=1}^{n} \frac{\Delta t_i}{t_{ri}} = 1 \tag{2.11}$$

where, N_f is the number of repetitions or cycles to failure, n is the number of steps within a cycle, δt_i is the time spent at stress level σ_i within a cycle, and t_{ri} is the rupture time for stress level σ_i.

Using the Monkmann-Grant equality, this equation becomes (Equation 2.12):

$$N_f \left(\sum_{i=1}^{n} \frac{\Delta t_i \cdot \dot{\varepsilon}_{cri}}{\varepsilon_f} \right) = 1 \tag{2.12}$$

Further simplifying to (Equation 2.13):

$$N_f = (C' \cdot \varepsilon_{acc})^{-1} \tag{2.13}$$

where N_f is cycles to failure, ε_{acc} is accumulated creep strain per cycle, and C' is the inverse of creep ductility.

A fracture mechanics approach [40] yields an energy density-based life prediction model (Eq. 2.14):

$$N_f = (W' \cdot w_{acc})^{-1} \tag{2.14}$$

where w_{acc} is accumulated creep energy density per cycle and W' is creep energy density for failure.

Syed applied these models to SAC solder using ANSYS, determining constants for both double power law and hyperbolic constitutive equations. The resulting life prediction models (Eqs. 2.15 and 2.16) show a good fit to the data:

Using accumulated creep strain:

$$N_f = (0.0513 \cdot \varepsilon_{acc})^{-1} \tag{2.15}$$

Using creep energy density:

$$N_f = (0.0019 \cdot w_{acc})^{-1} \tag{2.16}$$

This emphasizes the model's dependence on deformation constants (C' and W') used in material behavior simulation [39].

Schubert et al. [33] extensively discuss life prediction models for SnPb(Ag) and SnAgCu solder joints under thermal cycling conditions. Their study explores the impact of different solder alloys and package types on fatigue life, considering various temperature cycling conditions.

When solder alloys are above 0.5 times of their melting point temperature, Schubert et al. [33] suggest that creep processes dominate deformation kinetics. Solder joint failure involves complex mechanisms such as grain/phase coarsening, grain boundary sliding, matrix creep, micro-void formation, and linking, leading to crack initiation and propagation. Notably, SnAgCu solder joints exhibit less microstructure coarsening during the damage accumulation process.

The authors propose life prediction models based on two damage parameters: accumulated creep strains and viscoplastic strain energy density for both SnPb(Ag) and SnAgCu solder joints. These models provide accurate predictions within 2X of actual life measurements.

For the empirical model used by the authors, mean cycles to failure (N_f) are estimated using a "Coffin-Manson" type relation:

$$N_f = \Theta_1 \left(\epsilon_{cr}^{acc}\right)^{-c_1} \tag{2.17}$$

where ϵ_{cr}^{acc} is the actual minimum (or average) over the path of the local maximum equivalent creep strain.

Additionally, a viscoplastic strain energy-based method is presented:

$$N_f = \Theta_2 \left(\Delta W_{cr}^{acc}\right)^{-c_2} \tag{2.18}$$

Here, ΔW_{cr}^{acc} is the actual minimum (or average) over the path of the local maximum of viscoplastic strain energy density.

The constants for Schubert et al.'s life prediction model are summarized in Table 2.8.

Table 2.8: Life prediction model constants by Spraul et al. [37].

Solder Alloy	Θ_1	C_1	Θ_2	C_2
Sn37Pb	537.15	1.0722	12.213	1.1361

In the context of fatigue life of solder joints, **Vasu Vasudevan and Xuejun Fan [42]** proposed a set of AF model constants based on the Norris-Landzberg equation and validated using thermal cycle tests. The thermal cycling experimental data from various sources, with different types of packages, have been used for determining the model fit. The model fit to experimental data was excellent, with less than 6% error. The analysis showed that the acceleration factor (AF) model is not significantly different from the Sn/Pb model, and the proposed model provides the best fit to the experimental results.

Solder joint fatigue is considered low-cycle failure, and almost all lifetime prediction models originate from the Coffin-Manson equation:

$$N(\Delta\varepsilon_p)^n = C \tag{2.19}$$

where N is the number of cycles to failure, $\Delta\varepsilon_p$ is the plastic strain range per cycle, n is an empirical material constant, and C is a proportionality factor.

The acceleration factor (AF) can be defined based on Equation (2.19) as follows:

$$AF = \frac{N_1}{N_2} = \left(\frac{\Delta\varepsilon_p^1}{\Delta\varepsilon_p^2}\right)^{-n} \tag{2.20}$$

Norris and Landzberg [25] assumed that the plastic strain range is proportional to the temperature excursion range, and introduced factors to account for the effects of temperature-cycling frequency f and the maximum temperature T_{max} of the solder material. The AF equation obtains the form:

$$\begin{aligned} AF =& \frac{N_{\text{field}}}{N_{\text{test}}} \\ =& \left(\frac{f_{\text{field}}}{f_{\text{test}}}\right)^{-m} \left(\frac{\Delta T_{\text{field}}}{\Delta T_{\text{test}}}\right)^{-n} \exp\left(\frac{E_a}{k}\left(\frac{1}{T_{\text{max, field}}} - \frac{1}{T_{\text{max, test}}}\right)\right) \end{aligned} \tag{2.21}$$

where "field" and "test" denote the field and the test condition, respectively. The Norris-Landzberg model can be used to compare two different field or test conditions. For SnPb eutectic solder, $m = 1/3$, $n = 1.9$, and $\frac{E_a}{k} = 1414$.

Pan et al. [29] obtained a new set of parameters based on their experiments for SnAgCu soldered components:

$$m = 0.136, \quad n = 2.65, \quad \text{and} \quad \frac{E_a}{k} = 2185 \tag{2.22}$$

Engelmaier [8] proposed an analytical form of the Coffin-Manson's equation:

$$N(x\%) = \frac{1}{2}\left(\frac{2\varepsilon_c}{\varepsilon_p}\right)^{-\frac{1}{c}}\left(\frac{\ln(1-0.01x)}{\ln 0.5}\right)^{\frac{1}{\beta}} \tag{2.23}$$

Salmela [31] recalibrated the Engelmaier model and **Osterman [27]** presented a new set of constants based on the Engelmaier model for SnAgCu solder.

An extension to the original Coffin-Manson equation is to use the plastic strain energy density instead of the plastic strain range to represent the cumulative damage on solder joint.:

$$N(\Delta W_p)^m = C' \tag{2.24}$$

$$AF = \left(\frac{\Delta W_p^1}{\Delta W_p^2}\right)^m \tag{2.25}$$

Using Garafola's Hyperbolic Creep Model (Equation 2.2), **Rudi Hechfellner and his co-authors [13]** calculated the fatigue life of Philips Lumileds LUXEON Rebel LED Carrier on a Metal Core PCB substrate. They utilized Schubert's constant (Table 2.5) and the modified Norris-Landzberg equation (Equation 2.21).

$$AF = \left(\frac{\Delta T_\mathrm{t}}{\Delta T_\mathrm{o}}\right)^a \left(\frac{\Delta T_\mathrm{t}}{\Delta T_\mathrm{o}}\right)^b \exp\left(c\left(\frac{1}{T_{\text{max, o}}} - \frac{1}{T_{\text{max, t}}}\right)\right) \tag{2.26}$$

where, AF = acceleration factor, N = thermal fatigue life, ΔT = temperature difference, t = dwell time (min), T_{max} = maximum cycle temperature, o, t = operating or test condition, a, b, c = coefficients, 1.84, 0.11, 6167 respectively.

Initially, the modified Engelmaier equation [5] was used to obtain the strain range:

$$\Delta\gamma = C\left(\frac{L_D}{h_s}\right)\Delta\alpha\Delta T \tag{2.27}$$

where C is a correlation factor dependent(C $= 1/\sqrt{2}$ for leadless components [14]), L_D is the diagonal distance of the solder to the neutral, α is CTE, ΔT is the temperature cycle, h_s is solder joint height.

The shear force applied to the solder joint was calculated using the following formula:[5]

$$|\alpha_1 - \alpha_2|\Delta T \cdot L = F\left(\frac{L}{E_1 A_1} + \frac{L}{E_2 A_2} + \frac{h_s}{A_s G_s} + \frac{h_c}{A_c G_c} + \frac{2-v}{9 G_b a}\right) \tag{2.28}$$

where F is the shear force, L is length, E is the elastic modulus, A is the area, h is thickness, G is the shear modulus, and a is the edge length of the bond pad. Subscripts 1, 2, s, c, and b refer to component, board, solder joint, bond pad, and board, respectively.

The formula takes into consideration the foundation stiffness and both the shear and axial loads.

Finally, the strain energy dissipated by the solder joint was determined:

$$\Delta W = 0.5\Delta\gamma\frac{F}{A_s} \tag{2.29}$$

The cycles to failure (N_{50}) were then calculated using the energy-based fatigue models for SAC developed by Syed (Equation 2.16):

$$N_f = (0.0019 \cdot w_{\text{acc}})^{-1} \tag{2.30}$$

Models were constructed for the LUXEON Rebel LED on a total of 15 different substrates with Thermo Mechanical Analyses performed on nine for three different environments using the SAC405 solder alloy. The results from the FEA analysis indicated an excellent correlation between the models and experimental data.

Luiten [20] employed a 1D analytical model to analyze the impact of cycle time on lead-free solder fatigue in LEDs.

Thermal mismatch occurs in an LED package (length L, Coefficient of Thermal Expansion α_{led}) on a printed circuit board (CTE α_{board}). Upon temperature change ΔT, the board and LED package expand or contract according to their CTE, creating a thermal mismatch. The solder joint, being the most flexible part, bridges this mismatch through shear deformation. Figure 2.4 illustrates the thermal mismatch and deformed solder joint.

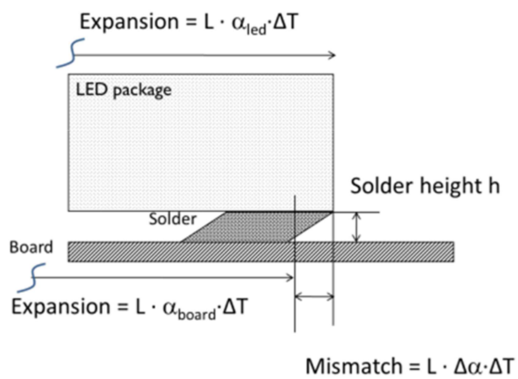


Figure 2.4: Thermal mismatch loads and deformation in the solder joint[20]

At short time scales, solder joint deformation is elastic. Over time, plastic deformation occurs, introducing solder creep. The creep strain rate is stress and temperature dependent.

The thermal fatigue cycle involves cyclic loading, creep, and stress relaxation in the solder joint. The shear stress (σ) is proportional to the elastic strain (ϵ_e) given by $\sigma = G\epsilon_e$. Solder creep leads to stress relaxation, and the joint eventually becomes stress-free. The process repeats with temperature cycles.

Dr. Blattau [34] conducted groundbreaking research on a light-emitting diode (LED) soldered with SAC305 solder on a metal core printed circuit board. Figure 2.5 illustrates the LED considered in their study.

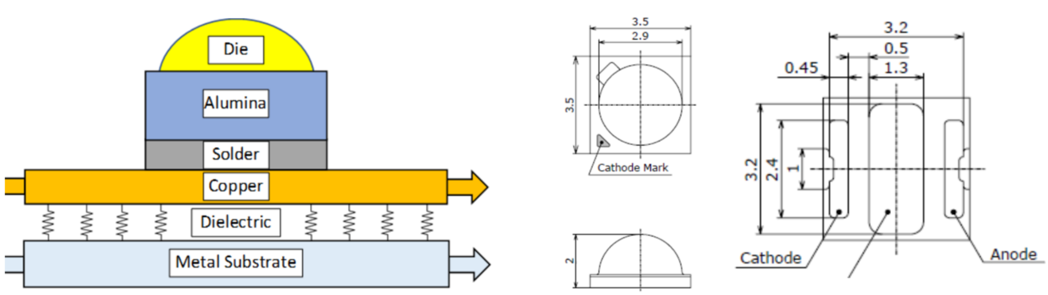


Figure 2.5: LED considered by M. Serebreni & N. Blattau [34]

Using the constants for different dielectric materials of the PCB defined in Table 2.9 and material properties for the LED substrate in Table 2.10, they calculated the elastic modulus of the dielectric above its glass transition temperature (T_g) following the formula:

$$E_{\text{delT}} = a \cdot E_{\text{mold}} \cdot \left(b + \tanh\left(\frac{T_{\text{max}} - T_g}{S} \right) \right) \tag{2.31}$$

Table 2.9: Constants used in Equation 2.31 for three dielectric materials [34]

	HT	MP	CML
a	0.5	0.425	0.425
b	1.2	1.4	1.5
S (°C)	55	25	35
T_g (°C)	140	85	85

Table 2.10: Measured LED ceramic substrate properties

Material	E (GPa)	α (ppm/°C)	ν (Poisson ratio)
Aluminum Nitride	330	4.9	0.27
Silicon Nitride Si_3N_4	166	2.7	0.25

They calculated the fatigue life using the strain energy model, considering stiffness. The stiffness formula employed is

$$K_{\text{eff}} = \frac{1}{\left(\frac{1}{K_{\text{c}}} + \frac{1}{K_{\text{j}}} + \frac{1}{K_{\text{b}}} + \frac{1}{K_{\text{pa}}} + \frac{1}{K_{\text{pf}}}\right)} \tag{2.32}$$

Here, the individual stiffness components are defined as follows:

$$K_{\text{c}} = \frac{C_{\text{thickness}} \cdot C_{\text{width}} \cdot E_{\text{submount}}}{C_{\text{length}}} \tag{2.33}$$

$$K_{\text{j}} = \frac{G_{\text{solder}} \cdot C_{\text{thickness pad}} \cdot E_{\text{pad}}}{h_{\text{solder}}} \tag{2.34}$$

$$K_{\text{pa}} = \frac{2 \cdot E_{\text{substrate}} \cdot t_{\text{substrate pad}} \cdot C_{\text{width}}}{C_{\text{length}}} \tag{2.35}$$

Since sub-mount LEDs have two or three solder pads, the elastic modulus of solder is area-weighted by the solder area encompassed by anode, cathode, and thermal pads.
The stiffness of the elastic foundation can be represented in various ways, including using the rigid surface foundation on a homogeneous elastic half-space approach:

$$K_{\text{b},1} = \frac{8G \cdot R}{2 - v_{\text{s}}} \tag{2.36}$$

Where $R = \sqrt{\frac{WL}{2}}$ is the radius of the circular rigid loading area, G and v_s are the shear modulus and Poisson ratio of the homogeneous half-space.

The second approach to the stiffness of the elastic dielectric layer is provided by the beam on elastic foundation.

$$K_{\text{b},2} = \frac{0.95 \cdot E_{\text{s}}}{1 - (v_{\text{s}})^2} \cdot \left(\frac{E_{\text{s}} \cdot T \cdot B^4}{(1 - (v_{\text{s}})^2) \cdot EI}\right)^{0.108} \tag{2.37}$$

Where E_s is the temperature-dependent elastic modulus of the dielectric, v_s is the Poisson ratio of the dielectric, B is the copper pad width, E is the elastic modulus of the copper pad, and I is the second moment of inertia of the copper pad.

Stiffness parameter for plate on elastic half-space is describes by equation 2.38.This representation of the relative stiffness parameter, defined by Gorbunov-Posadov and Serebrjanyi [11], provides a measure of the foundation flexibility [21]. Where $K_{\text{b,3}} = 0$ represents a perfectly rigid plate, and $K_{\text{b,3}} = \infty$ represents theoretically a perfectly flexible plate.

$$K_{\text{b},3} = \frac{12\pi(1 - (v_{\text{p}})^2) \cdot R \cdot (T_s)}{(1 - (v_{\text{s}})^2) \cdot E_{\text{p}}} \cdot \left(\frac{a}{t_{\text{p}}}^2\right) \cdot \left(\frac{b}{t_{\text{p}}}^2\right) \tag{2.38}$$

Where a and b are the width and length of the copper layer.
The shear stress is then calculated using the reduced force-displacement compatibility form:

$$\tau_{\mathsf{s}} = \frac{\Delta\alpha \cdot \Delta T \cdot L_{\mathsf{sub}}}{A_{\mathsf{pad}}} \cdot K_{\mathsf{eff}} \tag{2.39}$$

Here, L_{sub} is the half-diagonal length of the ceramic sub-mount, incorporating the distance to the neutral effect during CTE mismatch. The shear strain is calculated from the global CTE mismatch driven by metal substrate and ceramic sub-mount:

$$\gamma_{\mathsf{s}} = \frac{\Delta\alpha \cdot \Delta T \cdot L_{\mathsf{sub}}}{h_{\mathsf{solder}}} \tag{2.40}$$

The Modified Norris-Landzberg equation is used to correlate plastic strain to creep strain under test conditions[34][41]:

$$\bar{\gamma}_{\mathsf{max}}^{\mathsf{dwell}} = \bar{\gamma}_{\mathsf{max}} \left(\frac{T_{\mathsf{dwell}}}{T_{\mathsf{test}}}\right)^{-n} \exp\left(\frac{E_{\mathsf{a}}}{k}\left(\frac{1}{T_{\mathsf{max,f}}} - \frac{1}{T_{\mathsf{max,t}}}\right)\right) \tag{2.41}$$

The strain energy is calculated using equation 2.42, assuming the hysteresis loop is roughly equilateral:

$$W = \frac{1}{2} \cdot \gamma_{\mathsf{s}} \cdot \tau_{\mathsf{s}} \tag{2.42}$$

A deterministic component is introduced using a 2-parameter Weibull distribution (equation 2.43), enabling the representation of predicted cycles to failure with a characteristic life of a 63.2 percent failure probability:

$$N_f(63.2\%) = C_1 \cdot (W_{\mathsf{max}})^{-C_2} \cdot \left(\frac{\ln(1 - 0.01\text{x}63.2)}{\ln(0.05)}\right)^{\frac{1}{\beta}} \tag{2.43}$$

The two-parameter Weibull distribution fits solder joints' fatigue data well in mildly accelerated test conditions. The relationship between cycles to failure and strain energy density is based on work by Sayed.[40]

Shaygi [35] preferred to use Ansys FEA simulation to get average creep strain on the solder which can be used to calculate fatigue life. His simulated LED is shown on Figure 2.6

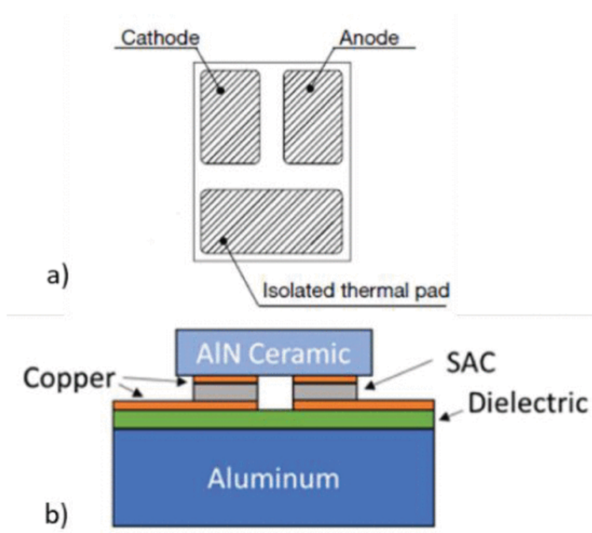


Figure 2.6: (a) Solder footprint of a 3-pad LED [35] (b) Simplified geometry of the structure [35]

Fan's work [9], building upon previous research, focuses on optimizing the reliability of wafer-level chip scale LED packages soldered with SAC305 alloy on aluminum substrates. The study employs finite element analysis, considering factors such as substrate type, solder thickness, void ratio, and PCB substrate influence. Insights gained contribute to enhancing the fatigue damage resistance of LED packaging technology. He considered the deformation resistance, s, can be regarded as proportional to the equivalent stress σ, as shown in Equation 2.44:

$$\sigma = c \cdot s, \quad c < 1 \tag{2.44}$$

where c is defined as:

$$c = \frac{1}{\zeta} \cdot \sinh^{-1}\left(\left(\frac{\dot{\epsilon}_p}{A} \cdot \exp\left(\frac{Q}{RT}\right)\right)^m\right) \tag{2.45}$$

where $\dot{\epsilon}_p$ is the plastic strain rate, A is the pre-exponential factor, Q the activation energy, m is the strain rate sensitivity, ζ is the stress multiplier, R is the universal gas constant, and T is the absolute temperature. The plastic strain rate $\dot{\epsilon}_p$ can be described as:

$$\dot{\epsilon}_p = A \cdot \exp\left(-\frac{Q}{R\Theta}\right) \cdot \left(\sinh(\zeta \cdot \frac{\sigma}{s})\right)^{\frac{1}{m}} \tag{2.46}$$

The evolution equation is given by:

$$s = h_0 \cdot \left|1 - \left(\frac{s}{s^*}\right)\right|^a \cdot \text{sign}\left(1 - \frac{s}{s^*}\right) \cdot \dot{\epsilon}_p, \quad a > 1 \tag{2.47}$$

$$s^* = \widehat{s} \cdot \left(\left(\frac{\dot{\epsilon}_p}{A} \cdot \exp\left(\frac{Q}{RT}\right)\right)^n\right) \tag{2.48}$$

Where h_0 is the hardening/softening constant, a is the strain rate sensitivity of hardening/softening, s^* is the saturation value of s, $\hat{s}$ is the coefficient for the saturation value of deformation resistance, and n is the strain rate sensitivity. Parameters used in the Anand model are related to material properties, such as A, Q, ζ, m, h_0, $\hat{s}$, n, a, and S_0. S_0 is the initial value of the deformation resistance. These parameters for SAC305 are listed in Table 2.11.

Table 2.11: Anand Model Parameter for SAC305 [15]

Anand's Constants	Units	Value	Description
S_0	MPa	45.9	Initial value of deformation resistance
Q/R	1/K	7460	Q = Activation Energy R = Universal gas constant
A	s^{-1}	5.87×10^6	Pre-exponential factor
ζ	-	2	Stress Multiplier
m	-	0.0942	Strain rate sensitivity of stress
h_0	MPa	9350	Hardening/Softening constant
scap	MPa	58.3	Coefficient for deformation resistance saturation value
n	-	0.015	Strain rate sensitivity of saturation (deformation resistance) value
a	-	1.5	Strain rate sensitivity of hardening/softening

He then used ANSYS software to do the FEA simulation, considering Poisson's ratio and CTE of SAC305 solder to be 0.32 and 19.1 ppm/K.

These investigations collectively contribute to advancing the understanding of solder joint reliability in LED applications, offering valuable insights into the factors that influence fatigue, thermal cycling, and thermal shock. The integration of experimental testing and numerical simulations further refines reliability prediction models applicable to diverse LED packages and applications.

Nevertheless, none of the existing studies appears to effectively predict solder joint reliability, particularly for bottom-terminated automotive LED components on IMS/metal-core printed circuit boards. Current approaches involve prediction followed by the use of calibration factors to align the predicted result with the actually lab-measured results or are constrained to specific geometries. Moreover, these methodologies are contingent upon specific soldering geometries at the bottom of the LED component.

This thesis work aims to address this gap by concentrating on the simulation of the Bottom-Terminated Component (BTC) model for LEDs on an IMS PCB board using ANSYS Sherlock. The results obtained from these simulations are then compared with real laboratory-measured results, and a comprehensive analysis of the findings is conducted.

2.3 Weibull Analysis

The Weibull distribution is a widely used probability distribution for modeling the reliability and lifetime of electronic components, including solder joints in Light-Emitting Diodes (LEDs). It is characterized by two parameters: the shape parameter (β) and the scale parameter (η). The probability density function (PDF) of the Weibull distribution and its cumulative distribution function (CDF) are given by:[44][43][12]

$$f(t;\beta,\eta) = \frac{\beta}{\eta}\left(\frac{t}{\eta}\right)^{\beta-1}\exp\left[-\left(\frac{t}{\eta}\right)^{\beta}\right] \tag{2.49}$$

$$F(t;\beta,\eta) = 1-\exp\left[-\left(\frac{t}{\eta}\right)^{\beta}\right] \tag{2.50}$$

where:

- t is the time to failure,
- β is the shape parameter,
- η is the scale parameter.

The shape parameter influences the failure behavior:

- If $\beta < 1$, the failure rate decreases over time (infant mortality).
- If $\beta = 1$, the failure rate is constant over time (random failures).
- If $\beta > 1$, the failure rate increases over time (wear-out).

The scale parameter represents the characteristic life of the distribution.

In this thesis, the Weibull distribution is employed to analyze solder fatigue in various models of Light-Emitting Diodes. The distribution offers insights into the reliability and failure characteristics of solder joints subjected to thermal cycling.

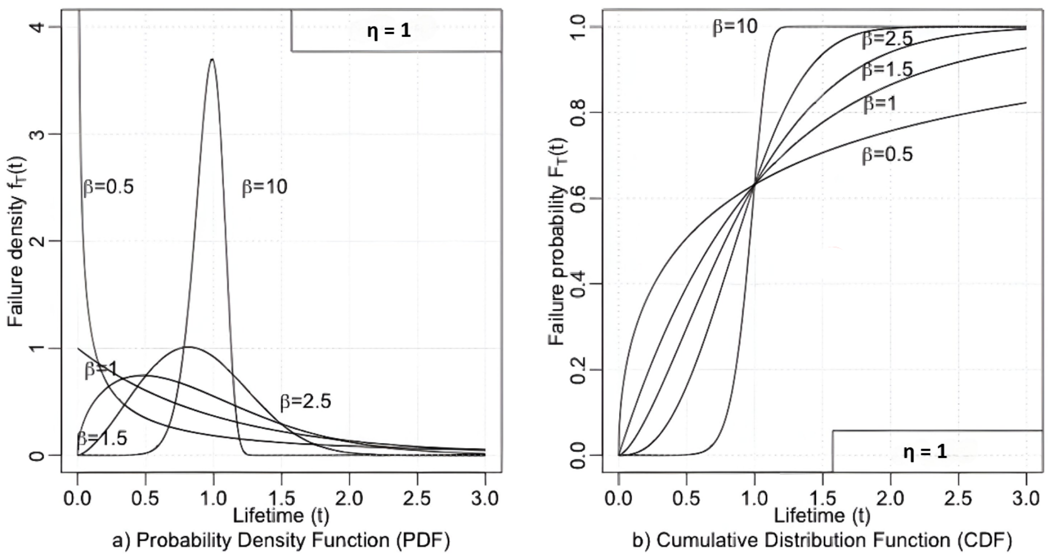


Figure 2.7: Weibull Curves example.
The original image was sourced from[4] and has been modified.

Sample Weibull curves in Figure 2.7 display Weibull distribution functions for distinct shape parameters (β). By adjusting the shape parameter, the Weibull function can approximate the form of distributions such as the Exponential ($\beta = 1$) or Normal ($\beta = 10$, $\beta = 2.5$).

2.3.1 Modifying Ansys Sherlock Simulation Results

In the solder fatigue simulation of Ansys Sherlock, it produces a Weibull curve with a shape factor (β) of 3. However, practical Weibull curves are often based on a different shape factor, necessitating the need to adjust the simulation graph data for meaningful comparison.

It is possible to obtain the value of the scale parameter (λ) from the following cumulative distribution function (CDF) equation:

$$F(t;\beta,\eta) = 1 - \exp\left[-\left(\frac{t}{\eta}\right)^{\beta}\right] \tag{2.51}$$

Thus,

$$\lambda \text{ or } \eta = \frac{t}{\sqrt[3]{\log(1 - F(t;\beta,\eta))}} \tag{2.52}$$

This λ can be used in the following equation to obtain graphical data for different Weibull factors:

$$P(t;\beta,\eta) = \left(1 - \exp\left(-\left(\frac{t}{\lambda}\right)^{\beta}\right)\right) \tag{2.53}$$

However, Ansys Sherlock produces Weibull curves where the cumulative distribution function (CDF) on the y-axis is multiplied by 100 to represent the probability of failure from 0 to 100 .The above equations, on the other hand, work for probabilities in the range 0 to 1.

Therefore, a Python function can be created to preprocess the graph data produced by Ansys Sherlock. The function takes the graphical data in Pandas dataframe format and the desired Weibull factor (n) as parameters and returns the modified dataframe for the specified Weibull factor or shape factor n. The function also excludes the initial row since it contains only zero values by default, and logarithms are undefined for zero.

```python
def Preprocess(dataframe, n):
    modifi = dataframe.drop(0, axis=0).reset_index(drop=True)

    # Check for zero values before computing cbrt
    zero_mask = (1 - (modifi['Life Prediction'] / 100)) == 0
    modifi = modifi[~zero_mask]

    modifi['lambda_val'] = (modifi['Lifetime (hrs)'])/np.cbrt(-np.log(1-(modifi['Life Prediction']/100)))

    # Check for zero values before computing exponential
    zero_mask = modifi['lambda_val'] == 0
    modifi = modifi[~zero_mask]

    lambda_val = modifi['lambda_val'].iloc[0]
    modifi['Life Prediction'] = (1 - np.exp(-((modifi['Lifetime (hrs)']/ lambda_val) ** n)))*100

    return modifi
```

2.4 Softwares

2.4.1 An Overview of ANSYS Sherlock

ANSYS Sherlock is a specialized software tool developed by ANSYS, Inc. to help engineers in the electronics industry predict the reliability of their electronic systems. It is designed to evaluate various factors affecting the reliability of electronic components and assemblies. One of its primary applications is the prediction of solder fatigue in electronic assemblies.

2.4.1.1 Key Features

Sherlock offers a range of features, including:

- **Materials Database**: The software provides access to a comprehensive materials database, offering detailed information on electronic components and materials commonly used in electronics manufacturing.
- **Predictive Analysis**: Using advanced predictive analytics, Sherlock assesses the reliability of electronic systems under different operating conditions and environmental stresses.
- **Finite Element Analysis (FEA)**: Sherlock employs FEA to simulate the mechanical and thermal behavior of electronic components and solder joints, allowing engineers to understand the stress and strain distribution.
- **Solder Fatigue Analysis**: Sherlock includes tools to model and predict solder joint fatigue, a frequent cause of failure in electronic assemblies .

2.4.2 Python (Jupyter Notebook)

Python is a versatile and widely-used programming language. When used in Jupyter Notebook, it provides an interactive coding environment conducive to data analysis, manipulation, and visualization. The Jupyter Notebook's format supports the creation of documents that combine code, explanations, and visualizations, making it a popular choice for researchers and data scientists.

2.4.3 Microsoft Office PowerPoint

Microsoft PowerPoint is a presentation software application that forms part of the Microsoft Office suite. It is primarily designed for the creation and delivery of slide-based presentations. PowerPoint's intuitive user interface, multimedia capabilities, and transition effects make it a preferred tool for generating professional presentations and documentation.

Chapter 3

Theoretical Analysis

This chapter performs a theoretical analysis of how ANSYS Sherlock predicts solder fatigue life.

ANSYS Sherlock's theory references do not provide a theoretical overview for explaining the prediction of solder fatigue life for the BTC model. However, it provides the theory for CC and LCCC model components. For reference, Figure 3.1 shows the solder differences between CC and BTC models. It looks similar, and ANSYS Sherlock doesn't consider the Meniscus for solder fatigue life calculation of the CC & LCCC models. So, it makes sense to make the decision that ANSYS Sherlock follows the similiar procedure for the BTC model as followed for the CC model. According to ANSYS Sherlock's theory references, Sherlock also follows the same theory as CC Model for LCCC model solder. The only difference is that, because LCCC components are more like square-shaped, while considering the component length in calculating the total displacement due to thermal expansion, ANSYS considers the diagonal component length for the LCCC model instead of the normal component length. SAC305 solder is recalled as the used solder for reference; for INNOLOT solder, it will be the same, only material properties and material constants will be different.

The first section of this chapter will go through the theory for CC & LCCC models, and later, there will be an approach to link it to calculate the solder fatigue of the BTC model in second section.

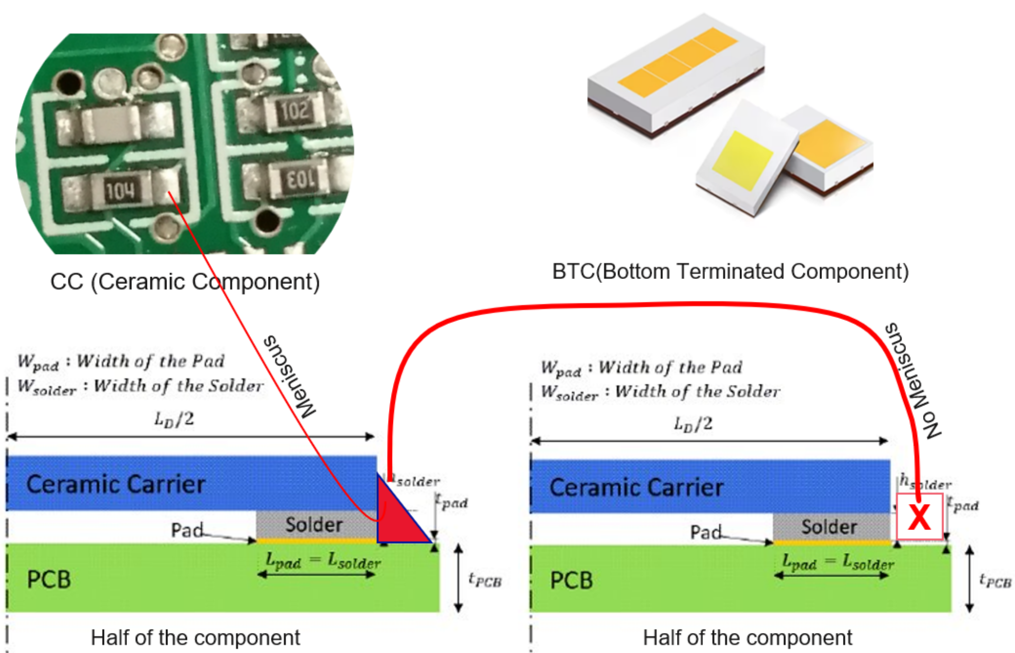


Figure 3.1: Differences Between CC & BTC Components [32][16][2]

3.1 Solder Fatigue Theory for CC & LCCC Components

A simple procedure of how ANSYS Sherlock calculates solder fatigue life for CC & LCCC models is shown in Figure 3.2.

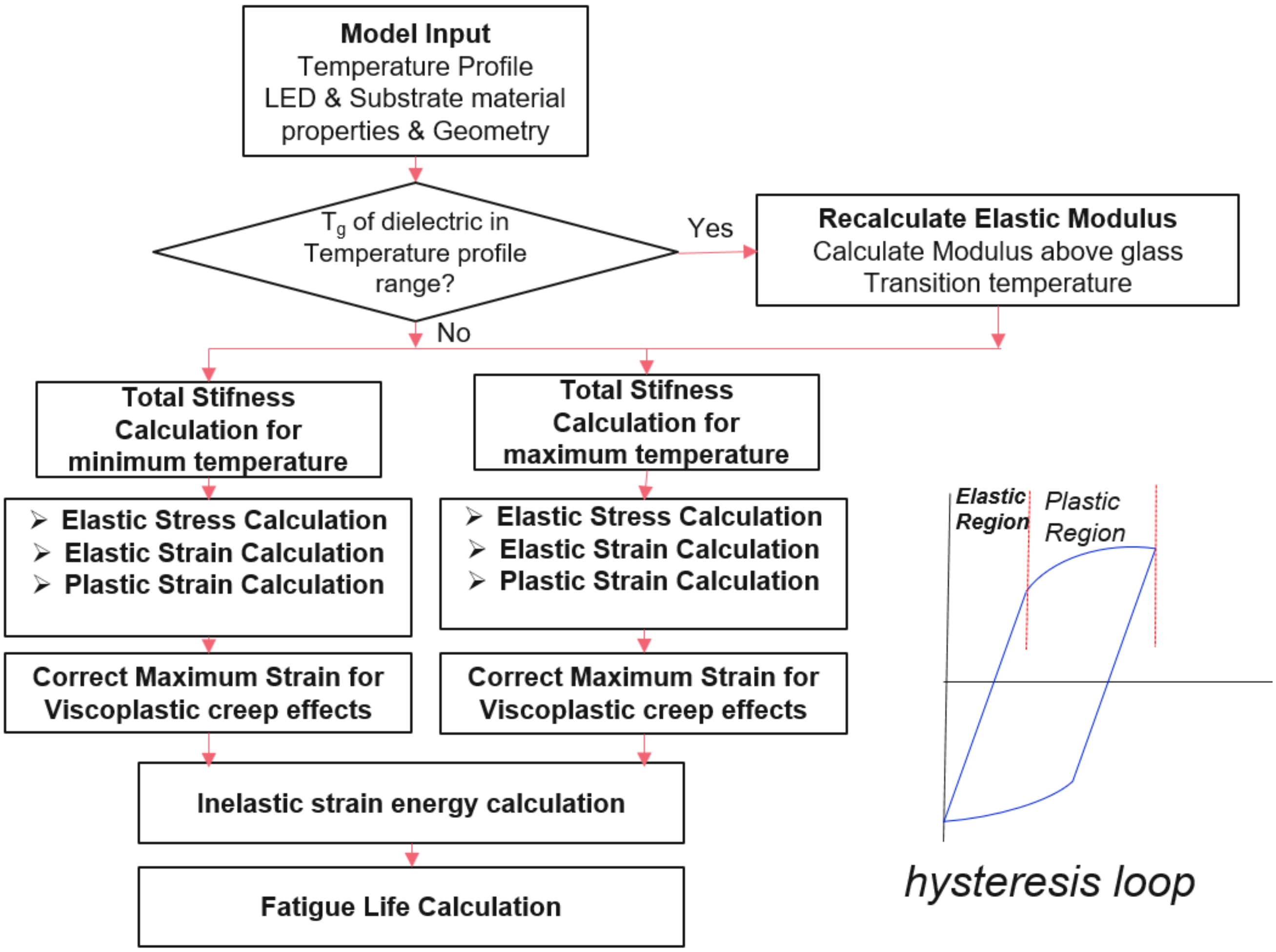


Figure 3.2: Flow Diagram of Solder Fatigue Life Calculation for CC & LCCC Solder Model in ANSYS Sherlock

Because the shear modulus (also elastic modulus) of the solder SAC305 is highly dependent on temperature, in Sherlock, the calculation is done in two parts, for hot parts and cold parts.

From Figure 3.2, first Sherlock takes the temperature profile and LED & PCB substrate properties, stack up data, and geometrical inputs, and it checks if the glass transition temperature (T_g) of the dielectric used in the PCB is within the temperature range or not. If it is within the temperature range, it recalculates the elastic modulus of the dielectric above the glass transition temperature (T_g). ANSYS Sherlock calculates the total stiffness for a solder (it only does it for one solder, ignoring other solders). It then calculates the elastic region stress and calculates the strain induced in the elastic region. Then it calculates the strain for the plastic region. Then it corrects the maximum strain considering the viscoplastic creep happens at a higher temperature. When this is done for both hot and cold temperature, inelastic strain energy is calculated considering both of the maximum strains. Then fatigue life is calculated from the inelastic strain energy.

Table 3.1: Explanation of used Symbols

Symbolic Representation	**Explanation**	**Source**
L_{comp}	Length of the component	Component datasheet
W_{comp}	Width of the component	Component datasheet
t_{comp}	Thickness of the component	Datasheet
E_{comp}	Elastic modulus of the component	Component datasheet / Material datasheet
$\frac{L_d}{2} = 0.5\sqrt{(L_{\text{comp}})^2 + (L_{\text{comp}})^2}$	Half diagonal length of LCCC components	Calculation
$L_d/2$	Normal half length of CC components	Calculation
L_{pad}	Length of the copper pad below the solder	Component datasheet
W_{pad}	Width of the copper pad below the solder	Component datasheet
t_{pad}	Thickness of the copper pad (equivalent to the thickness of the signal layer in PCB)	Set up dependent / Datasheet
$L_{\text{joint}} = C_1 \cdot L_{\text{pad}}$	Length of the solder joint, C_1 = multiplication factor	Component datasheet / Calculation
$W_{\text{joint}} = C_2 \cdot W_{\text{pad}}$	Width of the solder joint, C_2 = multiplication factor	Component datasheet / Calculation
$h_{\text{solder}} = t_{\text{solder}}$	Height of the solder joint	Setup dependent
A_{joint}	Area of the solder joint	Calculation
E_{joint}	Elastic modulus of solder joint	ANSYS Sherlock / Publications
V_{joint}	Poisson's ratio of the solder, 0.36 for SAC305 solder	Material datasheet
G_{joint} or $G_{\text{solder}} = \frac{E_{\text{joint}}}{2(1+V_{\text{joint}})}$	Shear modulus of the solder joint (Temperature dependent)	Calculation
E_{pcb}	Elastic Modulus (axial) of the PCB	ANSYS Sherlock
V_{pcb}	Poisson's ratio of the PCB (ANSYS Sherlock considers 0.18 for all setups)	ANSYS Sherlock
t_{pcb}	Total Thickness of the PCB	Setup dependent
$G_{\text{pcb}} = \frac{E_{\text{joint}}}{2(1+V_{\text{pcb}})}$	Shear modulus of the PCB	Calculation
CTE_{pcb}	Coefficient of Thermal Expansion (axial) of the PCB	ANSYS Sherlock
V_{pad}	Poisson's ratio of the copper pad on the board side where soldering is done	Material datasheet / ANSYS Sherlock
$G_{\text{pad}} = \frac{E_{\text{joint}}}{2(1+V_{\text{pad}})}$	Shear modulus of the pad at the bottom of the solder where soldering is done	Material datasheet / ANSYS Sherlock

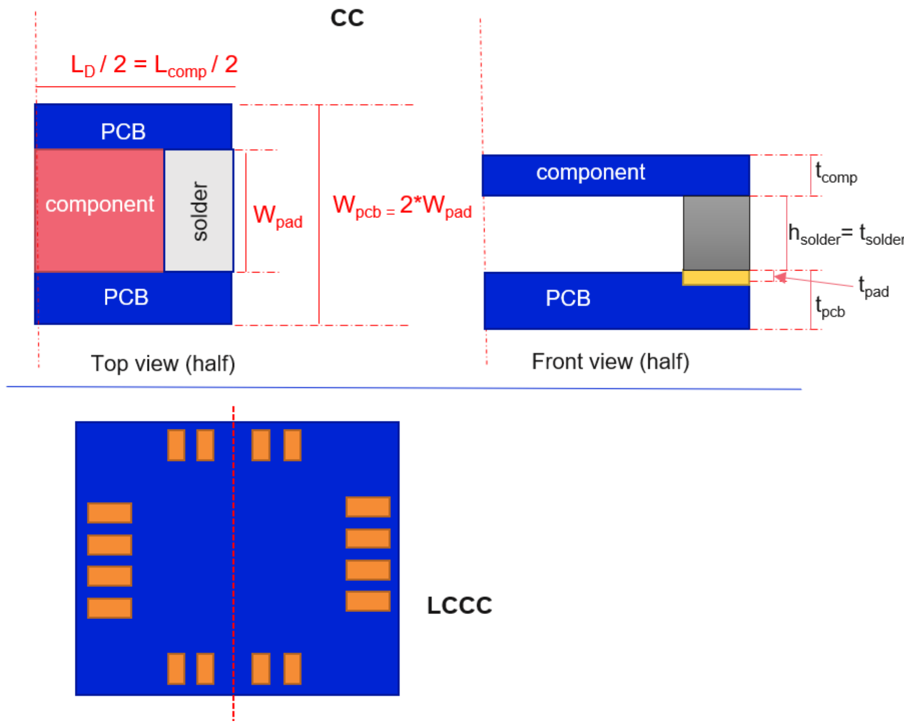


Figure 3.3: CC & LCCC Model, Theoretical considerations

For a visual representation of the solder, component, copper pad, and the PCB, Figure 3.3 relates the symbols with the setup. For clarification, ANSYS Sherlock considers the length of the PCB to be the same as the length of the component, but the width of the PCB is double the width of the component.

3.1.1 Elastic Modulus Above T_g (Glass Transition Temperature Calculation)

To calculate the elastic modulus above the glass transition temperature, equation 3.1 is used (as provided by Sherlock authority in a discussion).

$$\text{Elastic Modulus above } T_g = 0.8 \times \left(\frac{\text{Elastic modulus below } T_g}{2}\right) \times \left(1.5 + \tanh\left(-1 \times \frac{\mathrm{T_{max}} - T_g}{\text{Shape Factor}}\right)\right) \tag{3.1}$$

here $\mathrm{T_{max}}$ is the maximum temperature of the thermal cycle
For Bergquist MP dielectric:

$$T_g = 90^\circ\mathrm{C}$$
$$\text{Elastic Modulus below } T_g = 1220.209894\,\mathrm{MPa}$$
$$\text{Shape Factor} = 25$$

However, a paper published by Dr. Nathan Blattau, R & D Fellow at ANSYS, Inc., the same is done in another formula [4]:

$$E_{\text{dielectric}}(\Delta T) = a \times E_{\text{mold}} \times \left(b + \tanh\left(\frac{\mathrm{T_{max}} - T_g}{S}\right)\right) \tag{3.2}$$

Where S is a shape parameter corresponding to the length of the transition region, T_g is the glass transition temperature of the dielectric, T_{max} is the maximum temperature cycle of the thermal profile, E_{mold} is the measured elastic modulus of the dielectric prior to the glass transition temperature, and a and b are empirical regression constants (for Bergquist MP dielectric, $a = 0.425$, $b = 1.4$, $S = 90°$C). Dr. Blattau considered $T_g = 85°$C for Bergquist MP Dielectric.

However, this shouldn't affect the simulation result, since there is no option to put this calculation result in the Bergquist MP material library in ANSYS Sherlock. The only possible option is to put the T_g in Sherlock, and Sherlock calculates the elastic modulus of the PCB implicitly by itself, taking the above T_g elastic modulus of the dielectric into consideration.

3.1.2 Total Stiffness Calculation

Sherlock calculates the total stiffness for two distinct parts, the hot part denoted as T_{max} and the cold part denoted as T_{cold}. These are determined using the following expressions:

$$T_{\text{max}} = \frac{T_{\text{max}} - T_{\text{mean}}}{2} \tag{3.3}$$

$$T_{\text{cold}} = \frac{T_{\text{mean}} - T_{\text{min}}}{2} \tag{3.4}$$

where,
$T_{\text{mean}} = \frac{T_{\text{max}}+T_{\text{min}}}{2}$

T_{mean} is actually the reference temperature. In the older version of ANSYS Sherlock, there was an available option to set this temperature manually, as found in the ANSYS Theory reference on pages 50 and 51. However, in Sherlock 2022 R2 Version, this option is hidden.

The reference temperature is expected to be the temperature at which all initial dimensions were measured (for PCB, component, solder, solder pad, etc.). However, it is not yet clear why ANSYS Sherlock took the average. It might be for simplicity, while the reference temperature can differ.

In ANSYS Sherlock Theory, The temperatures are given in degrees Celsius, with T_{min} being the minimum temperature and T_{max} being the maximum temperature of the thermal cycle.

For the total stiffness calculation, axial component stiffness, axial PCB stiffness, a single solder shear stiffness, shear stiffness of the pad at the bottom of the solder, and shear stiffness of the foundation are considered.

Since the combined total stiffness for the cold part is the same as the hot part, only the combined stiffness for the hot part is discussed here. The only difference is that for the hot part, the shear modulus of the solder at the maximum thermal cycling temperature is considered, while for the cold part, the shear modulus at the minimum thermal cycling temperature is considered.

The shear modulus at the maximum temperature and minimum temperature can be calculated with the following formulas:

$$G_{\text{T}_{\text{max}}}^{\text{solder}} = \frac{E_{\text{joint}}^{\text{max}}}{2(1 + V_{\text{solder}})} \tag{3.5}$$

$$G_{\text{T}_{\text{min}}}^{\text{solder}} = \frac{E_{\text{joint}}^{\text{min}}}{2(1 + V_{\text{solder}})} \tag{3.6}$$

where $G_{\text{T}_{\text{max}}}^{\text{solder}}$ = shear modulus of the solder at maximum thermal cycling temperature

$G_{\text{T}_{\text{min}}}^{\text{solder}}$ = shear modulus of the solder at minimum thermal cycling temperature

$E_{\text{joint}}^{\text{max}}$ = Elastic modulus of the solder at maximum thermal cycling temperature

$E_{\text{joint}}^{\text{min}}$ = Elastic modulus of the solder at minimum thermal cycling temperature

V_{solder} = Poisson's ratio of the solder

Total combined stiffness formula used by Sherlock is (for hot part)

$$K_{\Delta T_{\text{hot}}}^{\text{combined}} = \left(\frac{1}{K_{\text{pcb}}} + \frac{1}{K_{\text{foundation}}} + \frac{1}{K_{\text{copper pad}}} + \frac{1}{K_{\text{solder}}} + \frac{1}{K_{\text{component}}}\right)^{-1} \tag{3.7}$$

Individual stiffness details:

$$K_{\text{pcb}} = \frac{E_{\text{pcb}} A_{\text{pcb}}}{L_{\text{comp}}} \tag{3.8}$$

where $L_{\text{comp}} = L_{\text{pcb}}$ since Sherlock considers the length of the PCB is the same as the length of the component, and $A_{\text{pcb}} = t_{\text{pcb}} \times (2W_{\text{component}})$ because Sherlock considers the width of the PCB is double the width of the component.

$$K_{\text{foundation}} = \frac{9G_{\text{pcb}}(W_{\text{pad}}/2)}{2 - V_{\text{pcb}}} \tag{3.9}$$

$$K_{\text{copper pad}} = \frac{G_{\text{pad}} A_{\text{pad}}}{t_{\text{pad}}} \tag{3.10}$$

where $A_{\text{pad}} = L_{\text{pad}} \times W_{\text{pad}}$

$$K_{\text{solder}} = \frac{G_{\text{T}_{\text{max}}}^{\text{solder}} A_{\text{solder}}}{h_{\text{solder}}} \tag{3.11}$$

where $A_{\text{solder}} = L_{\text{joint}} \times W_{\text{joint}}$

$$K_{\text{component}} = \frac{E_{\text{comp}} A_{\text{comp}}}{L_{\text{Ocomp}}} \tag{3.12}$$

where $A_{\text{comp}} = t_{\text{comp}} \times W_{\text{comp}}$

3.1.3 Maximum Strain range calculation

Below is the detailed procedure for the maximum strain range calculation for the hot part. The same procedure is followed for the cold part.

First, the total axial displacement $U_{\Delta T_{\text{hot}}}^{\text{total}}$ is calculated using the following formula:

$$U_{\Delta T_{\text{hot}}}^{\text{total}} = |CTE_{\text{comp}} - CTE_{\text{pcb}}| \cdot \Delta T_{\text{hot}} \cdot \frac{L_{\text{D}}}{2} \tag{3.13}$$

Here $L_{\text{D}}/2$ is the half-length of the component for CC component; it is the half-diagonal length of LCCC component because LCCC component is more square in shape.

The differential force generated due to the CTE mismatch of the component and the PCB can be attained with the following equation:

$$F = U_{\Delta T_{\text{hot}}}^{\text{total}} \cdot K_{\Delta T_{\text{hot}}}^{\text{combined}} \tag{3.14}$$

Elastic portion of the stress and strain are determined by the following two equations:

$$\sigma_{\Delta T_{\text{hot}}}^{\text{e}} = \frac{F}{A_{\text{solder}}} = \frac{U_{\Delta T_{\text{hot}}}^{\text{total}} \cdot K_{\Delta T_{\text{hot}}}^{\text{combined}}}{L_{\text{solder}} \cdot W_{\text{solder}}} \tag{3.15}$$

$$\epsilon_{\Delta T_{\text{hot}}}^{\text{e}} = \frac{\sigma_{\Delta T_{\text{hot}}}^{\text{e}}}{G_{\text{T}_{\text{max}}}^{\text{solder}}} \tag{3.16}$$

Now, plastic strain can be calculated with the following equation:

$$\epsilon^{\mathsf{p}}_{\Delta T_{\mathsf{hot}}} = \epsilon^{\mathsf{t}}_{\Delta T_{\mathsf{hot}}} - \epsilon^{\mathsf{e}}_{\Delta T_{\mathsf{hot}}} \tag{3.17}$$

where elastic strain $\epsilon^{\mathsf{t}}_{\Delta T_{\mathsf{hot}}} = \frac{U^{\mathsf{total}}_{\Delta T_{\mathsf{hot}}}}{h_{\mathsf{solder}}}$

Now, inelastic (plastic) strain and Engelmeier equation are used to correct the maximum strain considering viscoplastic creep effects. The corrected maximum strain range is shown below:

$$\epsilon^{\mathsf{max}}_{\Delta T_{\mathsf{hot}}} = \epsilon^{\mathsf{p}}_{\Delta T_{\mathsf{hot}}} \cdot \left(\frac{t_{\mathsf{dwell\text{-}max}}}{t_{\mathsf{critical}}}\right)^{0.136} \cdot e^{E_{\mathsf{a}} \cdot \left(\frac{1}{T_{\mathsf{critical}}} - \frac{1}{T_{\mathsf{max}}+273.15}\right)} \tag{3.18}$$

where

$t_{\mathsf{dwell\text{-}max}}$ = Maximum time the solder stays at a specific temperature

t_{critical} = Critical time. In Sherlock, a critical time of 360 minutes is used for SAC305 without further explanation available from Sherlock or scientific publications.

T_{critical} = Critical temperature in Kelvin(K), which is half of the melting point of the solder. For SAC305 solder, the melting temperature is 217°C, so half of this is 108.5°C is the critical temperature. However Sherlock considered 105°C as the critical temperature for SAC305 solder.

E_{a} = Activation energy of the solder. Sherlock considers the original activation energy of the solder divided by the Boltzmann constant K, and the result is denoted as E_{a}.

Following the same procedure, cold part maximum strain range $\epsilon^{\mathsf{max}}_{\Delta T_{\mathsf{cold}}}$ is calculated.

Now, total Inelastic Strain energy can be obtained by adding the hot part strain energy and cold part strain energy, where strain energy is the multiplication of elastic stress and maximum strain range:

$$\Delta W = \epsilon^{\mathsf{max}}_{\Delta T_{\mathsf{hot}}} \cdot \sigma^{\mathsf{e}}_{\Delta T_{\mathsf{hot}}} + \epsilon^{\mathsf{max}}_{\Delta T_{\mathsf{cold}}} \cdot \sigma^{\mathsf{e}}_{\Delta T_{\mathsf{cold}}} \tag{3.19}$$

3.1.4 Fatigue Life (N_f) Calculation

The mean fatigue life N_f is calculated from the obtained inelastic strain energy with the following formula:

$$\text{mean fatigue life } N_f = ((SE)_{\mathsf{coeff}} \cdot C3 \cdot \Delta W)^{-1} \tag{3.20}$$

where $(SE)_{\mathsf{coeff}}$ is the Strain Energy Coefficient, which is 0.001 in Sherlock by default for SAC305, $C3$ is an empirical constant used by Sherlock. It can be obtained from Table 3.2 in the ANSYS Sherlock theory reference.

Table 3.2: Constants Used by Sherlock

Material	Size (mm)	C1	C2	Size(mm)	C3
SAC305	0 to 3.2	1	1	0 to 5.0	1
	$>$ 3.2	1	1	$>$ 5.0	1.5

3.1.5 Validation of CC / LCCC Solder Fatigue Theory

In the Sherlock Theory Reference, a raw theory is presented without a detailed explanation, providing a basic understanding of how ANSYS Sherlock calculates solder fatigue life using the CC & LCCC Model. This raises the question of whether the theory explained in subsection 2.1 is sensible or acceptable. To address this, a validation is conducted.

To validate the theory, simulations are performed on two resistors, R14 and R15, using Ansys Sherlock. R14 corresponds to the resistor with part number SMT 0402, and R15 corresponds to SMT 0603 in the Sherlock resistors part library. No further modifications are made to the component property settings, and the default settings are utilized. The part properties for R14 & R15 resistor in Ansys Sherlock are illustrated in Figure 3.4, with the stack-up shown in Figure 3.4 for SAC305 solder.

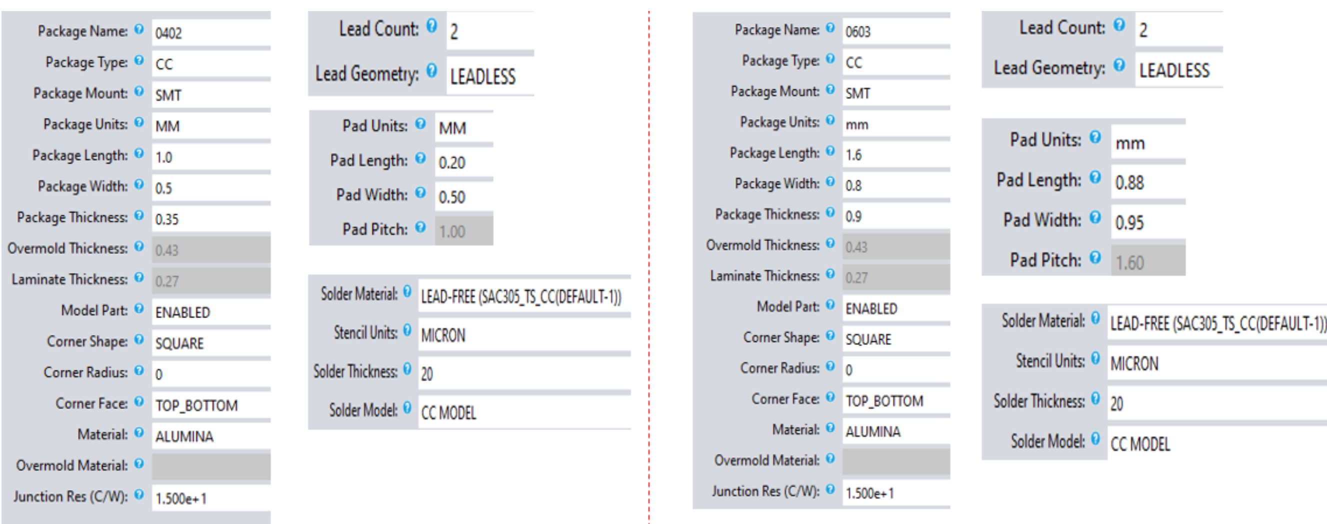
Table 3.3: Thermal Cycle Profile

Step	Temperature (°C)	Duration (minutes)	Phase
1	-40	29.8	hold
2	125	0.2	ramp
3	125	29.8	hold
4	-40	0.2	ramp

The simulation results are then compared with the manual calculation results based on the theory explained in subsection 3.1. This comparison aims to assess the agreement between the simulated and manually calculated outcomes, providing insights into the validity and accuracy of the solder fatigue theory.

Table 3.4: AL IMS Stack up considering Tg (Tg: Glass Transition Temperature)

Layer	Type	Material	Thickness
1	Signal	Copper(50.0% / RM_L1	35 micron
2	Di-electric	Bergquist MP(Considering Tg)	76 micron
3	Substrate	Aluminum	1.57 mm



(a) R14 Resistor (b) R15 Resistor

Figure 3.4: R14 & R15 Resistors Property Setting

Manual calculations are performed using Python, following the same theoretical approach. The units employed in the manual calculation include MPa for elastic modulus, eV for activation energy, and $1/°C$ (or $1/K$) for CTE, with all other units in SI units. The reference temperature (T_{mean}) used here is $(30.445 + 273.15)$K.

Both the simulated result and manually calculated result exactly match, as illustrated in Table 3.5.

Table 3.5: Comparison: Simulation vs Manual

Components	Simulated mean TTF	Manually Calculated mean TTF
R14	362	362
R15	648	648

3.2 Solder Fatigue Theory for BTC Components

3.2.1 Geometrical Impact Analysis

To explore the potential impacts of solder pad orientation on fatigue life, an trial simulation for a arbitrary LED on Aluminium IMS pcb is conducted. Figure 3.5 illustrates the considered variations in anode, cathode, and thermal pad orientations.

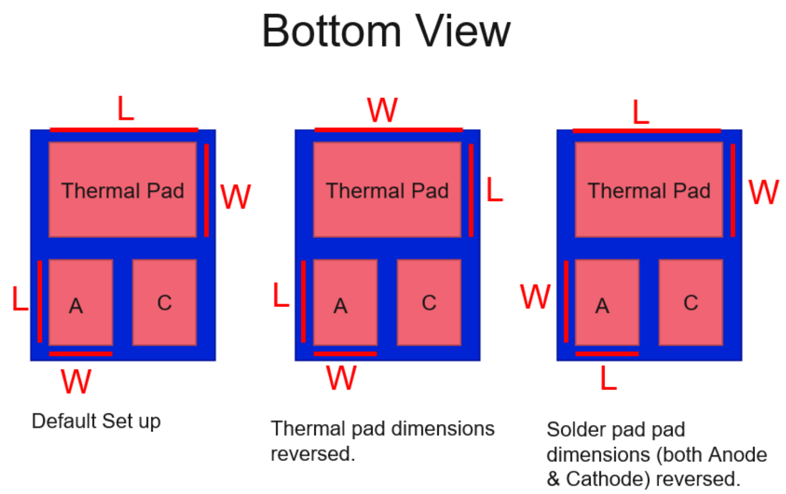


Figure 3.5: Arrangement of Anode, Cathode, and Thermal Pad Orientations

Surprisingly, a 90° alteration in anode or cathode orientation exhibits no discernible impact on fatigue life. However, changing the orientation of the thermal pad (swapping length and width) leads to a substantial increase in fatigue life, from approximately 510 hours or cycles to nearly 690 hours or cycles, as depicted in Figure 3.6.

This observation suggests the likelihood that ANSYS Sherlock employs a default geometry for the orientation of Anode, Cathode, and Thermal pads.

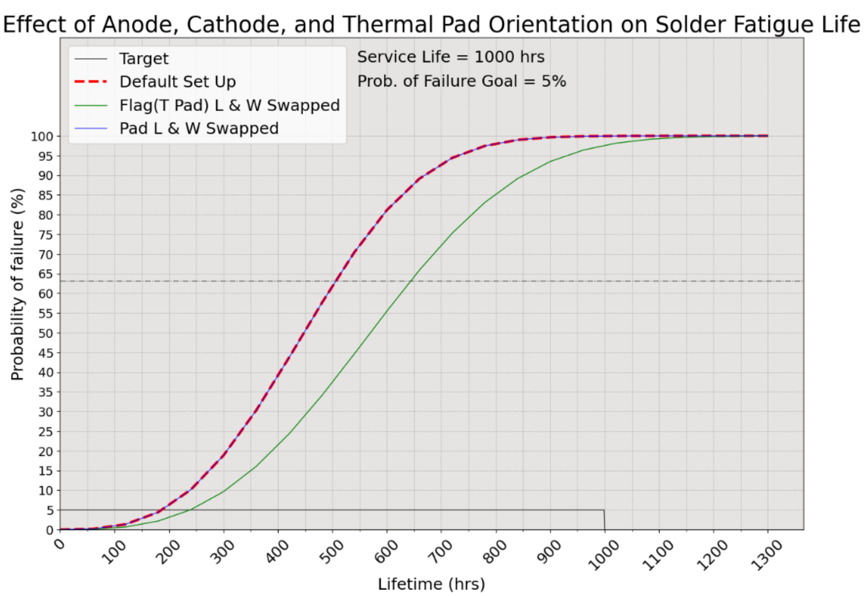


Figure 3.6: Effect of Anode, Cathode, and Thermal Pad Orientation Changes on Solder Fatigue Life

3.2.2 Development of BTC Model Theory

In the ANSYS Sherlock library, only two BTC components are available: (LED BTC 35, 3 pad) and (LED BTC 16, 2 pad). Unfortunately, LED BTC 16 does not function in simulations even with default settings.

Dr. Blattau, the primary architect and software development team leader of the Sherlock Automated Design Analysis (ADA) App program, published a study on LEDs soldered onto a Metal Core Printed Circuit Board with SAC305 solder in 2017[34]. Assuming the same geometric-shaped LED from his work is used in Sherlock, a manual calculation is performed. For both the manual calculation and simulation, the aluminum stack-up 3.4 and SAC305 solder are utilized, with the geometric shape shown in Figure 3.7.

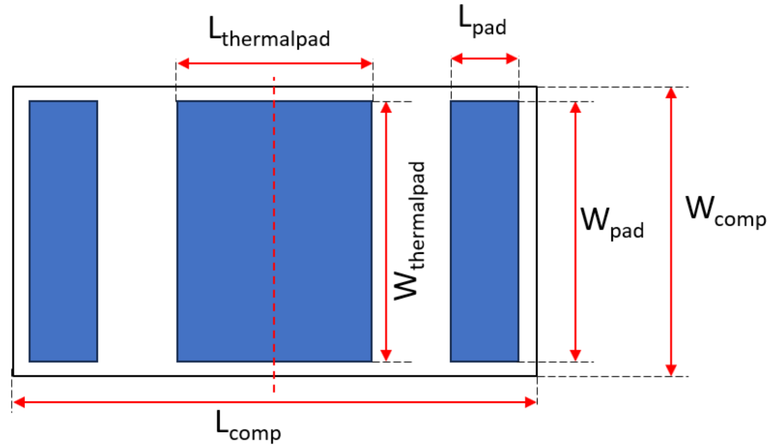


Figure 3.7: Default Footprint Geometry. The image is built with input from [34].

The part properties for the simulation are illustrated in Figure 3.8.

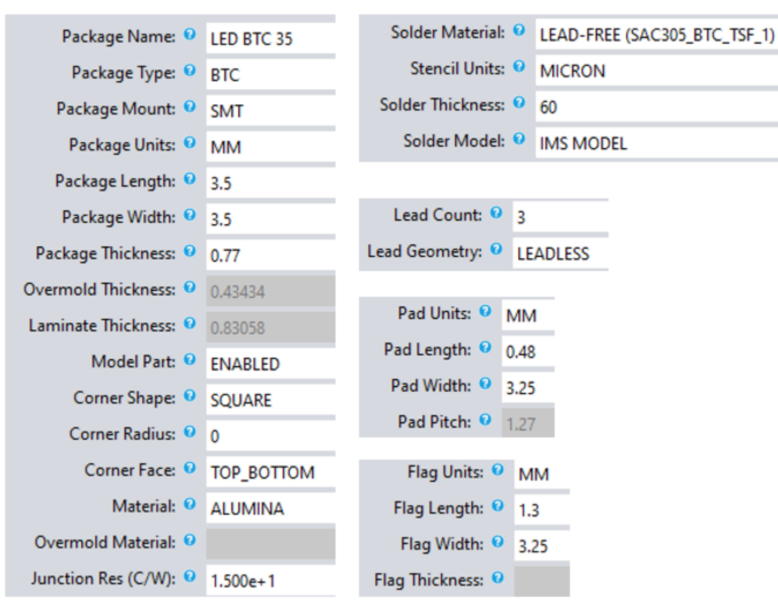


Figure 3.8: Part Properties for BTC 35 LED

In the manual calculation, considering the thermal pad's impact on fatigue life, the stiffness of the thermal pad is incorporated into the total stiffness calculation.

3.2.3 Total Stiffness Calculation

Referring to the stiffness equation 3.7, this time, considering the symmetric view, one solder (Anode or Cathode, both are the same in dimension), and thermal pad stiffness have to be used in the total stiffness calculation. For the foundation stiffness, the formula is:

$$K_f = \frac{9 \cdot G_{\text{pcb}} \cdot a}{2 - V_{\text{pcb}}} \tag{3.21}$$

where G_{pcb} is the shear modulus of the PCB, V_{pcb} is the Poisson ratio of the PCB, and a is the edge-to-edge distance of the bond pad, in this case, the length of the pad (Thermal pad / Anode Pad / Cathode pad).

To calculate the equivalent stiffness, we can adopt the approach illustrated in Figure 3.9, considering a symmetrical view.

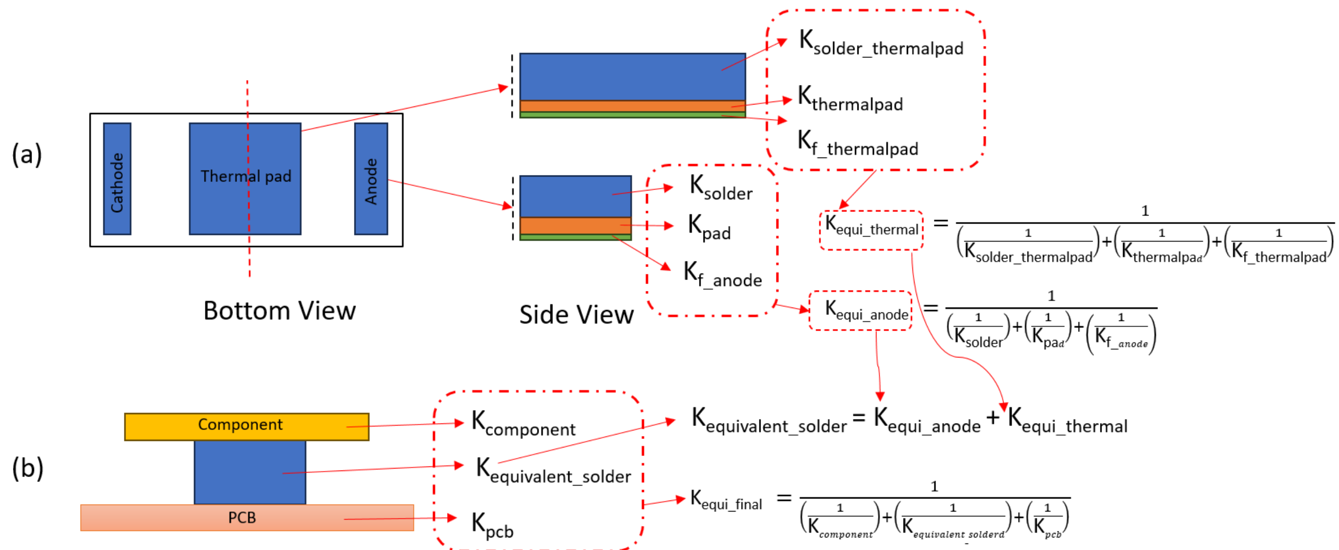


Figure 3.9: Equivalent Stiffness Calculation for BTC Component

For the Thermal Pad: The stiffness of the thermal pad solder, thermal pad (copper pad), and the foundation stiffness of the thermal pad are in series. Therefore, the equivalent stiffness for these three components is given by:

$$K_{\text{equi\_thermal}} = \frac{1}{\left(\frac{1}{K_{\text{solder\_thermalpad}}} + \frac{1}{K_{\text{thermalpad}}} + \frac{1}{K_{\text{f\_thermalpad}}}\right)} \tag{3.22}$$

where:

$$K_{\text{thermalpad}} = \frac{\text{thermal pad material shear modulus} \cdot \text{thermal pad length} \cdot \text{thermal pad width}}{\text{thermal pad thickness}}$$

$$K_{\text{solder\_thermalpad}} = \frac{\text{thermal pad solder material shear modulus} \cdot \text{thermal pad solder length} \cdot \text{thermal pad solder width}}{\text{thermal pad solder height}}$$

$$K_{\text{f\_thermalpad}} = \frac{9 \cdot \text{shear modulus of PCB} \cdot \text{thermal pad length}}{2 - \text{Poisson ratio of PCB}}$$

Similarly,
For the Anode: The stiffness of the Anode solder, Anode pad (copper pad), and the foundation stiffness of the Anode are in series. The equivalent stiffness for these three components is given by:

$$K_{\text{equi\_anode}} = \frac{1}{\left(\frac{1}{K_{\text{solder}}} + \frac{1}{K_{\text{pad}}} + \frac{1}{K_{\text{f\_anode}}}\right)} \tag{3.23}$$

where:

$$K_{\text{solder}} = \frac{\text{Anode solder material shear modulus} \cdot \text{Anode solder length} \cdot \text{Anode solder width}}{\text{Anode solder height}}$$

$$K_{\text{pad}} = \frac{\text{Anode pad material shear modulus} \cdot \text{Anode pad length} \cdot \text{Anode pad width}}{\text{Anode pad thickness}}$$

$$K_{\text{f\_anode}} = \frac{9 \cdot \text{shear modulus of PCB} \cdot \text{Anode pad length}}{2 - \text{Poisson ratio of PCB}}$$

Now, $K_{\text{equi\_thermal}}$ and $K_{\text{equi\_anode}}$ are in parallel to each other, so the equivalent stiffness of these two is:

$$K_{\text{equivalent\_solder}} = K_{\text{equi\_thermal}} + K_{\text{equi\_anode}} \tag{3.24}$$

Finally, $K_{\text{component}}$, $K_{\text{equivalent\_solder}}$, and K_{pcb} are in series. Therefore, the final stiffness is:

$$K_{\text{equi\_final}} = \frac{1}{\left(\frac{1}{K_{\text{component}}} + \frac{1}{K_{\text{equivalent\_solder}}} + \frac{1}{K_{\text{pcb}}}\right)} \tag{3.25}$$

where,

$$K_{\text{component}} = \frac{\text{Elastic modulus of Component} \times \text{width of component} \times \text{thickness of component}}{\text{Length of component}}$$

Considering the length of the PCB as the same length as the component, and the width of the PCB is double the width of the component,

$$K_{\text{pcb}} = \frac{\text{Elastic Modulus of PCB} \times (2 \times \text{width of component}) \times \text{Thickness of PCB}}{\text{Length of component}}$$

3.2.4 Maximum Strain range calculation

Similar to the CC and LCCC Model, the process involves partitioning into cold and hot sections. The same procedure is employed for calculating the maximum strain range in BTC Model. Below is the detailed procedure for the maximum strain range calculation for the hot part. The same procedure is followed for the cold part.

T_{mean} represents the reference temperature in °C, which is the temperature at which all dimensions of PCB components, solder, copper pads, etc., are measured.

T_{max} = maximum temperature of the thermal cycle in °C.
T_{min} = minimum temperature of the thermal cycle in °C.

$\Delta T_{\text{hot}} = T_{\text{max}} - T_{\text{mean}}$ $\Delta T_{\text{cold}} = T_{\text{mean}} - T_{\text{min}}$

for the hot part , total stiffness is ,

$$K_{\Delta T_{\text{hot}}}^{\text{combined}} = K_{\text{equi\_final}}$$

where $K_{\text{equi\_final}}$ is the same as in Equation 6.6. The only difference is that, while calculating the stiffness of each solder, the shear modulus of the solder material at the maximum temperature is considered for the hot part.

First, the total axial displacement $U_{\Delta T_{\text{hot}}}^{\text{total}}$ is calculated using the following formula:

$$U_{\Delta T_{\text{hot}}}^{\text{total}} = |CTE_{\text{comp}} - CTE_{\text{pcb}}| \cdot \Delta T_{\text{hot}} \cdot \left(\frac{L_{\text{D}}}{2} \cdot C\right) \tag{3.26}$$

Here $\left(\frac{L_{\text{D}}}{2} \cdot C\right)$ is, in fact, the distance to the solder from the neutral for the geometric shape of the LED(including footprint) shown in Figure 3.7, $C = \frac{1}{\sqrt{2}}$ in this case. [14]

The differential force generated due to the CTE mismatch of the component and the PCB can be attained with the following equation:

$$F = U^{\text{total}}_{\Delta T_{\text{hot}}} \cdot K^{\text{combined}}_{\Delta T_{\text{hot}}} \tag{3.27}$$

Elastic portion of the stress and strain are determined by the following two equations:

$$\sigma^{\text{e}}_{\Delta T_{\text{hot}}} = \frac{F}{A_{\text{solder}}} = \frac{U^{\text{total}}_{\Delta T_{\text{hot}}} \cdot K^{\text{combined}}_{\Delta T_{\text{hot}}}}{L_{\text{solder}} \cdot W_{\text{solder}}} \tag{3.28}$$

$$\epsilon^{\text{e}}_{\Delta T_{\text{hot}}} = \frac{\sigma^{\text{e}}_{\Delta T_{\text{hot}}}}{G^{\text{solder}}_{\text{T}_{\text{max}}}} \tag{3.29}$$

Now, plastic strain can be calculated with the following equation:

$$\epsilon^{\text{p}}_{\Delta T_{\text{hot}}} = \epsilon^{\text{t}}_{\Delta T_{\text{hot}}} - \epsilon^{\text{e}}_{\Delta T_{\text{hot}}} \tag{3.30}$$

where elastic strain $\epsilon^{\text{t}}_{\Delta T_{\text{hot}}} = \frac{U^{\text{total}}_{\Delta T_{\text{hot}}}}{h_{\text{solder}}}$

Now, inelastic (plastic) strain and Engelmeier equation are used to correct the maximum strain considering viscoplastic creep effects. The corrected maximum strain range is shown below:

$$\epsilon^{\text{max}}_{\Delta T_{\text{hot}}} = \epsilon^{\text{p}}_{\Delta T_{\text{hot}}} \cdot \left(\frac{t_{\text{dwell-max}}}{t_{\text{critical}}}\right)^{0.136} \cdot e^{E_{\text{a}} \cdot \left(\frac{1}{T_{\text{critical}}} - \frac{1}{T_{\text{max}+273.15}}\right)} \tag{3.31}$$

where

$t_{\text{dwell-max}} =$ Maximum time the solder stays at a specific temperature

$t_{\text{critical}} =$ Critical time. In Sherlock, a critical time of 360 minutes is used for SAC305 without further explanation available from Sherlock or scientific publications.

$T_{\text{critical}} =$ Critical temperature in Kelvin(K), which is half of the melting point of the solder. For SAC305 solder, the melting temperature is 217°C, so half of this is 108.5°C. But Sherlock considered 105°C as the critical temperature FOR SAC305.

$E_{\text{a}} =$ Activation energy of the solder. Sherlock considers the original activation energy of the solder divided by the Boltzmann constant K, and the result is denoted as E_{a}.

Following the same procedure, cold part maximum strain range $\epsilon^{\text{max}}_{\Delta T_{\text{cold}}}$ is calculated.

Now, total Inelastic Strain energy can be obtained by adding the hot part strain energy and cold part strain energy, where strain energy is the multiplication of elastic stress and maximum strain range:

$$\Delta W = \epsilon^{\text{max}}_{\Delta T_{\text{hot}}} \cdot \sigma^{\text{e}}_{\Delta T_{\text{hot}}} + \epsilon^{\text{max}}_{\Delta T_{\text{cold}}} \cdot \sigma^{\text{e}}_{\Delta T_{\text{cold}}} \tag{3.32}$$

3.2.5 Fatigue Life (N_f) Calculation

The mean fatigue life N_f is calculated from the obtained inelastic strain energy with the following formula:

$$\text{mean fatigue life } N_f = ((SE)_{\text{coeff}} \cdot C3 \cdot \Delta W)^{-1} \tag{3.33}$$

where $(SE)_{\text{coeff}}$ is the Strain Energy Coefficient, which is 0.001 in Sherlock by default for SAC305, $C3$ is an empirical constant used by Sherlock. It can be obtained from Table 3.2 in the ANSYS Sherlock theory reference.

3.2.6 Validation of BTC Solder Fatigue Theory

The theory discussed for the BTC model so far is built upon the CC model theory with input from the ANSYS Theory Reference and insights from N. Blattau [34] and A. Syed [40]. However, it is important to note that this theory is not endorsed by Sherlock. Consequently, there is a possibility that Sherlock may not follow the same theory when predicting the fatigue life of BTC components, raising questions about the validity of the theory.

To address this concern, the theory needs validation. For this purpose, the simulation utilizes the only available functioning BTC component, LED BTC 35, from the Ansys Sherlock part library. The fatigue life prediction is simulated with the default settings and the default properties is considered for the manual calculation, following the new BTC theory. The thermal cycle remains consistent with Table 3.3, the stack-up mirrors that of Table 3.4, and SAC305 solder is considered.

The simulation properties settings are illustrated in Figure 3.10.

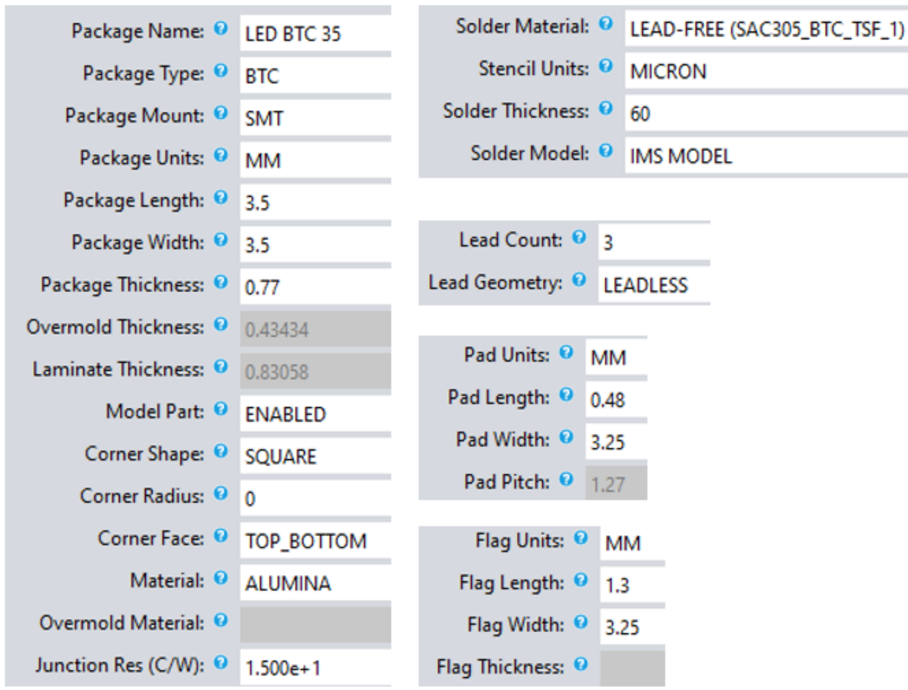


Figure 3.10: BTC 35 Property Settings

Manual calculations are performed using Python, following the same theoretical approach. The units employed in the manual calculation include MPa for elastic modulus, eV for activation energy, and $1/°C$ (or $1/°K$) for CTE, with all other units in SI units. The reference temperature (T_{mean}) used is $(27 + 273.15)\,°K$.

Both the simulated result and manually calculated result exactly match, as illustrated in Table 3.6.

Table 3.6: Comparison: BTC 35 Simulation vs Manual

Components	Simulated mean TTF	Manually Calculated mean TTF
BTC 35	538	538

This confirms the correctness of the BTC Model Theory. However, to ensure its accuracy, checking with other BTC components would be ideal. Unfortunately, other BTC parts were not found in the Sherlock library, so only BTC 35 was available for use. Notably, another component, BTC 16, didn't work in the simulation, even when attempting the default settings.

The Python code blocks for the manual calculation are available in the Appendix.

Chapter 4

Settings in ANSYS Sherlock

4.1 Pre-Settings

In the simulation process, default settings were uniformly applied based on the simulation needs. Regarding part property configurations, the 'BTC' Model is linked with the 'BTC' package type, and the 'IMS' solder model is chosen for all instances of 'BTC' Models.

Concerning solder materials, two options were considered: SAC305 solder and Innolot solder. The ANSYS Solder library already includes SAC305, while Innolot solder required manual creation. The specific properties for Innolot solder library creation are illustrated in Figure 4.1.

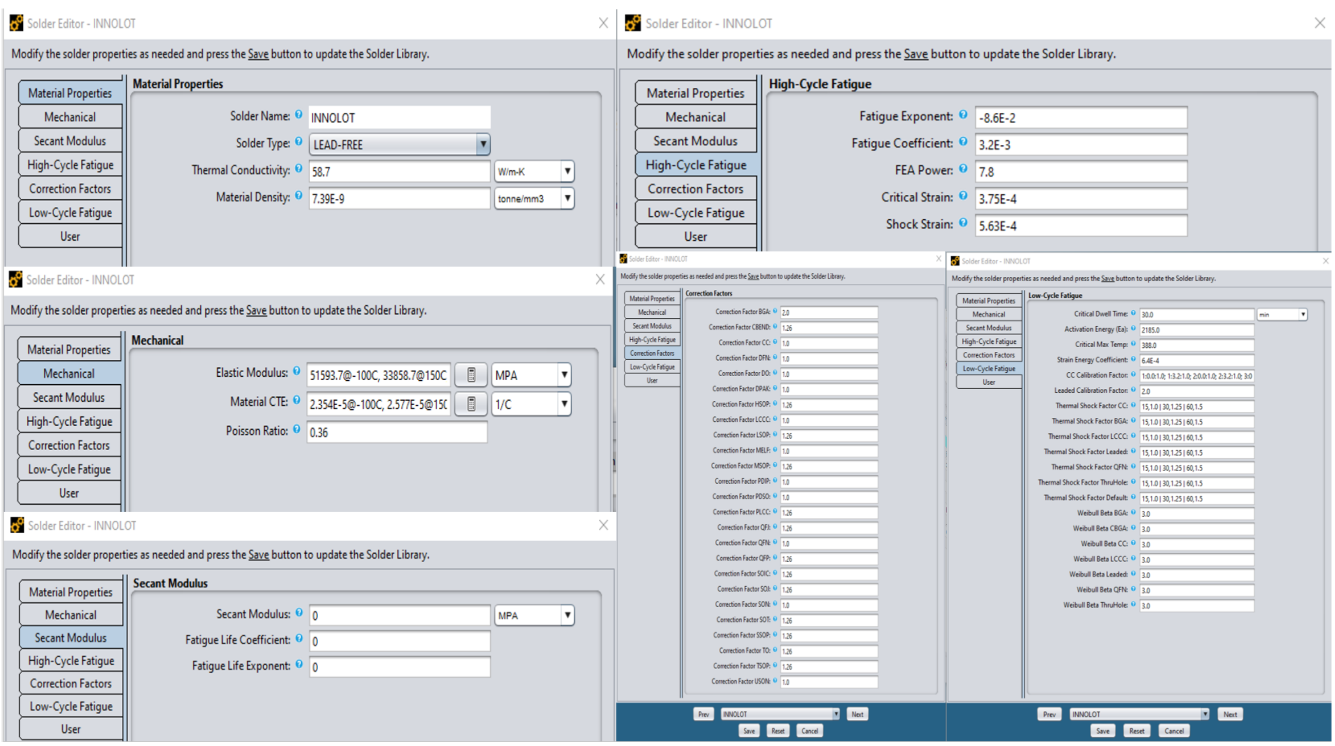


Figure 4.1: Innolot Solder Library Properties Setting

For the thermal profile setting, it is configured as shown in Figure 4.2.

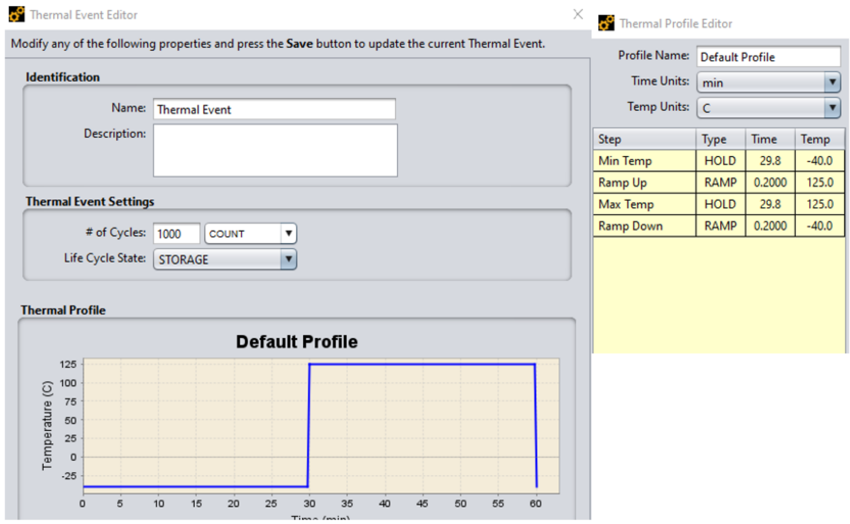


Figure 4.2: Thermal Profile Setting

In configuring the dielectric properties of Bergquist MP, the glass transition temperature (T_g) is crucial. T_g represents the temperature at which the material shifts from a rigid to a flexible state. For Aluminum IMS PCBs, T_g is set at 90°C, within the thermal shock range, accounting for varying coefficients of thermal expansion (CTE) below and above this threshold.

For the component material, a material named ALN Ceramic was created with the property settings shown in Figure 4.3.

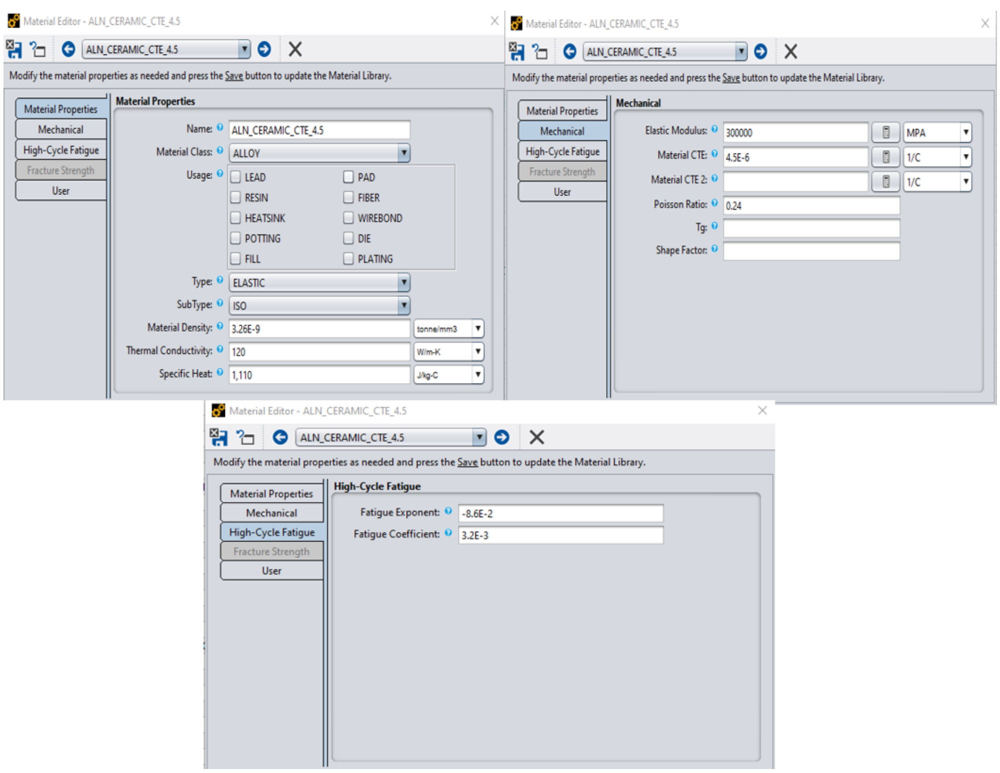


Figure 4.3: ALN CERAMIC Material Library Setting

4.1.1 Coefficient of Thermal Expansion (CTE) of ALN Ceramic Material & Thermal Shock Factor (TSF)

The Aluminum Nitride (ALN Ceramic) material employed in the LED packaging possesses a Coefficient of Thermal Expansion (CTE) of 4 ppm/°C. However, owing to the inclusion of metals within the material, the CTE may exhibit variability across different LED models due to varying metal concentrations. Consequently, it becomes imperative to establish a singular CTE value for the sake of simplicity, as individually measuring the CTE of each LED model involves considerable time and resource expenditures. Despite the varying CTE, a value of 4.5 ppm/°C is chosen, as this slight variation within the specified range has negligible impact on reliability. Figure 4.4 illustrates the minimal impact on reliability with a slight increase in CTE of a arbitrary LED soldered on Aluminum IMS PCB, utilizing SAC305 solder and CC model.

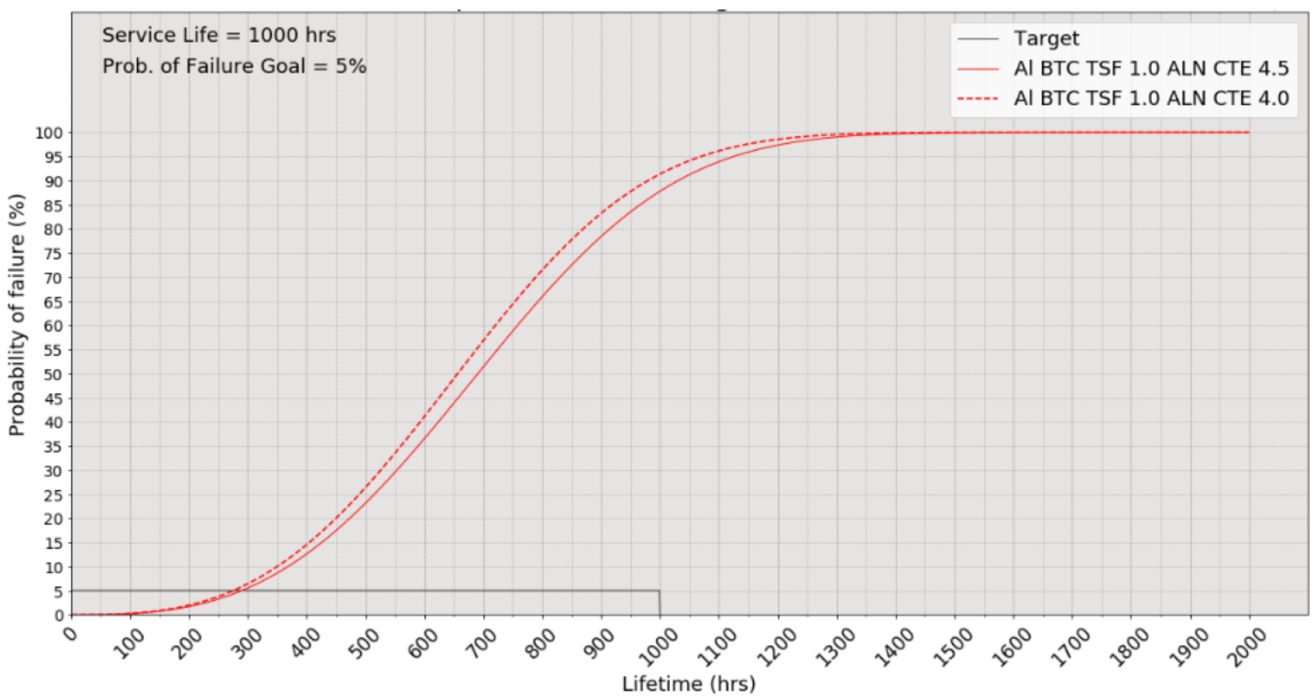


Figure 4.4: Influence of Variations in CTE in Aluminum Nitride (ALN) components on solder Reliability

Ramp time (transfer time to different temperatures) during thermal changes affects the solder fatigue life. A faster ramp time leads to a lower fatigue life of solder. To address this, a multiplication factor called the Thermal Shock Factor (TSF) is introduced. The TSF is employed to adjust the calculated mean fatigue life, making it align more closely with the fatigue life measured in the laboratory.

Through experimental simulations, it was determined that a TSF of 1 is suitable for the thermal profile considered in Figure 4.2. However, adjusting the TSF (Thermal Shock Factor) can be done in the Solder material setting, as illustrated in Figure 4.5.

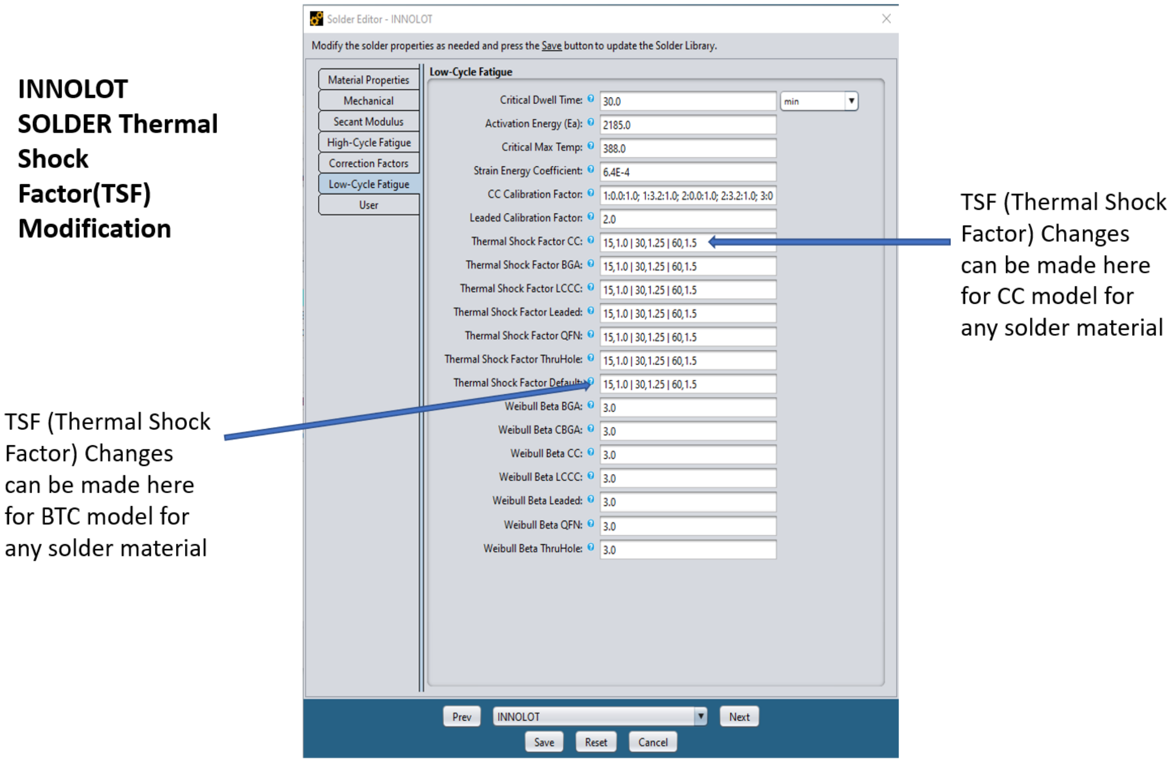


Figure 4.5: Changing TSF (Thermal Shock Factor) in Solder Material Setting

It's important to note that multiplying or dividing something by 1 doesn't alter the result. Therefore, the Thermal Shock Factor of 1 is not explicitly used in any manual calculation of fatigue life throughout this work.

4.2 Configuration of Samsung V 1 chip, V 2 chip, and V 3 chip LEDs

The highlighted options with red color in Figures 4.6, 4.7, and 4.8 represent the properties applied according to the following criteria:

- **Package Type:**
  - BTC for BTC Model (Samsung V Series LEDs)
- **Solder Material:**
  - SAC305 Solder
  - INNOLOT Solder
- **Solder Model:**
  - IMS selected for all BTC Models
- **Solder Thickness:**
  - 60 microns for SAC305 solder
  - 65 microns for INNOLOT Solder
- **Package Material:**
  - ALN CERAMIC CTE 4.5 ppm/°C

The three stack-ups employed for simulating solder fatigue of Samsung V 1 chip, 2 chip, and 3 chip LEDs are presented in Tables 4.1, 4.2, and 4.3.

Layer	Type	Material	Thickness
1	Signal	Copper(50.0% / RM_L1	35 micron
2	Di-electric	Bergquist MP(Considering Tg)	76 micron
3	Substrate	Aluminum	1.57 mm

Table 4.1: Al IMS Stack up for Samsung V series LEDs considering Tg (Tg: Glass Transition Temperature)

Layer	Type	Material	Thickness
1	Signal	Copper(50.0% / RM_L1	70 micron
2	Di-electric	Bergquist MP(Considering Tg)	76 micron
3	Substrate	Copper	1 mm

Table 4.2: Cu IMS Stack up for Samsung V series LEDs

Layer	Type	Material	Thickness
1	Signal	Copper(45.6% / RM_L1	57 micron
2	Laminate	Generic Fr-4	1.44 micron
3	Signal	Copper(45.4%)/ RM_L3	57 micron

Table 4.3: Fr-4 Stack up for Samsung V series LEDs

Regarding Fr-4 PCB, both the simulated and real lab-measured fatigue life is exceptionally high, indicating that it hardly fails. Furthermore, Fr-4 PCB is deemed unsuitable for the heat dissipation requirements of Automotive LEDs. Consequently, it is not further discussed for Fr-4 PCB in this work.
Regarding part properties settings, the following properties, as shown in Figures 4.6, 4.7, and 4.8, are used. It's noteworthy that a constant package thickness of 0.4 mm is applied to all three LEDs instead of the actual package thickness. This decision is made because the actual package thickness is nearly 0.8 mm; however, the LEDs include a yellow silicon part, which affects the effective thickness. As a result, approximately half of the thickness of the component is considered as the effective thickness.

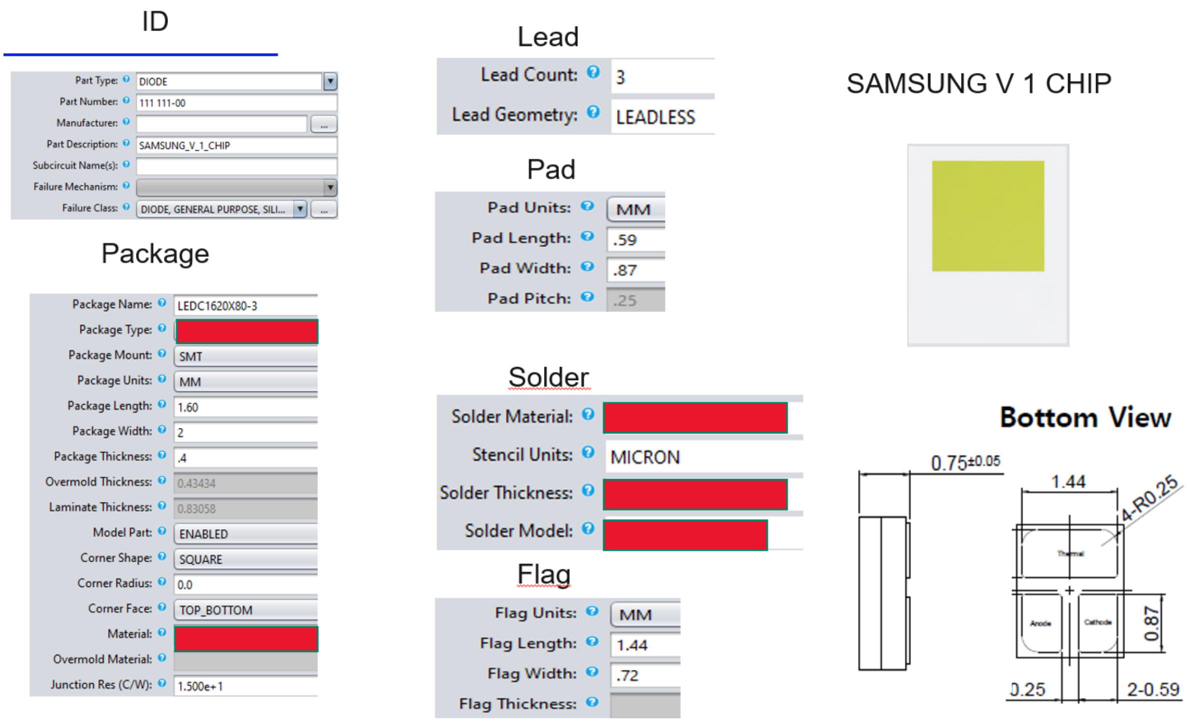


Figure 4.6: Properties setting for Samsung V 1 chip solder fatigue simulation.

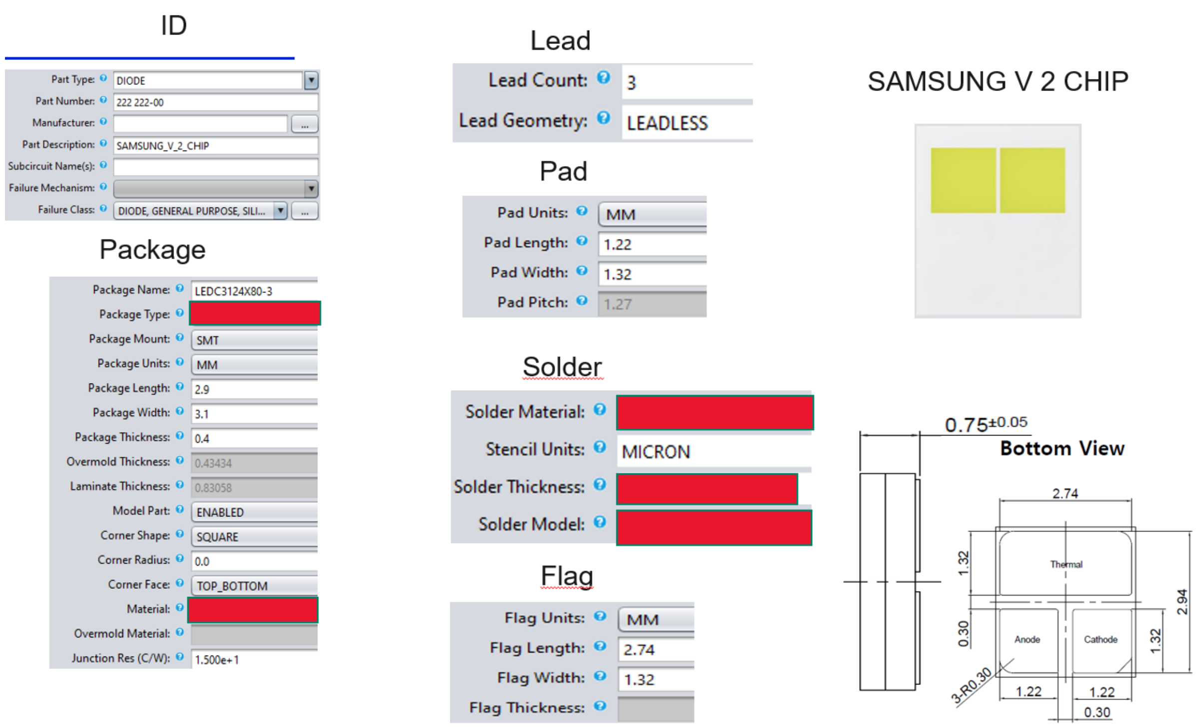


Figure 4.7: Properties setting for Samsung V 2 chip solder fatigue simulation.

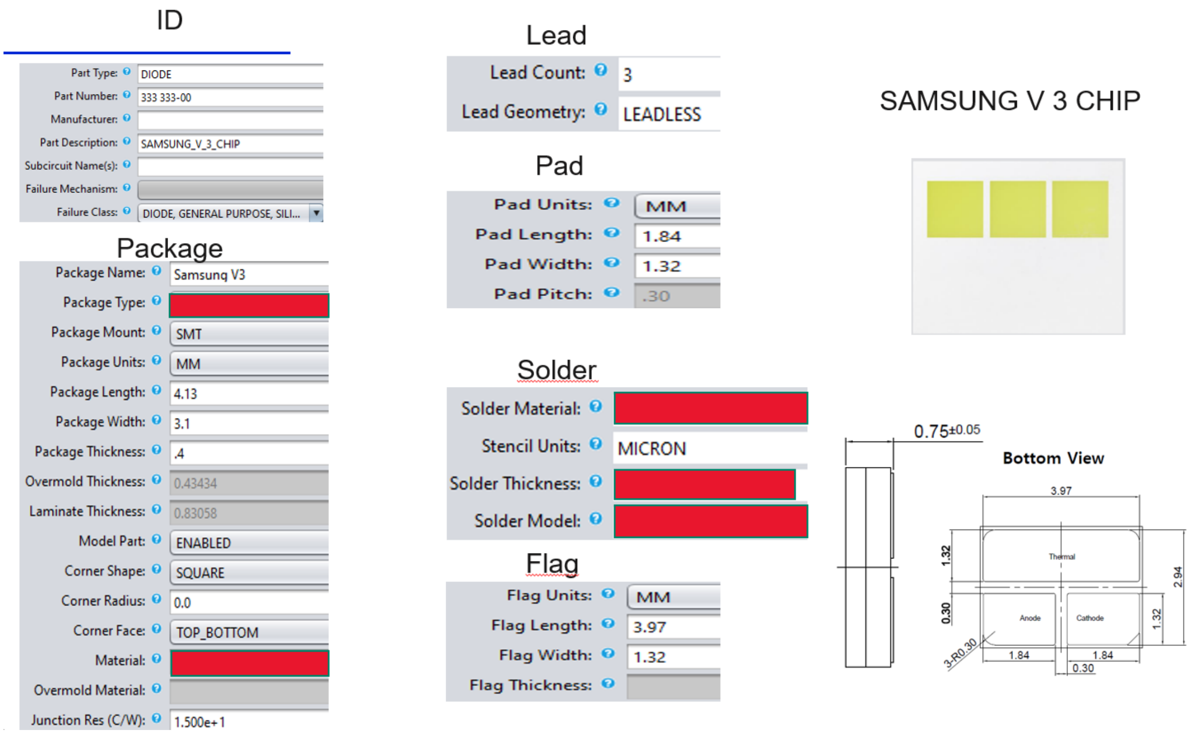


Figure 4.8: Properties setting for Samsung V 3 chip solder fatigue simulation.

Chapter 5

Solder Fatigue Simulation Outcome: Samsung V Series LEDs

This chapter presents simulation results for Samsung V Series LEDs, encompassing various configurations, including 1-chip, 2-chip, and 3-chip setups mounted on Aluminum and Copper IMS PCBs. The simulations consider both SAC305 and Innolot solder materials.

The simulation curves, derived from the output of ASNYS Sherlock, are based on the shape parameter $\beta = 3$. However, for enhanced comparability with real test results, adjustments are made to the curves by incorporating β values obtained from the actual test result curves. It's important to note that this adjustment does not impact the mean Time to Failure (TTF), as variations in the shape parameter (β) only affect the slope of the curves without altering the mean TTF point. The adjustment procedure is available in subsection 2.3.1 in Chapter 2.

In considering the mean Total Time to Failure (TTF), the lifetime (in hours) at a 63.2% probability of failure is taken into account.

This chapter includes a detailed comparison between the predicted fatigue life obtained through simulation and the actual fatigue life observed in real tests.

5.1 SAC305 Solder, Aluminum IMS

In Figure 5.1, life prediction curves for Samsung V 1, 2, and 3 chip LEDs are depicted, considering the implementation of SAC305 solder. The simulation results indicate a total time to failure of 1075 cycles for V1 chip LEDs. This time reduces to 833 cycles for V2 chip LEDs and slightly increases to 873 cycles for V3 LEDs than V2 LEDs,

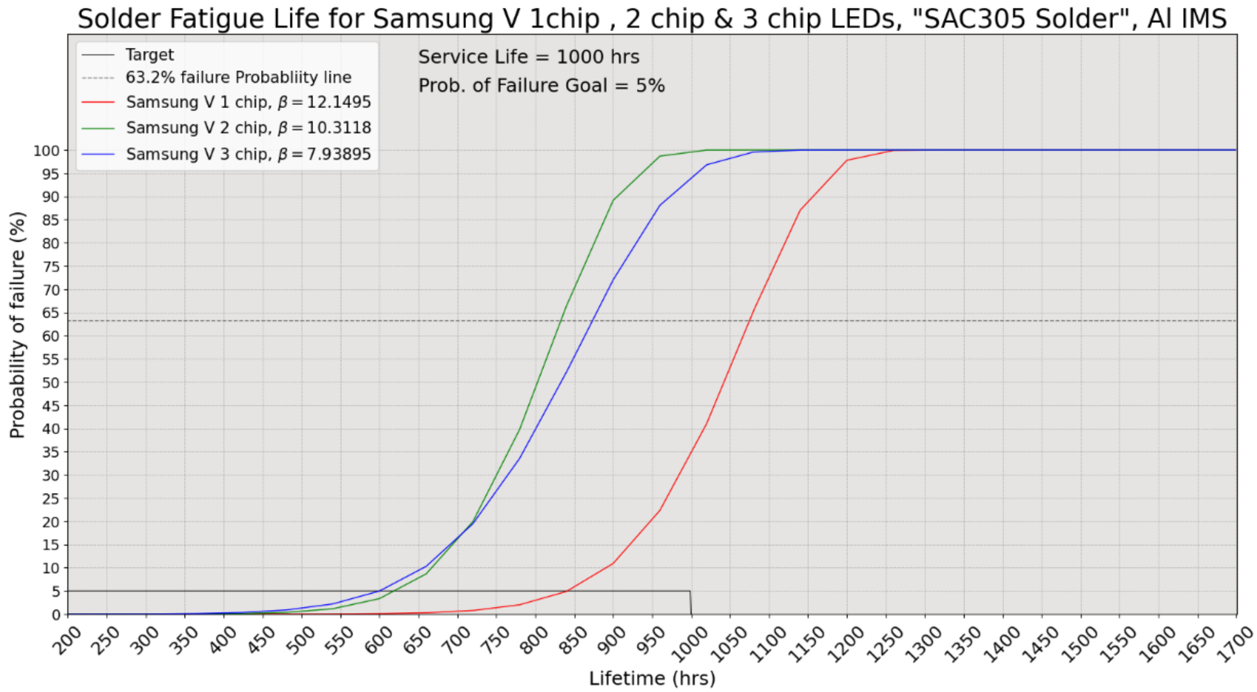


Figure 5.1: Simulation output for Samsung V Series LEDs on Al IMS PCB, SAC305 Solder

5.1.1 Simulated vs Real Fatigue Life: Samsung V1, SAC305 Solder, Al IMS

When comparing the simulated and actual fatigue life of the Samsung V1 chip LED on Aluminum IMS PCB, as shown in Figure 5.2, a substantial difference is observed. The simulation predicts a fatigue life of approximately 1075 thermal cycles, while the actual solder fatigue life is nearly 3180 cycles.

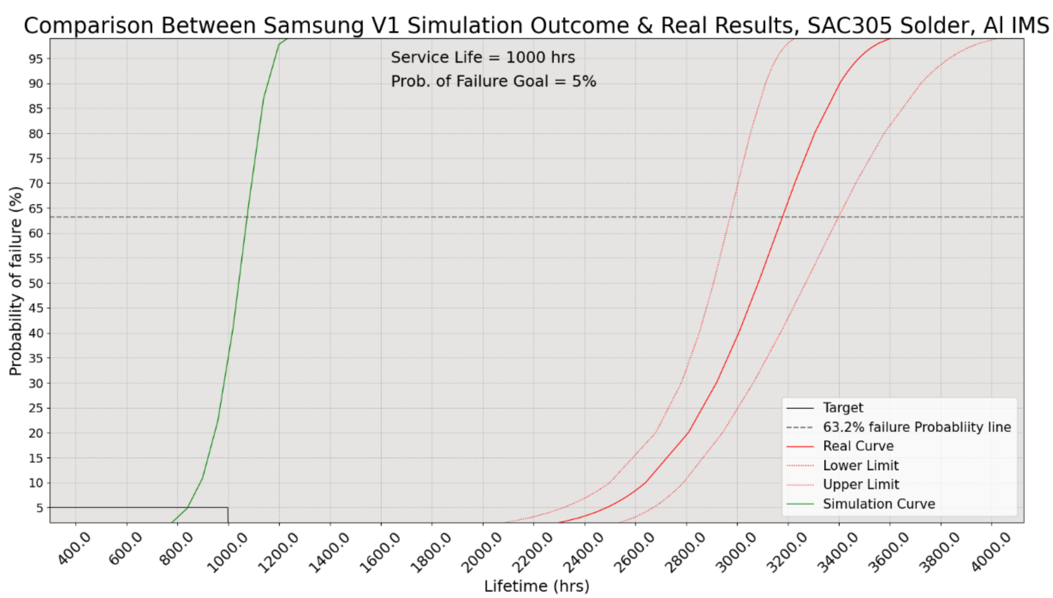


Figure 5.2: Comparison: Samsung V1 on Al IMS, SAC305 Solder (BTC Model, $\beta = 12.1495$)

5.1.2 Simulated vs Real Fatigue Life: Samsung V2, SAC305 Solder, Al IMS

The fatigue life comparison for the Samsung V2 chip LED with SAC305 solder on Aluminum IMS PCB, illustrated in Figure 5.3, exhibits a notable difference. The simulation suggests a fatigue life of approximately 833 thermal cycles, in contrast to the actual solder fatigue life of nearly 3090 cycles.

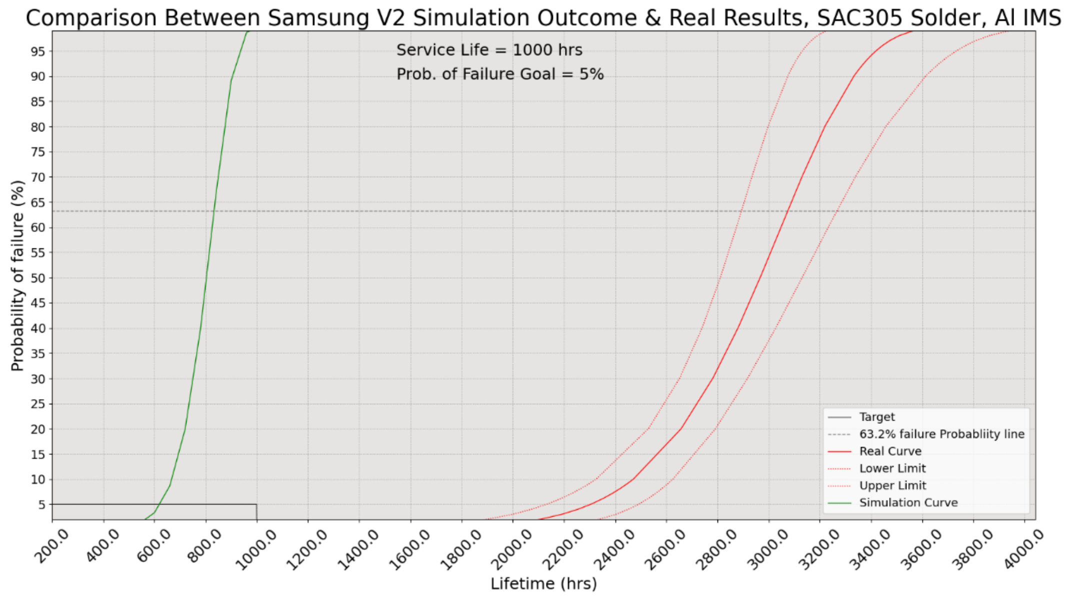


Figure 5.3: Comparison: Samsung V2 on Al IMS, SAC305 Solder (BTC Model, $\beta = 10.3118$)

5.1.3 Simulated vs Real Fatigue Life: Samsung V3, SAC305 Solder, Al IMS

Figure 5.12 portrays the fatigue life comparison for the Samsung V3 chip LED featuring SAC305 solder on Copper IMS PCB, emphasizing a notable divergence. The simulation anticipates a fatigue life of about 1316 thermal cycles, while the actual solder fatigue life is nearly 3094 cycles.

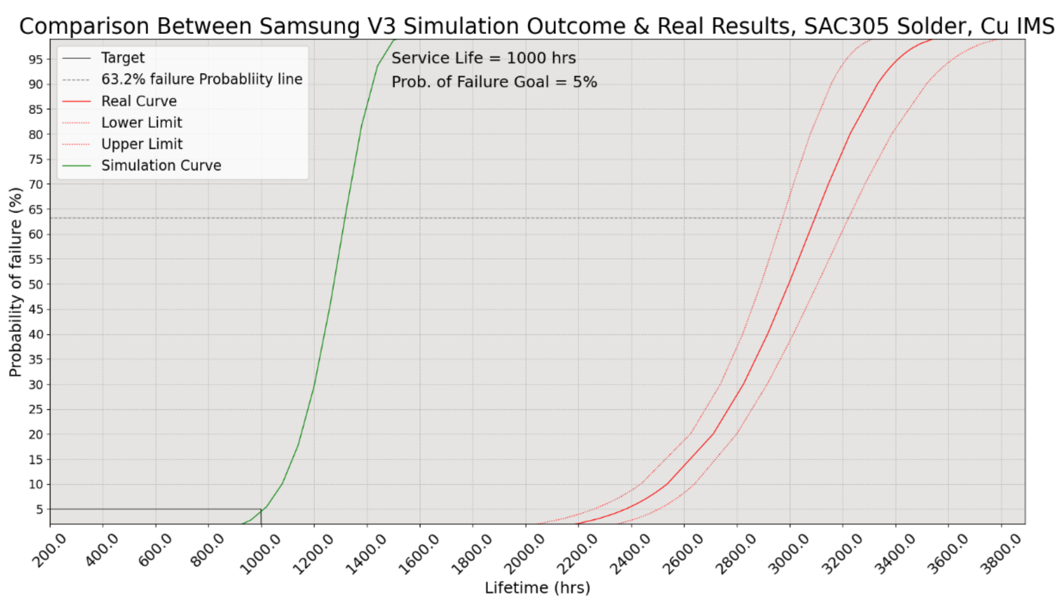


Figure 5.4: Comparison: Samsung V3 on Cu IMS, SAC305 Solder (BTC Model, $\beta = 11.3026$)

5.2 Innolot Solder, Aluminum IMS

The life prediction curves in Figure 5.5 showcase the performance of Samsung V 1, 2, and 3 chip LEDs, this time considering INNOLOT solder. According to the simulation, V1 chip LEDs exhibit a total time to failure of 1476 cycles. For V2 chip LEDs, this value decreases to 1140 cycles, and for V3 LEDs, it increases a little bit more to 1196 cycles than V2 LEDs.

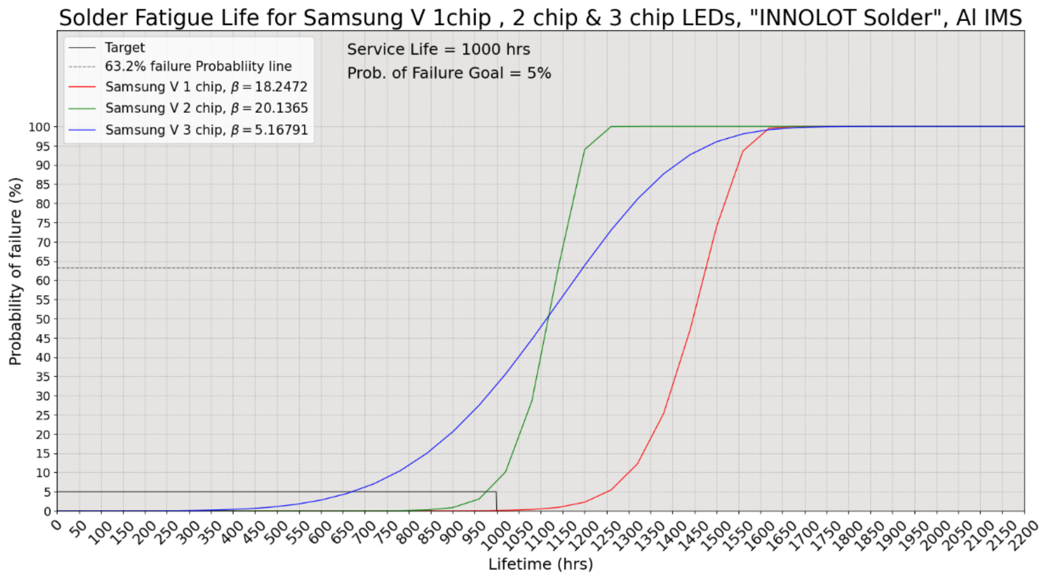


Figure 5.5: Simulation output for Samsung V Series LEDs on Al IMS PCB, Innolot Solder

5.2.1 Simulated vs Real Fatigue Life: Samsung V1, Innolot Solder, Al IMS

For the Samsung V1 chip LED with INNOLOT solder on Aluminum IMS PCB, Figure 5.6 reveals a significant difference between the simulated and actual fatigue life. The simulation estimates a fatigue life of approximately 1476 thermal cycles, whereas the real solder fatigue life is nearly 3490 cycles.

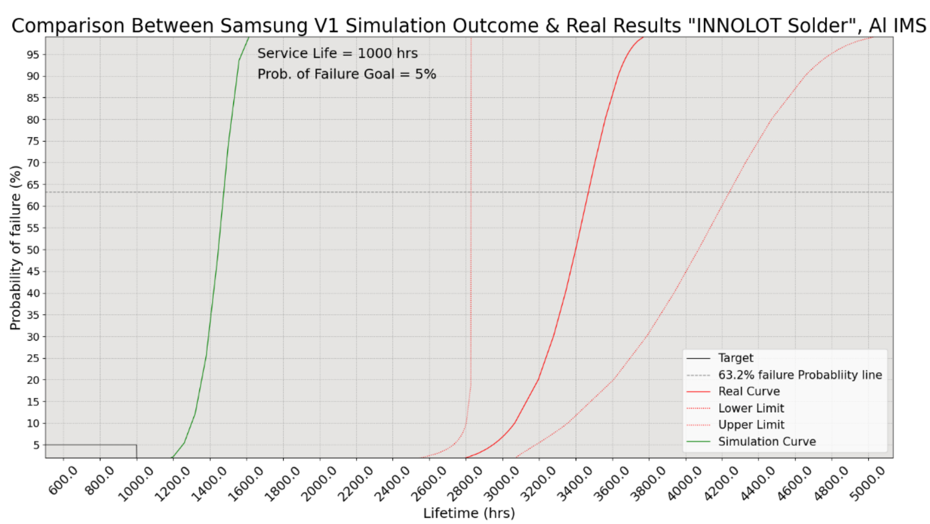


Figure 5.6: Comparison: Samsung V1 on Al IMS, INNOLOT Solder (BTC Model, $\beta = 18.2472$)

5.2.2 Simulated vs Real Fatigue Life: Samsung V2, Innolot Solder, Al IMS

In the case of the Samsung V2 chip LED with INNOLOT solder on Aluminum IMS PCB, Figure 5.7 demonstrates a substantial difference between the simulated and actual fatigue life. The simulation predicts a fatigue life of approximately 1140 thermal cycles, while the actual solder fatigue life is nearly 3200 cycles.

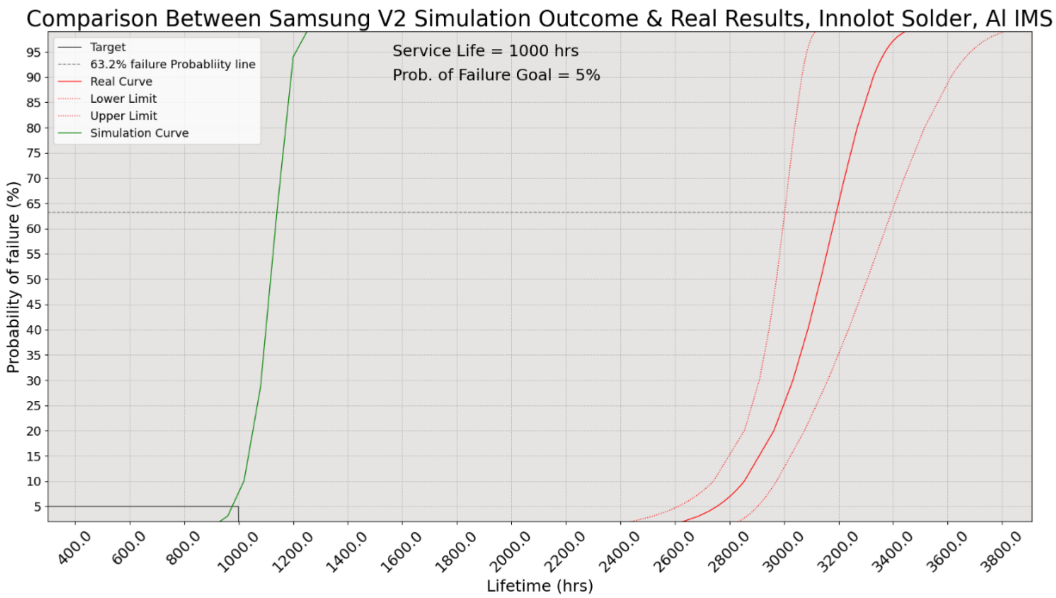


Figure 5.7: Comparison: Samsung V2 on Al IMS, INNOLOT Solder (BTC Model, $\beta = 20.1365$)

5.2.3 Simulated vs Real Fatigue Life: Samsung V3, Innolot Solder, Al IMS

The fatigue life comparison for the Samsung V3 chip LED with INNOLOT solder on Aluminum IMS PCB, as shown in Figure 5.8, indicates a significant difference between the simulated and actual fatigue life. The simulation suggests a fatigue life of approximately 1196 thermal cycles, while the actual solder fatigue life is nearly 4650 cycles.

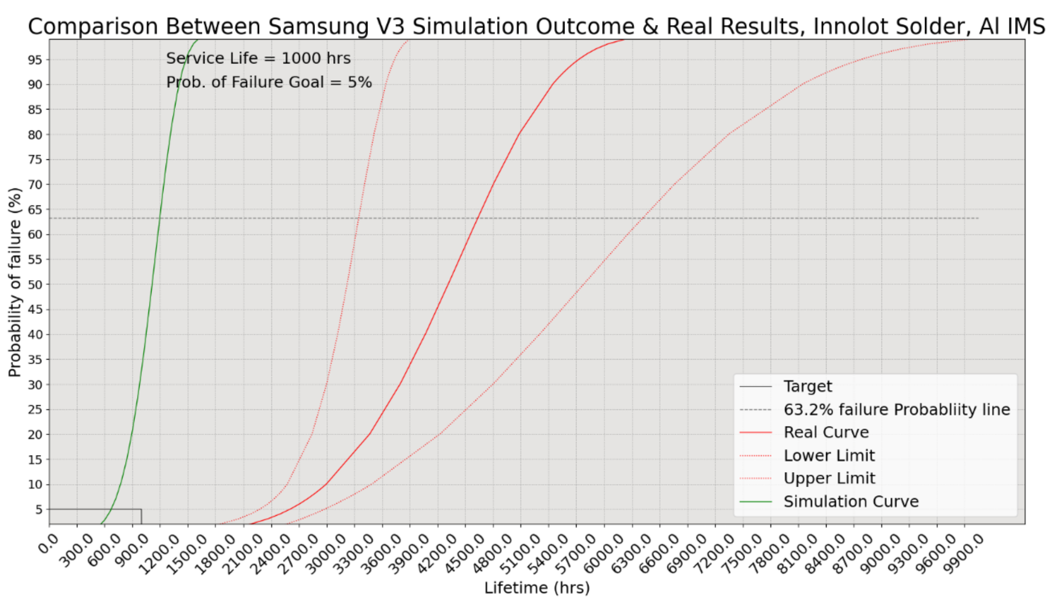


Figure 5.8: Comparison: Samsung V3 on Al IMS, INNOLOT Solder (BTC Model, $\beta = 5.16791$)

5.3 SAC305 Solder, Copper IMS

Figure 5.9 illustrates life prediction curves for Samsung V 1, 2, and 3 chip LEDs, Considering the use of SAC305 solder. Simulation results indicate a total time to failure of 1680 cycles for V1 chip LEDs. This time decreases to 1266 cycles for V2 chip LEDs and further increases a littlebite to 1316 cycles for V3 LEDs.

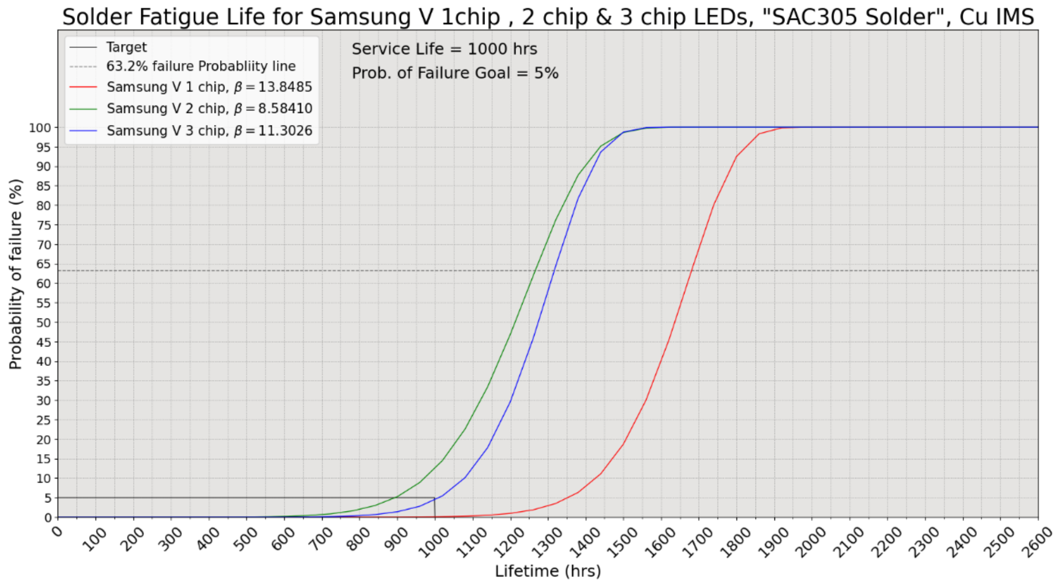


Figure 5.9: Simulation output for Samsung V Series LEDs on Copper IMS PCB, SAC305 Solder

5.3.1 Simulated vs Real Fatigue Life: Samsung V1, SAC305 Solder, Cu IMS

A distinct contrast emerges when examining the fatigue life predictions for the Samsung V1 chip LED with SAC305 solder on Copper IMS PCB, depicted in Figure 5.10. The simulation forecasts a fatigue life of around 1680 thermal cycles, while the observed solder fatigue life stands at nearly 3231 cycles.

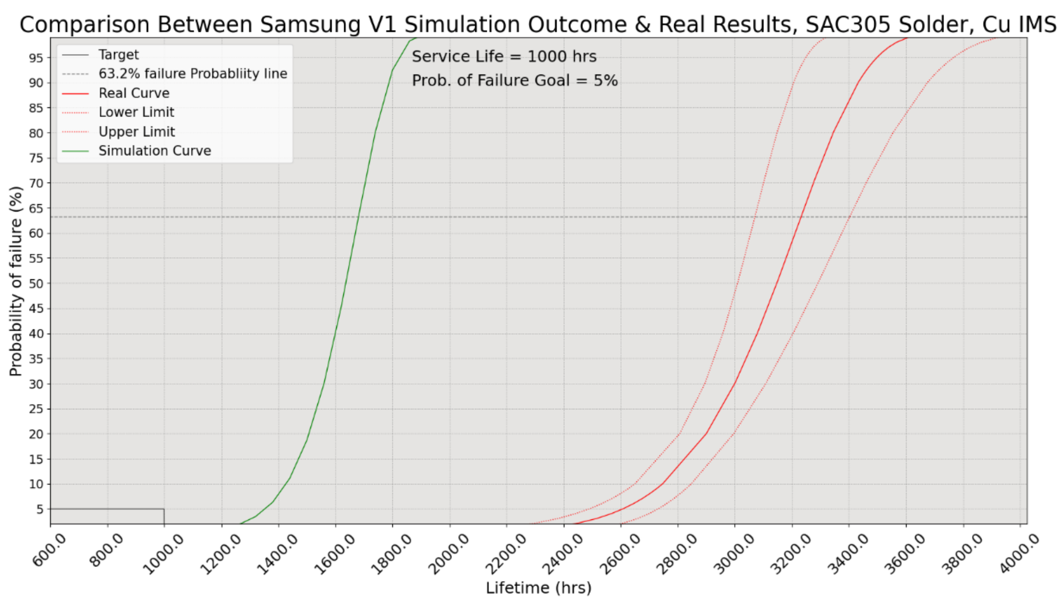


Figure 5.10: Comparison: Samsung V1 on Cu IMS, SAC305 Solder (BTC Model, $\beta = 13.8485$)

5.3.2 Simulated vs Real Fatigue Life: Samsung V2, SAC305 Solder, Cu IMS

Illustrated in Figure 5.11, the fatigue life comparison for the Samsung V2 chip LED employing SAC305 solder on Copper IMS PCB underscores a significant distinction. The simulation suggests a fatigue life of approximately 1266 thermal cycles, as opposed to the actual solder fatigue life, which approaches 3549 cycles.

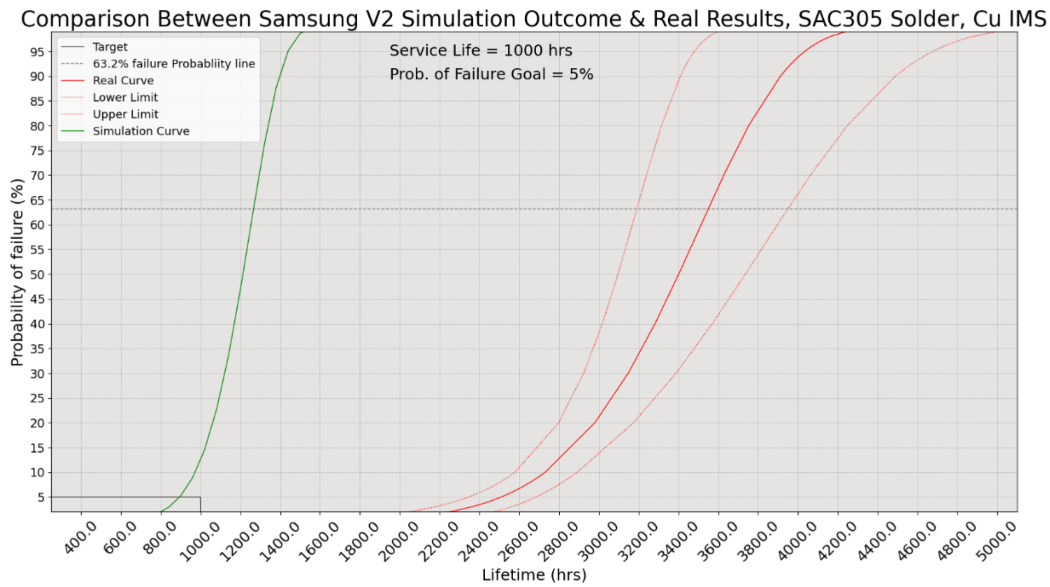


Figure 5.11: Comparison: Samsung V2 on Cu IMS, SAC305 Solder (BTC Model, $\beta = 8.58410$)

5.3.3 Simulated vs Real Fatigue Life: Samsung V3, SAC305 Solder, Cu IMS

Figure 5.12 portrays the fatigue life comparison for the Samsung V3 chip LED featuring SAC305 solder on Copper IMS PCB, emphasizing a notable divergence. The simulation anticipates a fatigue life of about 1316 thermal cycles, while the actual solder fatigue life is nearly 3094 cycles.

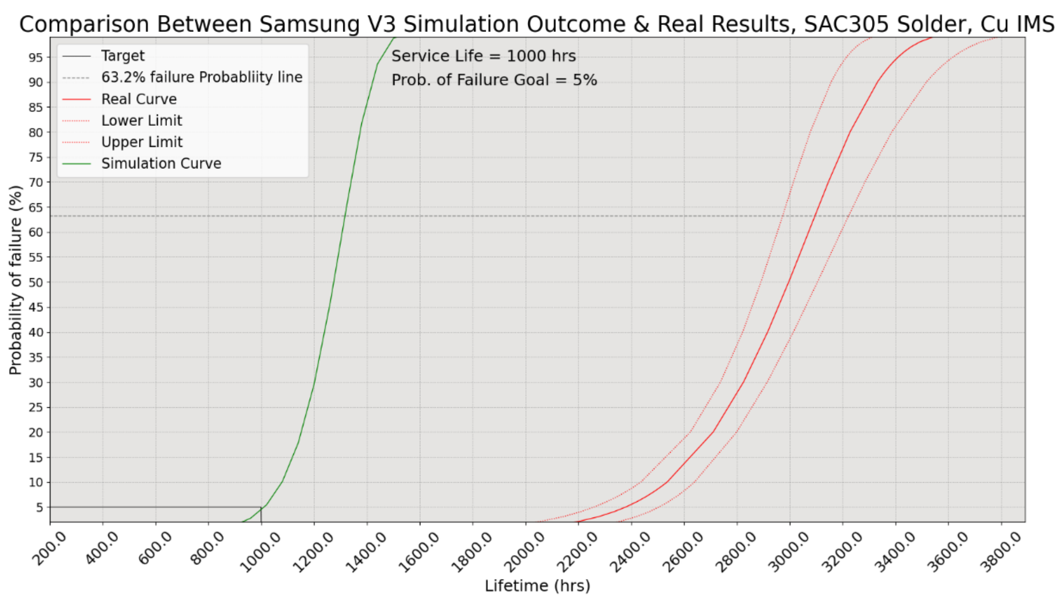


Figure 5.12: Comparison: Samsung V3 on Cu IMS, SAC305 Solder (BTC Model, $\beta = 11.3026$)

5.4 Innolot Solder, Copper IMS

In Figure 5.13, life prediction curves for Samsung V 1, 2, and 3 chip LEDs are presented, considering INNOLOT solder. According to the simulation, V1 chip LEDs demonstrate a total time to failure of 2310 cycles. For V2 chip LEDs, this value decreases to 1734 cycles, and for V3 LEDs, it slightly increases to 1803 cycles.

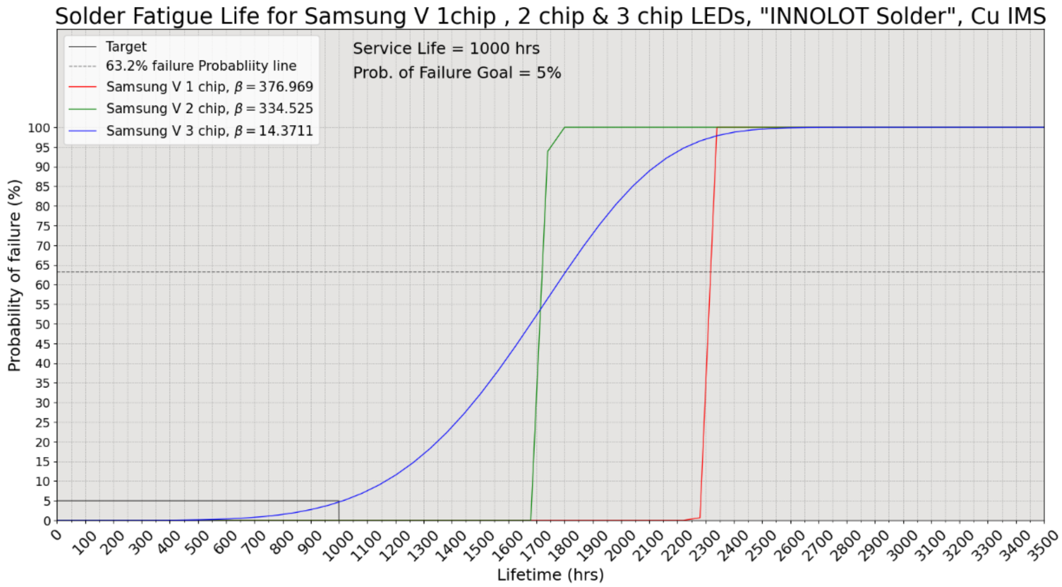


Figure 5.13: Simulation output for Samsung V Series LEDs on Copper IMS PCB, Innolot Solder

5.4.1 Simulated vs Real Fatigue Life: Samsung V1, Innolot Solder, Cu IMS

For the Samsung V1 chip LED utilizing INNOLOT solder on Copper IMS PCB, Figure 5.14 highlights a marked variance between simulated and actual fatigue life. The simulation approximates a fatigue life of about 2310 thermal cycles, whereas the real solder fatigue life extends to almost 3017 cycles.

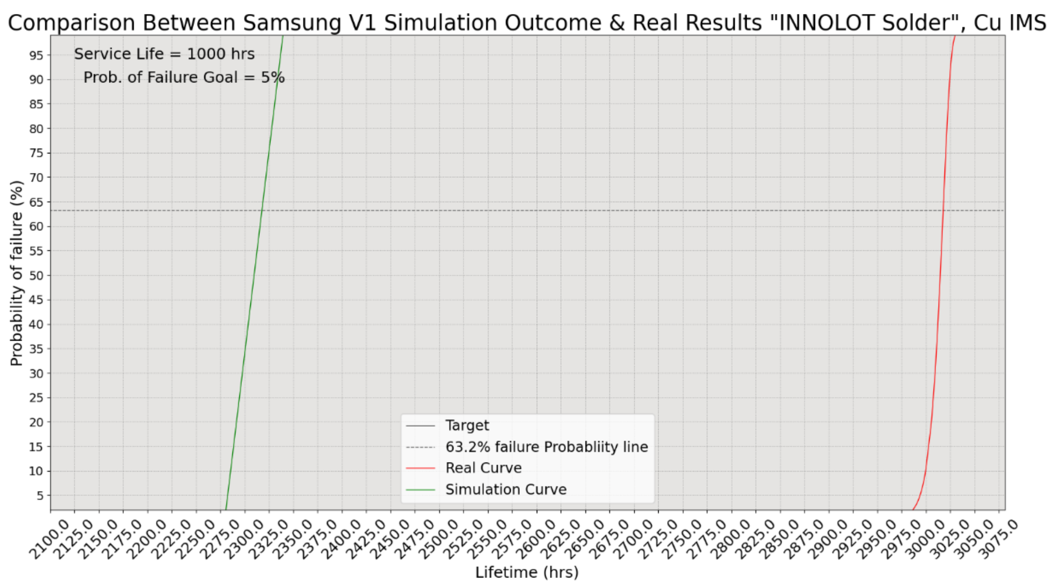


Figure 5.14: Comparison: Samsung V1 on Cu IMS, INNOLOT Solder (BTC Model, $\beta = 376.969$)

5.4.2 Simulated vs Real Fatigue Life: Samsung V2, Innolot Solder, Cu IMS

Examining the Samsung V2 chip LED with INNOLOT solder on Copper IMS PCB, as presented in Figure 5.15, reveals a substantial dichotomy between simulated and actual fatigue life. The simulation projects a fatigue life of approximately 1734 thermal cycles, whereas the genuine solder fatigue life nears 3034 cycles.

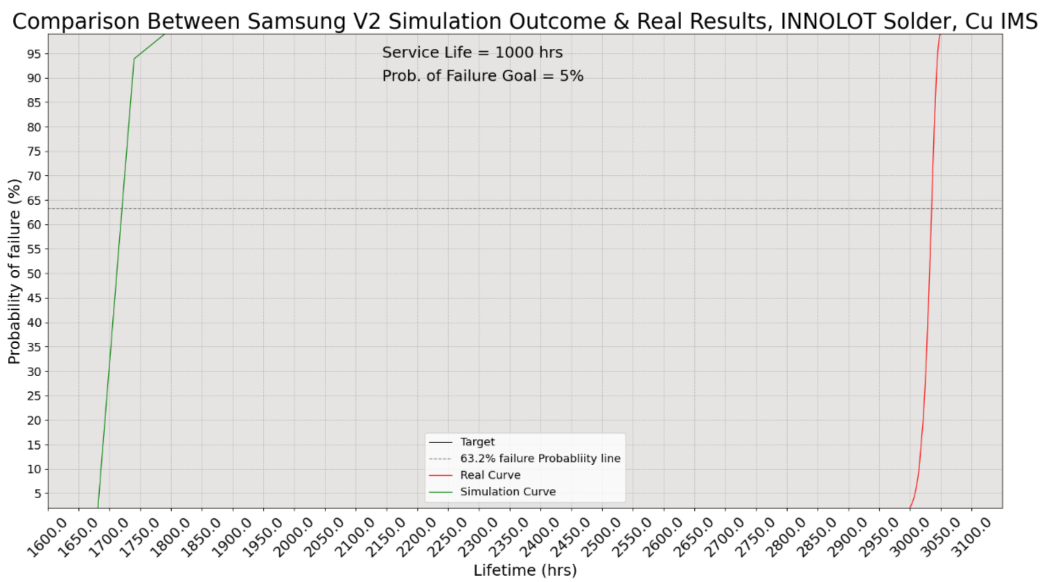


Figure 5.15: Comparison: Samsung V2 on Cu IMS, INNOLOT Solder (BTC Model, $\beta = 334.525$)

5.4.3 Simulated vs Real Fatigue Life: Samsung V3, Innolot Solder, Cu IMS

In the context of the Samsung V3 chip LED with INNOLOT solder on Copper IMS PCB, detailed in Figure 5.16, a significant incongruity surfaces between the simulated and actual fatigue life. The simulation proposes a fatigue life of roughly 1803 thermal cycles, while the authentic solder fatigue life reaches close to 3605 cycles.

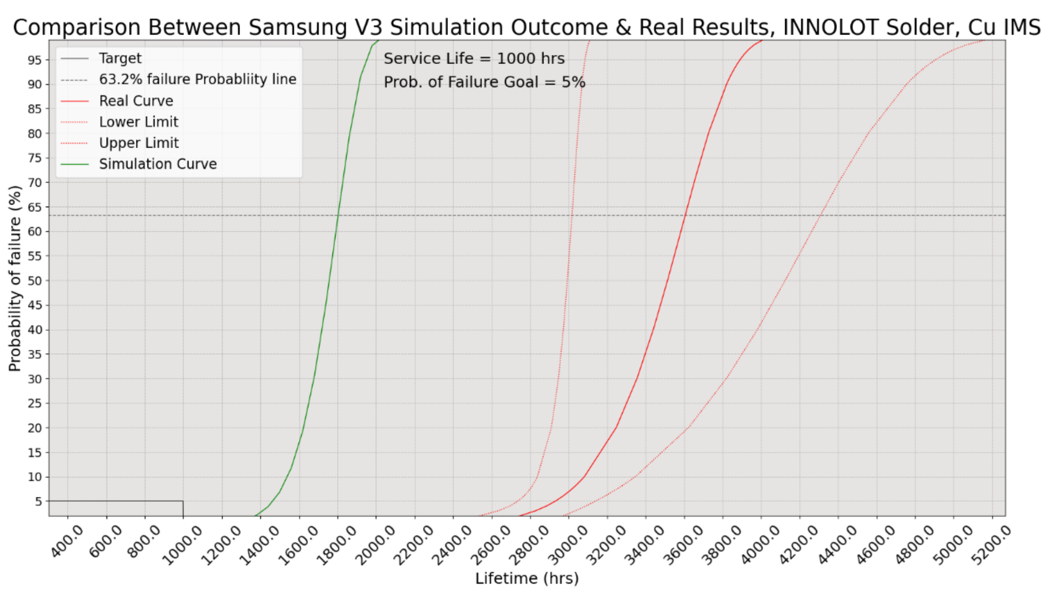


Figure 5.16: Comparison: Samsung V3 on Cu IMS, INNOLOT Solder (BTC Model, $\beta = 14.3711$)

5.5 Conclusion: Simulated vs. Real Fatigue Life

Comparison of the solder fatigue simulation results for Samsung V series LEDs reveals a significant and unacceptable disparity between the simulated and actual test results. For example, when examining Samsung V series LEDs, the simulation consistently yields lower values compared to the real-world results. Specifically, for SAC305 solder on Al IMS PCB, the difference ranges from 67.46% to 73%, while for Innolot solder on Al IMS PCB, the difference ranges from 57.70% to 74.27%. When considering copper PCBs, the difference for SAC305 solder ranges from 48% to 64.32%, and for Innolot solder, it ranges from 23.43% to 49.98%.

According to the BTC model theory in Chapter 3, it can be concluded that the large mismatch may be attributed to the fact that ANSYS Sherlock doesn't consider all of the component solders when calculating the total stiffness. Additionally, in the total inelastic strain energy calculation in Equation 3.32, a multiplication constant $C = 0.5$ is used by Dr. Blattau [34] in his work but is not considered by ANSYS Sherlock in its simulation. Also, due to the LED footprint geometry considered by ANSYS Sherlock in Subsection 3.2.2, the Samsung V Series LED's geometry and its footprint differ, leading to a different constant to be used in Equation 3.26.

It is important to note again that the BTC Theory for ANSYS Sherlock is not officially endorsed by Sherlock Authority; instead, it is adapted from the CC Theory found in the ANSYS Sherlock Theory Reference and informations from various publications. The theory is validated by comparing the manually calculated fatigue life and the simulation results, showing matching results for BTC 35 LED in the ANSYS Sherlock Part library, using SAC305 solder and Al IMS PCB & a reference temperature of 27°C.

Considering all these factors, including the constant $C = 0.5$ in Equation 3.32 and a different multiplication factor in Equation 3.26 for the specific geometric shape of Samsung V series LEDs and their solder footprints, a slightly modified (theoretical) approach to calculate the solder fatigue life of Samsung V Series LEDs solder is carried out in the next chapter, Chapter 6.

Chapter 6

Manual Calculation of Solder Fatigue Life for Samsung V Series LEDs

In this chapter, a manual approach is employed to calculate the mean fatigue life of Samsung V1, V2, and V3 chip LEDs. This calculation takes into account the specific geometry of the solder footprint of Samsung V series LEDs, as opposed to the default geometry referenced as 3.7.

6.1 Theoretical Approach

The geometry of Samsung V Series LEDs is depicted in Figure 6.2. To compute the solder fatigue life of these LEDs, the fatigue life is determined by accounting for stiffness in both the x and y directions. The final fatigue life is then derived by choosing the lower value between the two calculated values. These values are obtained by separately considering stiffness in the x and y directions.

6.1.1 Total Stiffness Calculation

The Total Stiffness calculation corresponds to the same procedure as described in BTC Model Theory in Chapter 3 with a little modification. This time, all the solders (Anode, Cathode, and Thermal Pad) are considered in total stiffness Calculation.

For the foundation stiffness, the formula is:

$$K_f = \frac{9 \cdot G_{\text{pcb}} \cdot a}{2 - V_{\text{pcb}}} \tag{6.1}$$

where G_{pcb} is the shear modulus of the PCB, V_{pcb} is the Poisson ratio of the PCB, and a is the edge-to-edge distance of the bond pad, in this case, the length of the pad (Thermal pad / Anode Pad / Cathode pad).

To calculate the equivalent stiffness, The approach illustrated in Figure 6.1 can be adopted.

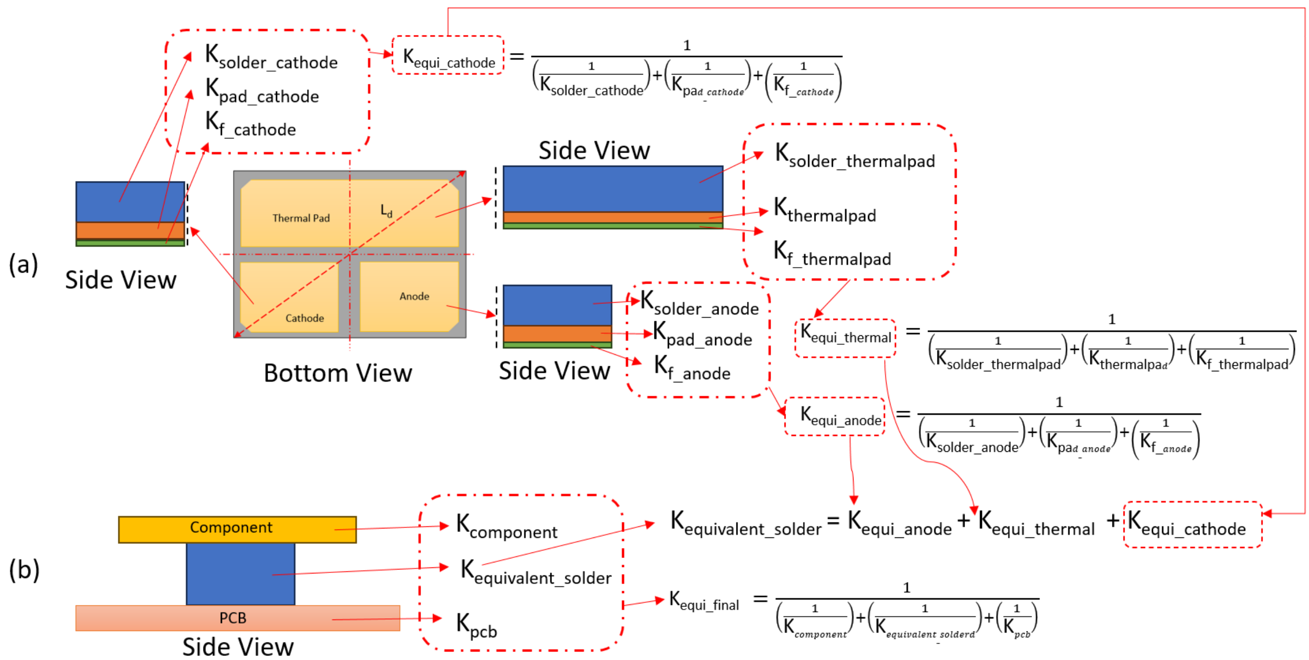


Figure 6.1: Considerations for Neutral-to-Solder Distance in Samsung V Series LEDs

For the Thermal Pad: The stiffness of the thermal pad solder, thermal pad (copper pad), and the foundation stiffness of the thermal pad are in series. Therefore, the equivalent stiffness for these three components is given by:

$$K_{\text{equi\_thermal}} = \frac{1}{\left(\frac{1}{K_{\text{solder\_thermalpad}}} + \frac{1}{K_{\text{thermalpad}}} + \frac{1}{K_{\text{f\_thermalpad}}}\right)} \tag{6.2}$$

where:

$$K_{\text{thermalpad}} = \frac{\text{thermal pad material shear modulus} \cdot \text{thermal pad length} \cdot \text{thermal pad width}}{\text{thermal pad thickness}}$$

$$K_{\text{solder\_thermalpad}} = \frac{\text{thermal pad solder material shear modulus} \cdot \text{thermal pad solder length} \cdot \text{thermal pad solder width}}{\text{thermal pad solder height}}$$

$$K_{\text{f\_thermalpad}} = \frac{9 \cdot \text{shear modulus of PCB} \cdot \text{thermal pad length}}{2 - \text{Poisson ratio of PCB}}$$

Similarly,
For the Anode: The stiffness of the Anode solder, Anode pad (copper pad), and the foundation stiffness of the Anode are in series. The equivalent stiffness for these three components is given by:

$$K_{\text{equi\_anode}} = \frac{1}{\left(\frac{1}{K_{\text{solder\_anode}}} + \frac{1}{K_{\text{pad\_anode}}} + \frac{1}{K_{\text{f\_anode}}}\right)} \tag{6.3}$$

where:

$$K_{\text{solder\_anode}} = \frac{\text{Anode solder material shear modulus} \cdot \text{Anode solder length} \cdot \text{Anode solder width}}{\text{Anode solder height}}$$

$$K_{\text{pad\_anode}} = \frac{\text{Anode pad material shear modulus} \cdot \text{Anode pad length} \cdot \text{Anode pad width}}{\text{Anode pad thickness}}$$

$$K_{\text{f\_anode}} = \frac{9 \cdot \text{shear modulus of PCB} \cdot \text{Anode pad length}}{2 - \text{Poisson ratio of PCB}}$$

and **For the Cathode:** The stiffness of the Cathode solder, Cathode pad (copper pad), and the foundation stiffness of the Cathode are in series. The equivalent stiffness for these three components is given by:

$$K_{\text{equi\_cathode}} = \frac{1}{\left(\frac{1}{K_{\text{solder\_cathode}}} + \frac{1}{K_{\text{pad\_cathode}}} + \frac{1}{K_{\text{f\_cathode}}}\right)} \tag{6.4}$$

where:

$$K_{\text{solder\_cathode}} = \frac{\text{Cathode solder material shear modulus} \cdot \text{Cathode solder length} \cdot \text{Cathode solder width}}{\text{Cathode solder height}}$$

$$K_{\text{pad\_cathode}} = \frac{\text{Cathode pad material shear modulus} \cdot \text{Cathode pad length} \cdot \text{Cathode pad width}}{\text{Cathode pad thickness}}$$

$$K_{\text{f\_cathode}} = \frac{9 \cdot \text{shear modulus of PCB} \cdot \text{Cathode pad length}}{2 - \text{Poisson ratio of PCB}}$$

Now, $K_{\text{equi\_thermal}}$, $K_{\text{equi\_anode}}$ and $K_{\text{equi\_cathode}}$ are in parallel to each other, so the equivalent stiffness of these three is:

$$K_{\text{equivalent\_solder}} = K_{\text{equi\_thermal}} + K_{\text{equi\_anode}} + K_{\text{equi\_cathode}} \tag{6.5}$$

Finally, $K_{\text{component}}$, $K_{\text{equivalent\_solder}}$, and K_{pcb} are in series. Therefore, the final stiffness is:

$$K_{\text{equi\_final}} = \frac{1}{\left(\frac{1}{K_{\text{component}}} + \frac{1}{K_{\text{equivalent\_solder}}} + \frac{1}{K_{\text{pcb}}}\right)} \tag{6.6}$$

where,

$$K_{\text{component}} = \frac{\text{Elastic modulus of Component} \times \text{width of component} \times \text{thickness of component}}{\text{Length of component}}$$

and

$$K_{\text{pcb}} = \frac{\text{Elastic Modulus of PCB} \times (\text{width of PCB}) \times \text{Thickness of PCB}}{\text{Length of PCB}}$$

here , total length and width of PCB is considered.

6.1.2 Maximum Strain range calculation

The Maximum Strain Range calculation for Samsung V Series LEDs follows the same approach as outlined in the BTC Theory in section 3.2.2 in Chapter 3. Here, the separation of hot and cold parts is also considered. A slight modification to Equation 3.26 for total axial displacement and Equation 3.32 for total inelastic strain energy has been made. Below is the detailed procedure for calculating the maximum strain range for the hot part. The same procedure is followed for the cold part.

T_{mean} represents the reference temperature, at which the dimensions of PCBs, components, solders, copper pads, etc., are measured. In the context of calculating Solder Fatigue life for Samsung V Series LEDs, $T_{\text{mean}} = (30.445 + 273.15)°\text{K}$ is considered as the reference temperature, the same as used in the CC Model Simulation of resistors R14 and R15, as discussed in section 3.1 in Chapter 3. The same Thermal Cycle in the Solder Fatigue Simulation of Samsung V Series LEDs in Figure 4.2 Chapter 4 is also considered here.

T_{max} = maximum temperature of the thermal cycle.
T_{min} = minimum temperature of the thermal cycle.

$\Delta T_{\text{hot}} = T_{\text{max}} - T_{\text{mean}}$ $\Delta T_{\text{cold}} = T_{\text{mean}} - T_{\text{min}}$

for the hot part , total stiffness is ,

$$K^{\text{combined}}_{\Delta T_{\text{hot}}} = K_{\text{equi\_final}}$$

where $K_{\text{equi\_final}}$ is the same as in Equation 6.6. The only difference is that, while calculating the stiffness of each solder, the shear modulus of the solder material at the maximum temperature is considered for the hot part.

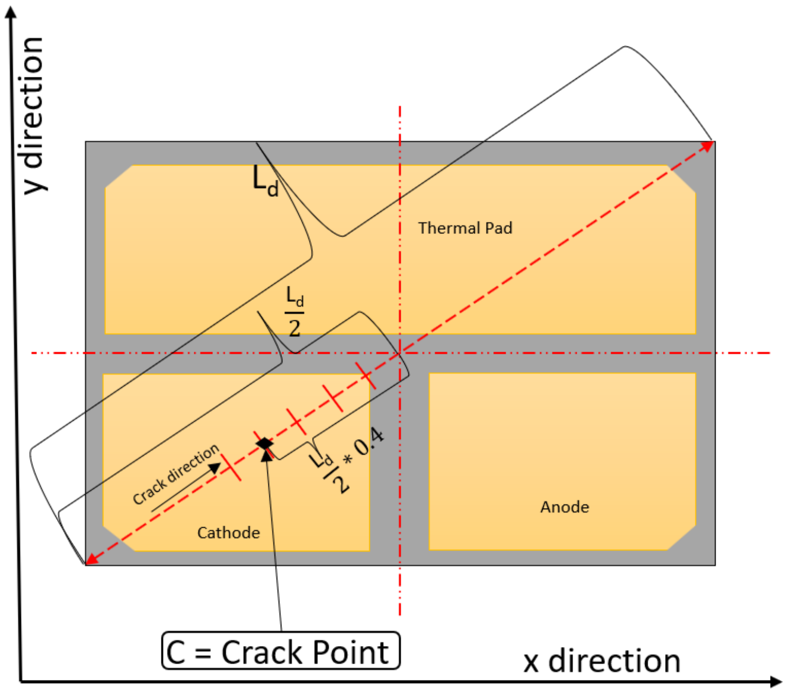


Figure 6.2: Equivalent Stiffness Calculation for Samsung V Series LEDs

First, the total axial displacement $U^{\text{total}}_{\Delta T_{\text{hot}}}$ is calculated using the following formula:

$$U^{\text{total}}_{\Delta T_{\text{hot}}} = |CTE_{\text{comp}} - CTE_{\text{pcb}}| \cdot \Delta T_{\text{hot}} \cdot \left(\frac{L_{\text{D}}}{2} \cdot C\right) \tag{6.7}$$

The term $\left(\frac{L_{\text{D}}}{2} \cdot C\right)$ represents the distance from the neutral point to the solder crack location for the geometric shape of the LED (including its footprint), as illustrated in Figure 6.2. $C = \frac{1}{2}$ is the expected consideration when the crack point is assumed to be at the middle of the solder. However, for Samsung V Series LEDs, the solder shapes are not precisely rectangular; there is a small cut in the rectangular shapes of the Anode, Cathode, and Thermal Pad on the outer corner. Considering this aspect, a value of $C = \frac{4}{10} = 0.4$ is adopted instead of $\frac{1}{2}$. The crack point represents the location where, if a crack occurs, the entire solder is considered to be cracked. This point varies with geometry. Although $C = 0.4$ serves as an initial estimate, it is recognized that obtaining a more precise value requires additional geometric analysis based on the crack point. Due to time constraints, $C = 0.4$ is provisionally accepted.

The differential force generated due to the CTE mismatch of the component and the PCB can be attained with the following equation:

$$F = U^{\text{total}}_{\Delta T_{\text{hot}}} \cdot K^{\text{combined}}_{\Delta T_{\text{hot}}} \tag{6.8}$$

Elastic portion of the stress and strain are determined by the following two equations:

$$\sigma^{\text{e}}_{\Delta T_{\text{hot}}} = \frac{F}{A_{\text{solder}}} = \frac{U^{\text{total}}_{\Delta T_{\text{hot}}} \cdot K^{\text{combined}}_{\Delta T_{\text{hot}}}}{L_{\text{solder}} \cdot W_{\text{solder}}} \tag{6.9}$$

$$\epsilon^{\text{e}}_{\Delta T_{\text{hot}}} = \frac{\sigma^{\text{e}}_{\Delta T_{\text{hot}}}}{G^{\text{solder}}_{\text{T}_{\text{max}}}} \tag{6.10}$$

Now, plastic strain can be calculated with the following equation:

$$\epsilon^{\mathsf{p}}_{\Delta T_{\mathsf{hot}}} = \epsilon^{\mathsf{t}}_{\Delta T_{\mathsf{hot}}} - \epsilon^{\mathsf{e}}_{\Delta T_{\mathsf{hot}}} \tag{6.11}$$

where elastic strain $\epsilon^{\mathsf{t}}_{\Delta T_{\mathsf{hot}}} = \frac{U^{\mathsf{total}}_{\Delta T_{\mathsf{hot}}}}{h_{\mathsf{solder}}}$

Now, inelastic (plastic) strain and Engelmeier equation are used to correct the maximum strain considering viscoplastic creep effects. The corrected maximum strain range is shown below:

$$\epsilon^{\mathsf{max}}_{\Delta T_{\mathsf{hot}}} = \epsilon^{\mathsf{p}}_{\Delta T_{\mathsf{hot}}} \cdot \left(\frac{t_{\mathsf{dwell\text{-}max}}}{t_{\mathsf{critical}}}\right)^{0.136} \cdot e^{E_{\mathsf{a}} \cdot \left(\frac{1}{T_{\mathsf{critical}}} - \frac{1}{T_{\mathsf{max}+273.15}}\right)} \tag{6.12}$$

where

$t_{\mathsf{dwell\text{-}max}}$ = Maximum time the solder stays at a specific temperature

t_{critical} = Critical time. In Sherlock, a critical time of 360 minutes is used. This same is utilized in manual calculations as well.

T_{critical} = Critical temperature in Kelvin(K), which is half of the melting point of the solder. For SAC305 solder, the melting temperature is 217°C, so half of this is 108.5°C. However, Sherlock considers 105°C as the critical temperature for SAC305 solder. The same is adopted here.

E_{a} = Activation energy of the solder. Sherlock considers the original activation energy of the solder divided by the Boltzmann constant K, and the result is denoted as E_{a}. in Manual Calculation , an activation energy of 2185 eV is considered for SAC305 solder alloy.

Following the same procedure, cold part maximum strain range $\epsilon^{\mathsf{max}}_{\Delta T_{\mathsf{cold}}}$ is calculated.

Now, total Inelastic Strain energy can be obtained by the following equation:

$$\Delta W = C \cdot \left(\epsilon^{\mathsf{max}}_{\Delta T_{\mathsf{hot}}} \cdot \sigma^{\mathsf{e}}_{\Delta T_{\mathsf{hot}}} + \epsilon^{\mathsf{max}}_{\Delta T_{\mathsf{cold}}} \cdot \sigma^{\mathsf{e}}_{\Delta T_{\mathsf{cold}}}\right) \tag{6.13}$$

here C = $\frac{1}{2}$ assumming the hysteresis loop is roughly equilateral [14]

6.1.3 Fatigue Life (N_f) Calculation

The mean fatigue life N_f is calculated from the obtained inelastic strain energy with the following formula:

$$\text{Mean fatigue life } N_f = \left((SE)_{\mathsf{coeff}} \cdot C3 \cdot \Delta W\right)^{-1} \tag{6.14}$$

where $(SE)_{\mathsf{coeff}}$ is the Strain Energy Coefficient, which is 0.001 in Sherlock by default for SAC305,same is used in manual calculation. $C3$ is an empirical constant used by Sherlock. It can be obtained from Table 3.2 in the ANSYS Sherlock theory reference.

6.2 Manually Calculated Fatigue Life (N_f) vs Real Fatigue Life

Due to time constraints, manual calculation for INNOLOT Solder was not feasible. However, the calculation was executed using SAC305 Solder for both Aluminum and Copper IMS PCBs. The comparative results are presented in Table 6.1.

The Python code blocks for these calculations are available in the Appendix.

Table 6.1: Comparison of Samsung V Series LEDs: Manually Calculated Mean TTF versus Real Lab Measured TTF

LED	Solder	PCB	Real Lab Measured Mean TTF	Manually Calculated Mean TTF	Difference
Samsung V 1 chip	SAC305	Al IMS	3047	3081	+1.12%
		Cu IMS	3113	2497	-19.79%
Samsung V 2 chip		Al IMS	2929	2731	-6.76%
		Cu IMS	3353	2094	-37.55%
Samsung V 3 chip		Al IMS	3048	2806	-7.94%
		Cu IMS	2958	2149	-27.35%

6.3 Conclusion: Manual Calculation of Solder Fatigue Life for Samsung V Series LEDs

It is noticeable in Table 6.1, For Aluminum IMS PCBs, the manually calculated results closely align with the real results, exhibiting minimal differences. Conversely, this congruence is not as evident for Copper IMS PCBs. A disparity of more than 25% is observed for Samsung V2 and V3 chip LEDs. During an internal discussion with Sherlock Ansys authorities, it was disclosed that the Sherlock model may not accurately predict outcomes for Copper IMS. In the manual calculation process, the elastic modulus and Poisson's ratio of the PCB (Sherlock assumes a Poisson's ratio of 0.18 for all types of PCB) are directly obtained from the Sherlock software. The specific methodology employed by Sherlock to calculate these values, particularly the elastic modulus of the PCB, is not fully revealed. It is plausible that the formula utilized by Sherlock for this purpose is not well-suited for Copper IMS PCBs. Further analysis on this matter is pending.

Chapter 7

LED Modification Effects on Solder Fatigue Reliability

In this chapter, the focus is on modifying the geometrical shape and dimensions of LEDs to examine their impact on the reliability of their solder joints.
This additional task aims to assess the impact of modifications on reliability, specifically how changes in parameters affect fatigue life behavior. It's worth noting that the Ansys Sherlock Simulation results do not align with real-world observations, as discussed in Chapter 5.
The simulations utilize an Aluminum and Copper Insulated Metal Substrate (IMS) test board crafted for the Samsung C model LEDs during a previous internship, as detailed in Tables 7.1 & 7.2. This specific setup is chosen to save time and effort, assuming that creating a new setup wouldn't yield significant differences. The packaging material considered is Aluminum Nitride (AlN) Ceramic with a Coefficient of Thermal Expansion (CTE) of 4.5 ppm/°C, as illustrated in Figure 4.3. The final fatigue life is quantified with a Thermal Shock Factor (TSF) of 1, which can be modified, as shown in Figure 4.5, following the specified Thermal Profile outlined in Figure 4.2. The solder model used is SAC305 solder, and the solder corresponds to the BTC (Bottom Terminated Components) model.

Table 7.1: Aluminum IMS PCB Stackup Consideration

Layer	Type	Material	Thickness
1	Signal	Copper (21.6% / RM_L1)	70 microns
2	Dielectric	Bergquist MP (Considering T_g)	76 microns
3	Substrate	Aluminum	1.57 mm

Table 7.2: Copper IMS PCB Stackup Consideration

Layer	Type	Material	Thickness
1	Signal	Copper (21.6% / RM_L1)	35 microns
2	Dielectric	Panasonic Ecool-M R-15T1	75 microns
3	Substrate	Copper	1.04 mm

7.1 OSRAM Model LED

7.1.1 Modification Concepts

The objective is to broaden and geometrically transform the LED into a more rectangular shape. Illustrated in Figure 7.1, two distinct concepts are elucidated. Concept 1 involves augmenting the LED's width and simultaneously thinning its profile, resulting in a broader configuration for both the thermal pad and the anode-cathode components. Concept 2 adheres to the same widened and thinner design of Concept 1; however, it introduces four Leads instead of three.Two of these solder pads showcase an expanded thermal pad, while the other two present reduced anode-cathode pairs in dimension.

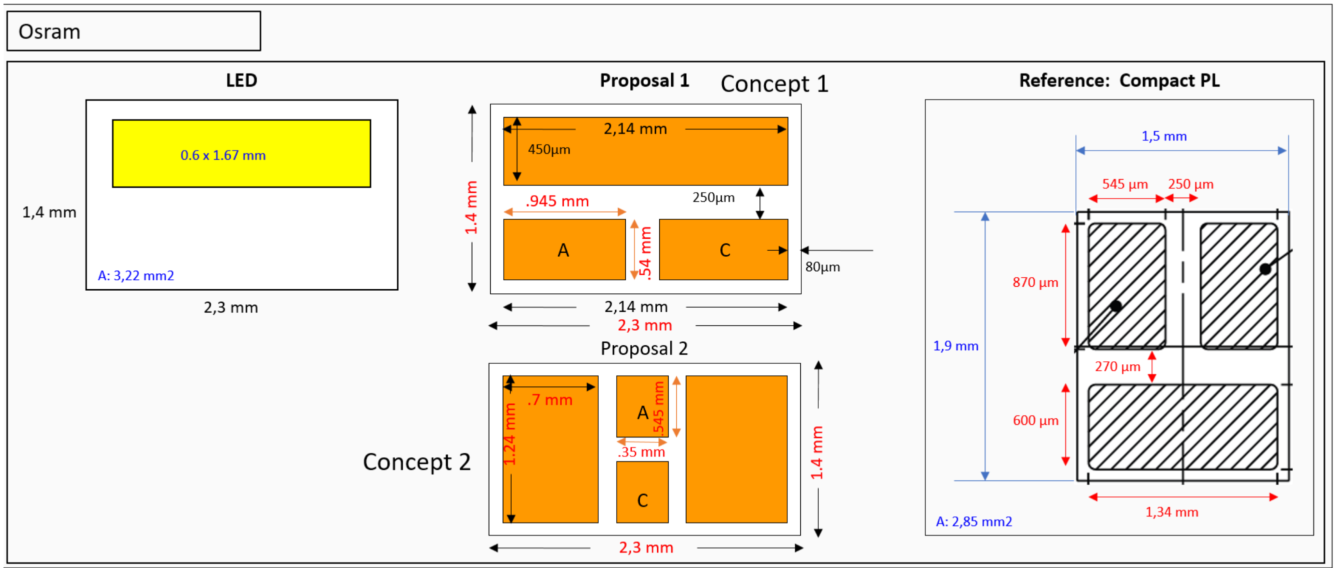


Figure 7.1: Concepts for Modifying OSRAM Model LED [16]

7.1.2 Configuration of Part Properties

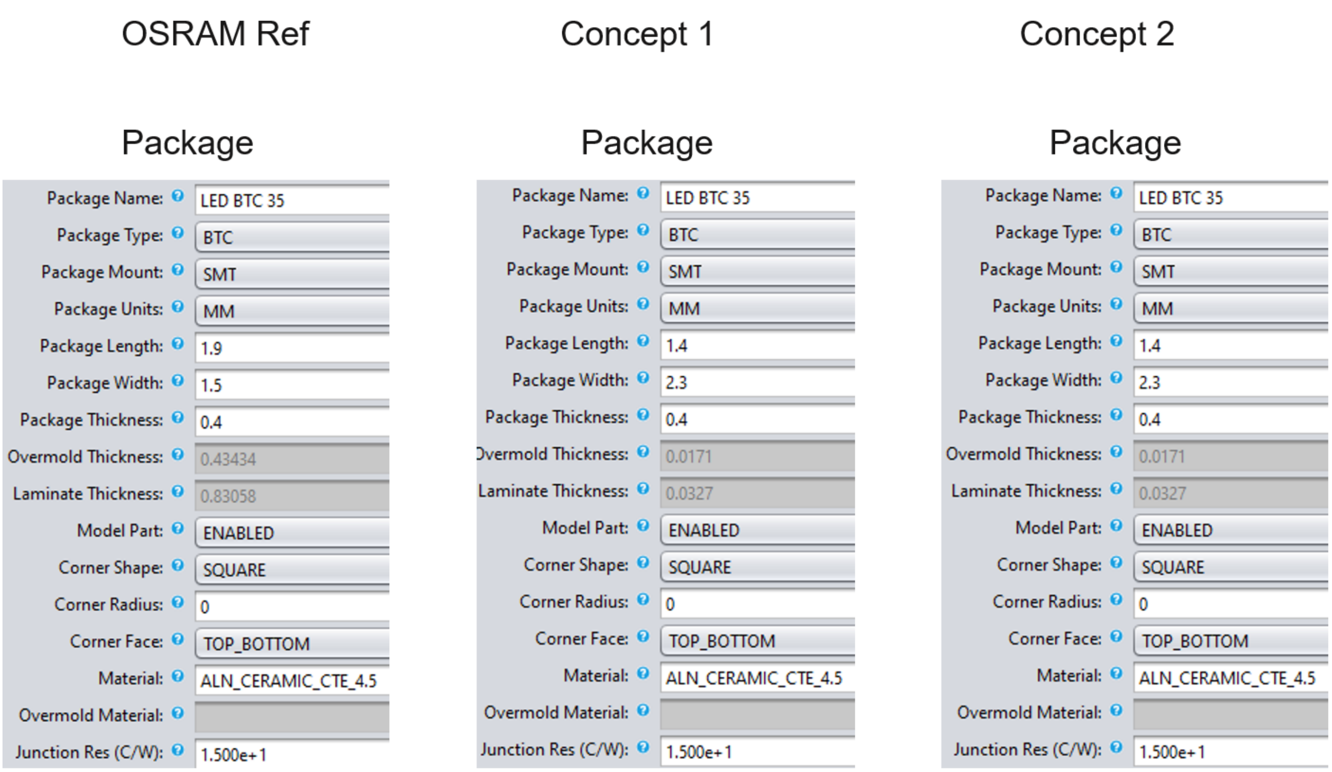


Figure 7.2: Package Property Setting in ANSYS Sherlock for Various Concepts and the Reference OSRAM LED

In Figure 7.2, the package properties employed for simulating solder fatigue in ANSYS Sherlock are presented. Additionally, Figure 7.3 demonstrates the configuration of various other part properties for the simulation.

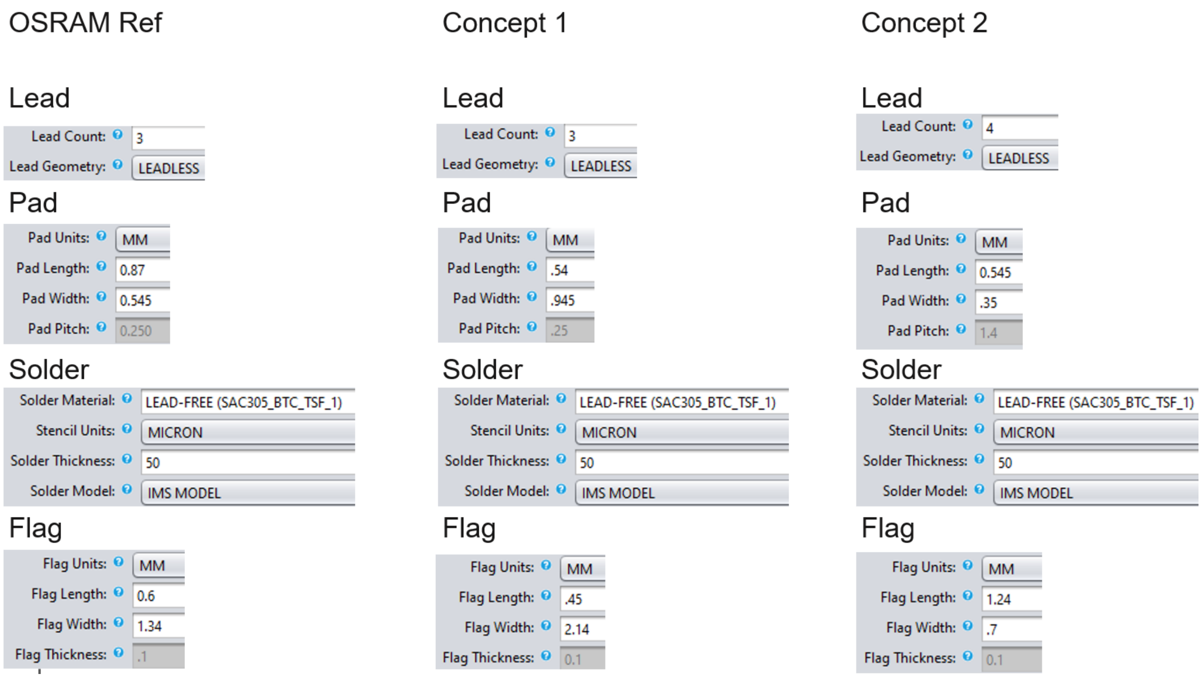


Figure 7.3: Different Part Property Setting in ANSYS Sherlock for Various Concepts and the Reference OSRAM LED

7.2 NICHIA Model LED

7.2.1 Modification Concepts

The objective is to enhance the LED by expanding and geometrically reshaping it into a more rectangular form. Figure 7.4 outlines two distinct concepts. Concept 1 involves increasing the number of solder pads, deviating from the reference LED with two solder pads to include four solder pads. While both Concept 1 and Concept 2 share similarities, Concept 1 exhibits a more pronounced rectangular shape compared to Concept 2.

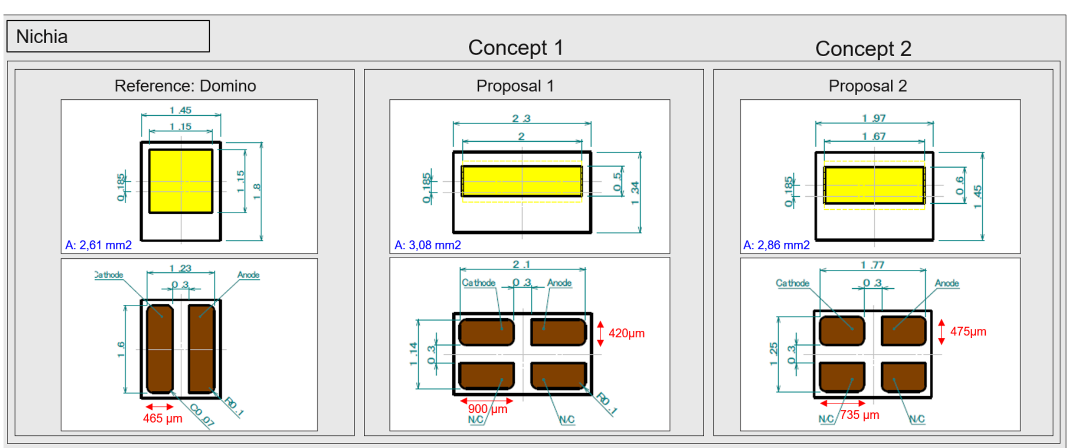


Figure 7.4: Concepts for Modifying NICHIA Model LED [16]

7.2.2 Configuration of Part Properties

Figure 7.5 illustrates the package properties configured for solder fatigue simulation in ANsys Sherlock. The settings specifically pertain to the package used in the analysis. Conversely, Figure 7.6 showcases the setup for other part properties associated with the simulation.

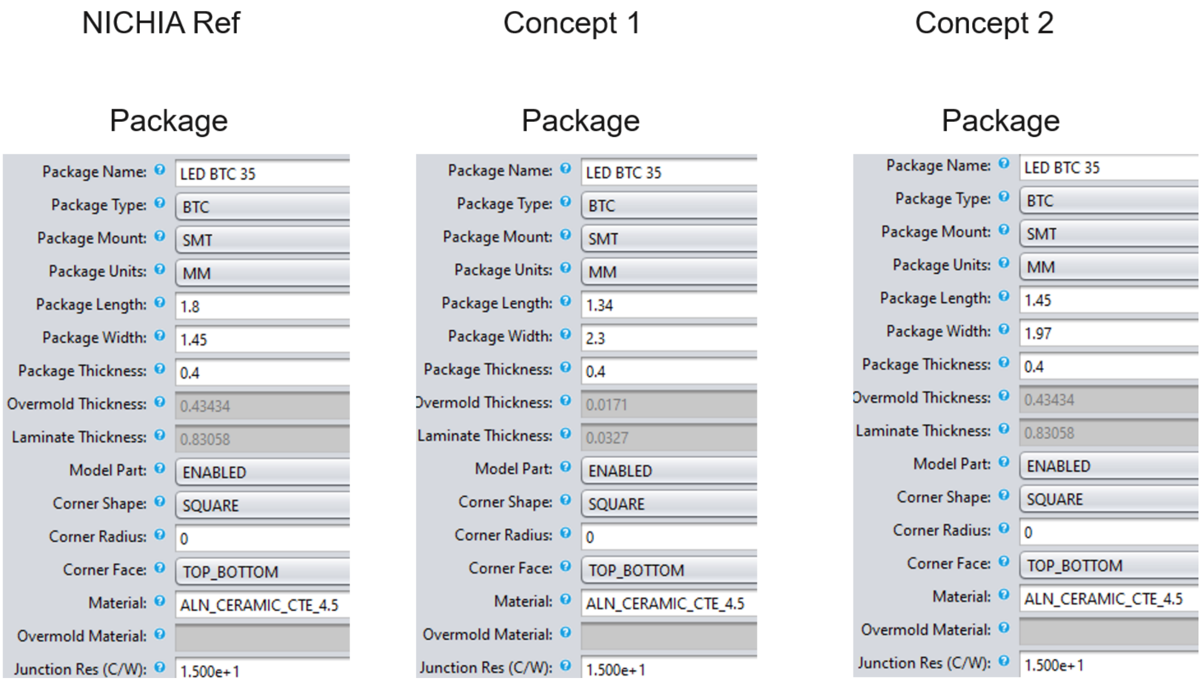


Figure 7.5: Configuration of Package Properties in ANSYS Sherlock for various concepts and the reference LED.

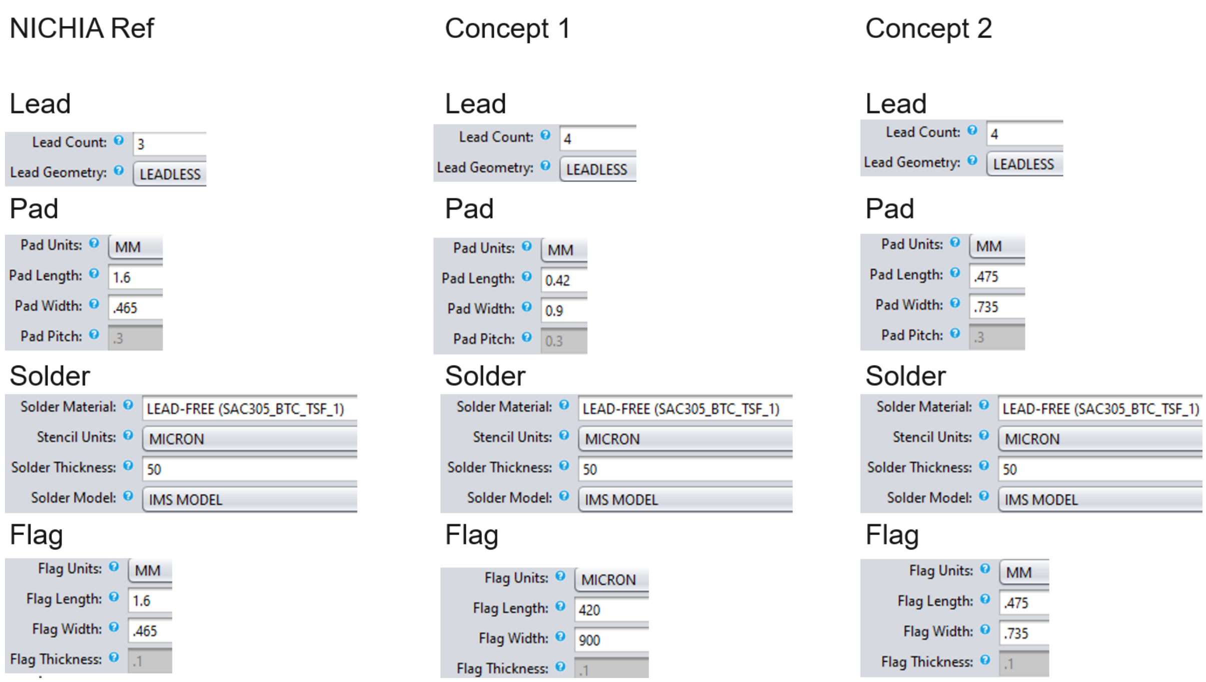


Figure 7.6: Configuration of Different Part Properties in ANSYS Sherlock for various concepts and the reference NICHIA LED

7.3 Simulation Results

The ANSYS Sherlock simulation provides a graphical representation of the probability of failure on the y-axis ranging from 0% to 100%, with lifetime (hours) on the x-axis. The Total Time to Failure (TTF) data can be extracted from the simulation's .csv output.

7.3.1 Comparison of OSRAM LED solder Fatigue Life on Aluminum IMS PCB for two different concepts

In Figure 7.7, the reliability curves compare a reference OSRAM LED with two conceptual LED designs on an Aluminum IMS PCB. The reference LED's TTF is 1341 thermal cycles (or hours, given the setup of 1 hour = 1 thermal cycle). Concept 1 reduces the TTF to 754, while Concept 2 further decreases it to 395.
By exchanging the pad dimension with the flag dimension in Concept 2, the fatigue life of Concept 2 extends to a TTF of 1539.

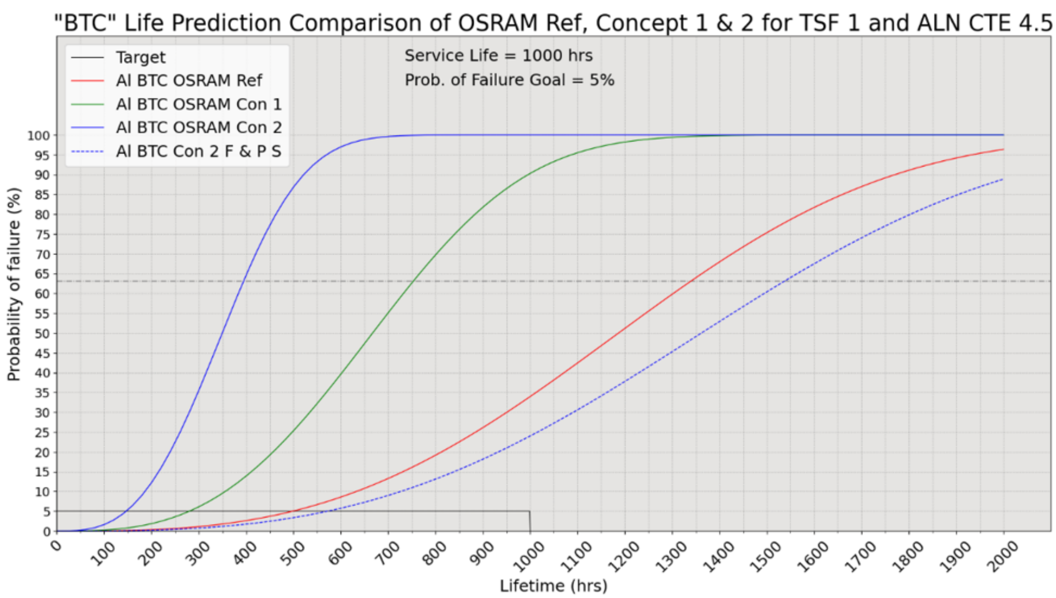


Figure 7.7: Simulation output for different concepts of OSRAM LED on Aluminum IMS PCB

7.3.1.1 Comparison of OSRAM LED solder Fatigue Life on Copper IMS PCB for two different concepts

Figure 7.8 illustrates the reliability curves for a reference OSRAM LED and two conceptual LED designs on a Copper IMS PCB. The reference LED's TTF is 1124 thermal cycles (or hours), Concept 1 reduces the TTF to 591, while Concept 2 significantly decreases it to 337.

By interchanging the pad dimension with the flag dimension in Concept 2, the fatigue life of Concept 2 extends to a TTF of 1646.

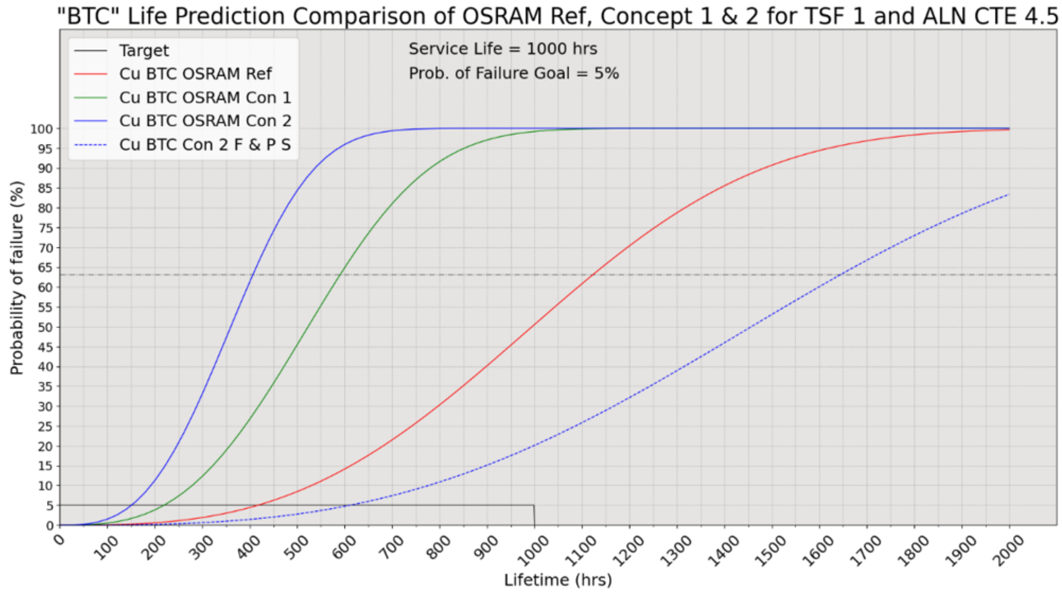


Figure 7.8: Simulation output for different concepts of OSRAM LED on Copper IMS PCB

7.3.2 Comparison of NICHIA LED solder Fatigue Life on Aluminum & Copper IMS PCB for two different concepts

In Figure 7.9, reliability curves are presented for a reference NICHIA LED and two conceptual LED designs on an Aluminum & Copper IMS PCB.
for Al IMS PCB, The reference LED exhibits a Time-to-Failure (TTF) of 2755 thermal cycles (or hours, considering 1 hour = 1 thermal cycle). Concept 1 reduces the TTF to 571, while Concept 2 reduces it to 729 from the TTF of the reference OSRAM LED.

for Cu IMS PCB, The reference LED exhibits a Time-to-Failure (TTF) of 2902 thermal cycles (or hours, considering 1 hour = 1 thermal cycle). Concept 1 reduces the TTF to 419, while Concept 2 reduces it to 545 from the TTF of the reference NICHIA LED.

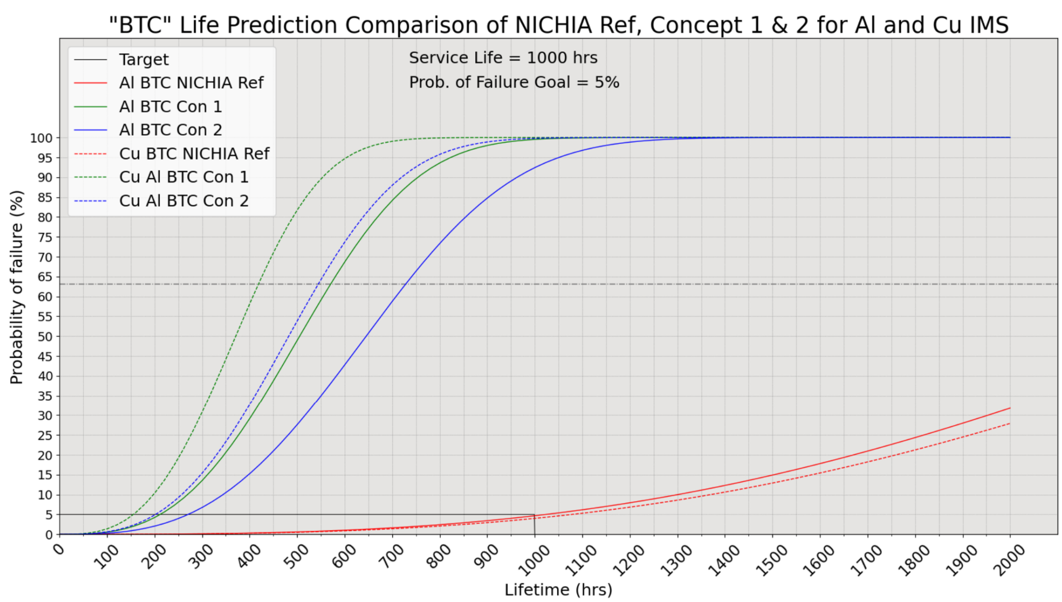


Figure 7.9: Reliability comparison of NICHIA LED for different concepts

7.4 Conclusion: LED Modification Effects on Solder Fatigue Reliability

Although the simulation was executed correctly for the OSRAM reference LED and its first concept, there is reasonable suspicion regarding the accuracy of the simulation for its second modification concept. This suspicion arises from the fact that concept 2 features four solder pads, a geometrical configuration not feasible to model in Ansys Sherlock. The only possibility is to set the lead number to be four. Similarly, the same issue arises with the NICHIA model LEDs reference. The reference, Domino LED, has two pads, yet both of its concepts feature four pads. For a two-pad configuration, Ansys Sherlock does not accept a lead count input 2 for BTC simulation, thus the simulation was performed by providing a lead count of three. Likewise, for the two concepts featuring four-pad configurations, the simulation was conducted by providing a lead count of four.

However, this chapter was undertaken as additional work and is not the main focus of this thesis. Despite the suspicion, the findings are as follows:

OSRAM LEDs on Aluminum IMS PCBs:

- Concept 1 reduces the TTF to 43.7% of the reference LED's TTF.
- Concept 2 reduces the TTF to 70.54% of the reference LED's TTF.
- Interchanging pad and flag dimensions in Concept 2 increases the TTF by 14.77% compared to the reference LED.

OSRAM LEDs on Copper IMS PCBs:

- Concept 1 reduces the TTF to 47.4% of the reference LED's TTF.
- Concept 2 reduces the TTF to 70.01% of the reference LED's TTF.
- Interchanging pad and flag dimensions in Concept 2 increases the TTF by 46.44% compared to the reference LED.

NICHIA LEDs on Aluminum IMS PCBs:

- Concept 1 reduces the TTF to 79.27% of the reference LED's TTF.
- Concept 2 reduces the TTF to 73.53% of the reference LED's TTF.

NICHIA LEDs on Copper IMS PCBs:

- Concept 1 reduces the TTF to 85.56% of the reference LED's TTF.
- Concept 2 reduces the TTF to 81.21% of the reference LED's TTF.

Chapter 8

Summary and Outlook

Throughout this study, a comprehensive investigation is conducted to assess the applicability of the simulation software ANSYS Sherlock in predicting the solder fatigue life of BTC (Bottom Terminated Component) solder. Following the introductory chapter 1, Chapter 2 delves into a mathematical explanation and literature review of related work.

Drawing on the CC Model theory from ANSYS theory references and information available in publications, a theory for the BTC model is developed and validated in Chapter 3 to understand how BTC solder simulation operates in ANSYS Sherlock. ANSYS Sherlock doens't provide explicit theories for simulating Bottom Terminated Component (BTC) solder. However, based on theories for Ceramic Component (CC) like SMD (surface mounted device) Resistor's solder, which suggest not using the solder's meniscus, and the fact that BTC solder doesn't have a meniscus either, it's likely that the solder theory for BTC is the same or similar to that of CC components. Although ANSYS Sherlock theory reference doesn't explain CC model theory in detail, a detailed exploration of the provided ANSYS Sherlock CC model theoretical input from its theory reference is carried out in the same chapter and to validate this, manual solder fatigue life calculations were conducted for two resistors, R14 & R15 (part 0402 & 0603 in ANSYS Sherlock Part library), following CC Model theory. The simulation is also done for the same components in ANSYS Sherlock and the simulation results matched the manual calculations precisely. In the same chapter, to develop BTC theory, the same procedure that is followed for CC Model solder fatigue life calculation is followed. An assumption was made that ANSYS Sherlock uses the default LED geometry for BTC model simulations. The LED geometry suggested by Dr. Nathan Blattau, who leads the development team for Sherlock Automated Design Analysis (ADA) CAE App, was used. Notably, BTC components have a thermal pad, unlike CC components. Simulations showed that changing the thermal pad's orientation significantly affected reliability, whereas the orientation of the anode and cathode did not. So, in developing the BTC model theory, the Thermal pad is considered in total stiffness calculation. A small adjustment was made in the foundational stiffness calculation also, considering the bond-to-bond pad distance to be the copper pad length, as it is supposed to be from scientific publications. Although ANSYS Sherlock considered it differently in CC model theory with no explanation at all. To validate this theory is used in ANSYS Sherlock simulation, the only available BTC component (BTC 35) in ANSYS Sherlock Part library was simulated and compared with manual calculations done with the built BTC model theory. Surprisingly, they matched exactly, giving confidence in the use of the developed BTC model theory in ANSYS Sherlock's solder simulation. However, due to the unavailability of alternative parts, further validation couldn't be conducted.

Chapter 4 explains the simulation setup for Samsung V Series LEDs on Aluminum and Copper IMS PCB with SAC305 and INNOLOt solder.

The simulation results for BTC model solder fatigue in ANSYS Sherlock for Samsung V series LEDs are then compared with real test results in Chapter 5, revealing unacceptable discrepancies. Consistently, the simulation yields lower values compared to the real-test results. Specifically, for SAC305 solder on Al IMS PCB, the difference ranges from 67.46% to 73%, while for Innolot solder on Al IMS PCB, the difference ranges from 57.70% to 74.27%. When considering copper PCBs, the difference for SAC305 solder ranges from 48% to 64.32%, and for Innolot solder, it ranges from 23.43% to 49.98%. The reasons behind these differences are explored and concluded in the same chapter as well.

Considering the findings from Chapter 5, a slight modification of the BTC theory is implemented in Chapter 6, and the solder fatigue life of Samsung V Series LEDs for SAC305 solder is manually calculated with this modified theory. All LED solders are considered for total stiffness calculation, and a different constant is used in the formula of total displacement calculation caused by thermal mismatch. Additionally, considering the hysteresis loop to be equilateral, a multiplication constant of 0.5 is used in the total strain energy calculation. The results of manual calculations with the modified theory are compared to real test results in the same chapter, and a satisfactory match is found for the Al IMS PCB manual calculation result and the real test result. The differences between the manually calculated solder fatigue life to real test result range from -7.94% to +1.12% , but an unsatisfactory correlation is observed for Copper IMS PCB where difference between the manually calculated solder fatigue life to real test result range from -37.55% to -19.79%. The Chapter 6 also concludes that The calculation methodology of the elastic modulus and Poisson ratio by ANSYS Sherlock is not disclosed. Internal discussions with Sherlock clarified that their model may not perform well for copper IMS PCBs. Hence, the discrepancies between manually calculated fatigue life and real test results for Cu IMS PCB may be attributed to this factor.

As an additional work, the impact of LED parameter changes—modifying the LED dimensions to make it more rectangular in shape and altering the LED footprint geometry—is assessed in Chapter 7. There is reasonable suspicion regarding whether this simulation was conducted correctly or not, as solder geometry cannot be accurately modeled in Ansys Sherlock for the BTC model. Additionally, there are geometrical changes in the modification concepts. This suspicion is heightened, particularly in the case of the second modification concept in the OSRAM reference LED, where the anode and cathode are positioned in the middle area of the component. The simulation indicates a very low fatigue life, whereas it should theoretically be higher. Nevertheless, individual fatigue life changes are discussed in the conclusion of that specific chapter 7.

However, The work conducted in this thesis provides a clear understanding that **the ANSYS Sherlock tool cannot be used confidently for predicting solder fatigue life of BTC components like LEDs.** This limitation arises from the theoretical approach of the simulation, which differs depending on the footprint geometry of each LED. ANSYS Sherlock considers a default geometrical shape for simulation, and considering the specific geometry for all BTC components solder fatigue simulation, such as LEDs, is not feasible.

Despite the clarification made in this thesis work, there are some points that require further analysis. However, due to time constraints, this was not possible at this time, and it can be considered for **future work**:

- The reference temperature used in the calculation differs. No input was provided on how to fix this reference temperature. Further analysis should be carried out on this point.
- In the solder foundation stiffness calculation, the edge-to-edge distance represented as 'a' is considered half of the width of the CC component for an unknown reason, while it should be the total length of the solder. In BTC theory, taking the total solder length as 'a' is validated by comparing manually calculated fatigue life with simulated fatigue life. Further validation should be done for BTC components, which was not possible this time due to the limited availability of BTC parts (only one working part is available) in the ANSYS Sherlock part library. CC model theory was verified with two CC components - R14 and R15 resistors. All of these were done with SAC305 solder. These should be checked with INNOLOT Solder. Due to time constraints, this was not done in this instance.
- In the manual solder fatigue life calculation of Samsung V series LEDs, the distance from the neutral point to the crack point was considered 40% of the half-diagonal length of the component. A further geometrical analysis should be carried out to fix this point, depending on the component footprint geometry.
- The analysis and disclosure of the theory for calculating the equivalent elastic modulus of PCB (Printed Circuit Board) and the Poisson ratio should be undertaken.

Appendix A

Appendix

A.1 Python Code sample for Data Visualisation

The follwoing python code(and it's modification) is used to make comparison curves and modification of ANSYS Sherlock's output Reliability curves.

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from scipy.stats import exponweib

ref = pd.read_csv("AL_BTC_Osram_Ref_TSF(1)_ALN_CTE(4.5).csv")
def Preprocess(dataframe, n):
    modifi = dataframe.drop(0, axis=0).reset_index(drop=True)

    # Check for zero values before computing cbrt
    zero_mask = (1 - (modifi['Life Prediction'] / 100)) == 0
    modifi = modifi[~zero_mask]

    modifi['lambda_val'] = (modifi['Lifetime (hrs)'])/np.cbrt(-np.log(1 - (modifi['Life Prediction']/100))

    # Check for zero values before computing exponential
    zero_mask = modifi['lambda_val'] == 0
    modifi = modifi[~zero_mask]

    lambda_val = modifi['lambda_val'].iloc[0]
    modifi['Life Prediction'] = (1 - np.exp(-((modifi['Lifetime (hrs)']/ lambda_val) ** n)))*100

    return modifi

ref1 = Preprocess(ref, 3)
ref2 = Preprocess(ref, 5)
x = np.arange(0,2001)
y1 = np.interp(x,ref1["Lifetime (hrs)"],ref1["Life Prediction"])
y2 = np.interp(x,ref2["Lifetime (hrs)"],ref2["Life Prediction"])

y3 = np.full(len(x),5.0)
y3[x == 1000] = 0.0
y3[x>= 1001] = np.nan
```

```
# Create a new figure and axes
fig, ax = plt.subplots(figsize=(20,10))
# Set the background color
ax.set_facecolor('#E5E4E2')  # Change to the desired gray shade
# Plot the three line charts
ax.plot(x, y3, color='black', label='Target',linewidth = 0.75)
ax.plot(x, y1, color='red', label='Weibull Factor 3',linewidth = 1, linestyle='-')
ax.plot(x, y2, color='blue', label='Weibull Factor 5',linewidth = 1, linestyle='-')


# Set labels and title
ax.set_xlabel('Lifetime (hrs)', fontsize = 18)
ax.set_ylabel('Probability of failure (%)', fontsize = 18)
ax.set_title('"BTC" Life Prediction Comparison of OSRAM Ref,
                Concept 1 & 2 for TSF 1 and ALN CTE 4.5', fontsize = 25)

# Set x-axis and y-axis ticks and labels
ax.set_xticks(np.arange(0, 2001, 100))
ax.set_xticks(np.arange(0, 2001, 50), minor = True)
ax.set_yticks(np.arange(0, 101, 5))

ax.tick_params(axis = 'x',labelsize = 17)
ax.tick_params(axis = 'y',labelsize = 14)

# Set x-axis and y-axis limits
ax.set_xlim(left=0)
ax.set_ylim(bottom=0, top=125)  # Adjust the upper limit as desired

# Rotate x-axis tick labels by 45 degrees
ax.set_xticklabels(ax.get_xticks(), rotation=45)

# Add a legend
ax.legend(fontsize = 18)

# Add the lines in the upper left corner
ax.text(0.025, 0.95, 'Service Life = 1000 hrs', transform=ax.transAxes, fontsize = 18)
ax.text(0.025, 0.9, 'Prob. of Failure Goal = 5%', transform=ax.transAxes, fontsize = 18)

# Add grid lines
ax.grid(which = "both" , color='gray', linestyle=':', linewidth=0.5)
ax.set_axisbelow(True)


# Display the plot
plt.show()
```

A.2 Python Code: Manual Calculation of R14 & R15 Resistor's Solder Fatigue (CC Model)

The following two subsections explain the initial setting input for resistors R14 and R15 in the manual solder fatigue life calculation. Please use only one subsection of the two at a time.

A.2.1 R14 Setting Block

```
# IMPORTS
import math
from decimal import Decimal, getcontext

getcontext().prec = 10

W_comp = 0.5*(1/1000)           # in meter
L_comp = 1*(1/1000)             # in meter
T_comp = 0.35*(1/1000)          # in meter

E_comp = 300000                 # Young's Modulus of component material in MPa
CTE_comp = 6.8/1e6              # Coefficient of Thermal Expansion of component material in 1/°C

W_pcb = 144*(1/1000)            # in meter
L_pcb = 144*(1/1000)            # in meter
T_pcb = 1.681 *(1/1000)         # in meter
E_pcb = 4432                    # Young's Modulus of PCB material in MPa
CTE_pcb = 49.021/1e6            # Coefficient of Thermal Expansion of PCB material in 1/°C
V_pcb = 0.18                    # Poisson ratio of PCB material
G_pcb = E_pcb / (2 * (1 + V_pcb))                   # Shear modulus of PCB material

E_copper = 113000               # Young's Modulus of copper pad in MPa
V_copper = 0.34                 # Poisson ratio of copper pad
G_pad = E_copper / (2 * (1 + V_copper))             # Shear Modulus of copper pad

V_solder = 0.36                 # Poisson ratio of SAC305 solder
E_solder_max = 1.072e4          # Elastic modulus of SAC305 Solder at 125°C in MPa
E_solder_min = 4.7990e4         # Elastic modulus of SAC305 Solder at -40°C in MPa
G_soldermax = E_solder_max / (2 * (1 + V_solder))   # Shear Modulus of SAC305 solder at 125°C
G_soldermin = E_solder_min / (2 * (1 + V_solder))   # Shear Modulus of SAC305 solder at -40°C

W_pad = 0.50*(1/1000)           # in meter
L_pad = 0.20*(1/1000)           # in meter
A_pad = L_pad * W_pad           # in m^2
T_pad = 35*(1e-6)               # in meter

W_solder = W_pad                # in meter
L_solder = L_pad                # in meter
A_solder = L_solder * W_solder  # in m^2
H_solder = 20*(1e-6)            # in meter

Temp_max = 125+273.15           # K
Temp_min = -40+273.15           # k
Temp_mean = 30.455 + 273.15     # Reference temperature considered to be 30.455°C
#Temp_mean = 25 + 273.15
del_T_hot = Temp_max - Temp_mean
del_T_cold = Temp_mean - Temp_min

T_dwell_max = 29.8 * 60         # seconds
time_critical = 360 * 60        # seconds
activation_energy = 2185        # in eV
Temp_critical = 378             # K
```

A.2.2 R15 Setting Block

```
# IMPORTS
import math
from decimal import Decimal, getcontext

# Set the global precision for the entire script
getcontext().prec = 20

W_comp = 0.8*(1/1000)           # in meter
L_comp = 1.6*(1/1000)           # in meter
T_comp = 0.9*(1/1000)           # in meter
E_comp = 300000#*(1e6)          # Young's Modulus of component material in MPa
CTE_comp = 6.8/1e6              # in 1/°C
W_pcb = 144*(1/1000)            # in meter
L_pcb = 144*(1/1000)            # in meter
T_pcb = 1.681 *(1/1000)         # in meter
E_pcb = 4432  # in MPa
CTE_pcb = 49.021/1e6            # in 1/°C
V_pcb = 0.18                    # poisson ratio of pcb
G_pcb = E_pcb / (2 * (1 + V_pcb))                   # shear modulus of pcb
E_copper = 113000               # in MPa
V_copper = 0.34                 # poisson ratio of copper pad
G_pad = E_copper / (2 * (1 + V_copper))             # Shear Modulus of copper pad
V_solder = 0.36                 # poisson ratio of SAC305 solder
E_solder_max = 1.072e4          # in MPa            # elastic modulus of SAC305 Solder at 125°C
E_solder_min = 4.799e4          # in MPa            # elastic modulus of SAC305 Solder at -40°C
G_soldermax = E_solder_max / (2 * (1 + V_solder))   # Shear Modulus of SAC305 solder at 125°C
G_soldermin = E_solder_min / (2 * (1 + V_solder))   # Shear Modulus of SAC305 solder at -40°C
W_pad = 0.95*(1/1000)           # in meter
L_pad = 0.88*(1/1000)           # in meter
A_pad = L_pad * W_pad           # in m^2
T_pad = 35*(1e-6)               # in meter
W_solder = W_pad                # in meter
L_solder = L_pad                # in meter
A_solder = A_pad                # in m^2
H_solder = 20*(1e-6)            # in meter
Temp_max = 125+273.15           # K
Temp_min = -40+273.15           # k
Temp_mean = 30.455 + 273.15     # Reference temperature considered to be 30.455°C
del_T_hot = Temp_max - Temp_mean
del_T_cold = Temp_mean - Temp_min
T_dwell_max = 29.8 * 60         # seconds
time_critical = 360 * 60        # seconds
activation_energy = 2185        # in eV
Temp_critical = 378             # K
```

A.2.3 Mean Fatigue Life Calculation:

```
def calculate_stiffness(temperature_type):
    K_pcb = 2 * E_pcb * W_comp * T_pcb / L_comp
    K_foundation = (9 * G_pcb * (W_pad / 2)) / (2 - V_pcb)
    K_copperpad = G_pad * A_pad / T_pad
    K_component = E_comp * W_comp * T_comp / L_comp
```

```
    if temperature_type == 'hot':
        K_solder = G_soldermax * A_solder / H_solder
    elif temperature_type == 'cold':
        K_solder = G_soldermin * A_solder / H_solder
    else:
        raise ValueError("Invalid temperature_type. Use 'hot' or 'cold'.")

    return 1 / ((1/K_foundation) + (1/K_copperpad) + (1/K_solder) + (1/K_component) + (1/K_pcb))

K_hot = calculate_stiffness('hot')
K_cold = calculate_stiffness('cold')



def calculate_ineleastic_strain_energy(temperature_types):
    inelastic_strain_energy = 0
    for temperature_type in temperature_types:
        # Select parameters based on temperature type
        K, G_solder, del_T = (K_hot, G_soldermax, del_T_hot)
                             if temperature_type == 'hot' else (K_cold, G_soldermin, del_T_cold)

        # Common calculations
        U_total_T = abs(CTE_comp - CTE_pcb) * del_T * (L_comp / 2)
        elastic_strain = ((K * U_total_T) / A_solder) / G_solder
        plastic_strain = ((U_total_T) / H_solder) - elastic_strain
        max_strain_range = plastic_strain * ((T_dwell_max / time_critical) ** 0.136) *
                            math.exp(activation_energy *
                            ((1 / Temp_critical) - (1 / Temp_max if temperature_type == 'hot'
                            else 1 / Temp_min)))

        inelastic_strain_energy += max_strain_range * ((K * U_total_T) / A_solder)

    return inelastic_strain_energy

ineleastic_strain_energy = calculate_ineleastic_strain_energy(['hot', 'cold'])

SE_coeff = 0.001

C3 = 1

N_f1 = round(1 / ( SE_coeff * C3 * ineleastic_strain_energy))
print("Mean Fatigue Life: "+str(N_f1))
```

A.3 Python Code: Manual LED BTC 35 Solder Fatigue (BTC Model)

A.3.1 BTC 35 Setting Block

```
# IMPORTS
import math
from decimal import Decimal, getcontext

# Set the global precision for the entire script
getcontext().prec = 20
```

```
W_comp = 3.5*(1/1000)             # in meter
L_comp = 3.5*(1/1000)           # in meter
L_d = (math.sqrt(((W_comp)**2)+((L_comp)**2)))
T_comp = 0.77*(1/1000)           # in meter
E_comp = 300000                 # Young's Modulus of component material in MPa
CTE_comp = 6.8/1e6              # in 1/°C
W_pcb = 144*(1/1000)            # in meter
L_pcb = 144*(1/1000)            # in meter
T_pcb = 1.681 *(1/1000)         # in meter
E_pcb = 4432                    # in MPa
CTE_pcb = 49.021/1e6            # in 1/°C
V_pcb = 0.18                    # poisson ratio of pcb
G_pcb = E_pcb / (2 * (1 + V_pcb))                   # shear modulus of pcb
E_copper = 113000               # in MPa
V_copper = 0.34                 # poisson ratio of copper pad
G_pad = E_copper / (2 * (1 + V_copper))             # Shear Modulus of copper pad
V_solder = 0.36                 # poisson ratio of SAC305 solder
E_solder_max = 1.072e4          # in MPa            # elastic modulus of SAC305 Solder at 125°C
E_solder_min = 4.799e4          # in MPa            # elastic modulus of SAC305 Solder at -40°C
G_soldermax = E_solder_max / (2 * (1 + V_solder))   # Shear Modulus of SAC305 solder at 125°C
G_soldermin = E_solder_min / (2 * (1 + V_solder))   # Shear Modulus of SAC305 solder at -40°C
W_pad = 3.25*(1/1000)           # in meter
L_pad = 0.48*(1/1000)           # in meter
A_pad = L_pad * W_pad           # in meter^2
T_pad = 35*(1e-6)               # in meter
W_solder = W_pad                # in meter
L_solder = L_pad                # in meter
A_solder = A_pad                # in meter^2
H_solder = 60*(1e-6)            # in meter
Temp_max = 125+273.15           # K
Temp_min = -40+273.15           # k

Temp_mean = 27 + 273.15     # Reference temperature considered to be 27°C
del_T_hot = Temp_max - Temp_mean
del_T_cold = Temp_mean - Temp_min
T_dwell_max = 29.8 * 60         # seconds
time_critical = 360 * 60        # seconds
activation_energy = 2185        # in eV
Temp_critical = 378             # K



W_thermalpad = 3.25 * (1/1000)                # in meter
L_thermalpad = 1.3 * (1/1000)               # in meter
A_thermalpad = L_thermalpad * W_thermalpad
T_thermalpad = 35 * (1e-6)                    # in meter

W_thermalpad_solder = W_thermalpad          # in meter
L_thermalpad_solder = L_thermalpad          # in meter
A_thermalpad_solder = W_thermalpad_solder * L_thermalpad_solder  # in meter^2
```

A.3.2 Mean Fatigue Life Calculation:

```
def calculate_stiffness(temperature_type):
    K_pcb = 2*E_pcb * W_comp * T_pcb / L_comp
    K_foundation = (9 * G_pcb * (L_pad)) / (2 - V_pcb)
    K_foundation_thermal = (9 * G_pcb * (L_thermalpad)) / (2 - V_pcb)
    K_copperpad = G_pad * A_pad / T_pad
    K_thermalpad = G_pad * A_thermalpad / T_pad
    K_component = E_comp * W_comp * T_comp / L_comp
    if temperature_type == 'hot':
        K_solder = G_soldermax * A_solder / H_solder
        K_solder_thermal = G_soldermax * A_thermalpad_solder / H_solder
    elif temperature_type == 'cold':
        K_solder = G_soldermin * A_solder / H_solder
        K_solder_thermal = G_soldermin *A_thermalpad_solder / H_solder
    else:
        raise ValueError("Invalid temperature_type. Use 'hot' or 'cold'.")
    a = 1/((1/K_foundation)+(1/K_copperpad)+(1/K_solder))
    b = a
    c = 1/((1/K_foundation_thermal)+(1/K_thermalpad)+(1/K_solder_thermal))
    d = a+c#+b
    e = 1/((1/d)+(1/K_component)+(1/K_pcb))
    return e


K_hot = calculate_stiffness('hot')
K_cold = calculate_stiffness('cold')


def calculate_ineleastic_strain_energy(temperature_types):
    inelastic_strain_energy = 0
    for temperature_type in temperature_types:
        # Select parameters based on temperature type
        K, G_solder, del_T = (K_hot, G_soldermax, del_T_hot) if temperature_type == 'hot'
                             else (K_cold, G_soldermin, del_T_cold)

        # Common calculations
        U_total_T = abs(CTE_comp - CTE_pcb) * del_T *(L_comp/2)*(1/math.sqrt(2))
        elastic_strain = ((K * U_total_T) / A_solder) / G_solder
        plastic_strain = ((U_total_T) / H_solder) - elastic_strain
        max_strain_range = plastic_strain * ((T_dwell_max / time_critical) ** 0.136) *
                           math.exp(activation_energy *
                           ((1 / Temp_critical) - (1 / Temp_max if temperature_type == 'hot'
                           else 1 / Temp_min)))

        inelastic_strain_energy += max_strain_range * ((K * U_total_T) / A_solder)

    return inelastic_strain_energy

ineleastic_strain_energy = calculate_ineleastic_strain_energy(['hot', 'cold'])

SE_coeff = 0.001

C3 = 1
```

```
N_f1 = round(1 / ( SE_coeff * C3 * ineleastic_strain_energy))
print("Mean Fatigue Life: "+str(N_f1))
```

A.4 Python Code: Manual Solder Fatigue Life Calculations for Samsung V Series LEDs (BTC Model)

The following six subsections refers to different settings of LEDs. Please use only one code block of these six at a time, depending on your requirements.

A.4.1 Samsung V 1 Chip LED Setting Block for Aluminum IMS

```
# IMPORTS
import math
from decimal import Decimal, getcontext

# Set the global precision for the entire script
getcontext().prec = 20

W_comp = 2*(1/1000)             # in meter
L_comp = 1.6*(1/1000)           # in meter
L_d = (math.sqrt(((W_comp)**2)+((L_comp)**2)))
T_comp = 0.4*(1/1000)           # in meter
E_comp = 300000                 # Young's Modulus of component material in MPa
CTE_comp = 4.5/1e6              # in 1/°C
W_pcb = 144*(1/1000)            # in meter
L_pcb = 144*(1/1000)            # in meter
T_pcb = 1.681 *(1/1000)         # in meter
E_pcb = 4432                    # in MPa
CTE_pcb = 49.021/1e6            # in 1/°C
V_pcb = 0.18                    # poisson ratio of pcb
G_pcb = E_pcb / (2 * (1 + V_pcb))                   # shear modulus of pcb
E_copper = 113000               # in MPa
V_copper = 0.34                 # poisson ratio of copper pad
G_pad = E_copper / (2 * (1 + V_copper))             # Shear Modulus of copper pad
V_solder = 0.36                 # poisson ratio of SAC305 solder
E_solder_max = 1.072e4          # in MPa            # elastic modulus of SAC305 Solder at 125°C
E_solder_min = 4.799e4          # in MPa            # elastic modulus of SAC305 Solder at -40°C
G_soldermax = E_solder_max / (2 * (1 + V_solder))   # Shear Modulus of SAC305 solder at 125°C
G_soldermin = E_solder_min / (2 * (1 + V_solder))   # Shear Modulus of SAC305 solder at -40°C
W_pad = 0.87*(1/1000)           # in meter
L_pad = 0.59*(1/1000)           # in meter
A_pad = L_pad * W_pad           # in meter^2
T_pad = 35*(1e-6)               # in meter
W_solder = W_pad                # in meter
L_solder = L_pad                # in meter
A_solder = A_pad                # in meter^2
H_solder = 60*(1e-6)            # in meter
Temp_max = 125+273.15           # K
Temp_min = -40+273.15           # k

Temp_mean = 30.455 + 273.15     # Reference temperature considered to be 30.455°C
```

```
del_T_hot = Temp_max - Temp_mean
del_T_cold = Temp_mean - Temp_min
T_dwell_max = 29.8 * 60          # seconds
time_critical = 360 * 60         # seconds
activation_energy = 2185         # in eV
Temp_critical = 378              # K




W_thermalpad = 0.72 * (1/1000)                # in meter
L_thermalpad = 1.44 * (1/1000)                # in meter
A_thermalpad = L_thermalpad * W_thermalpad
T_thermalpad = 35 * (1e-6)                    # in meter

W_thermalpad_solder = W_thermalpad          # in meter
L_thermalpad_solder = L_thermalpad          # in meter
A_thermalpad_solder = W_thermalpad_solder * L_thermalpad_solder  # in meter^2
```

A.4.2 Samsung V 1 Chip LED Setting Block for Copper IMS

```
# IMPORTS
import math
from decimal import Decimal, getcontext

# Set the global precision for the entire script
getcontext().prec = 20

W_comp = 2*(1/1000)              # in meter
L_comp = 1.6*(1/1000)            # in meter
L_d = (math.sqrt(((W_comp)**2)+((L_comp)**2)))
T_comp = 0.4*(1/1000)            # in meter
E_comp = 300000                  # Young's Modulus of component material in MPa
CTE_comp = 4.5/1e6               # in 1/°C
W_pcb = 144*(1/1000)             # in meter
L_pcb = 144*(1/1000)             # in meter
T_pcb = 1.681 *(1/1000)          # in meter
E_pcb = 6379                     # in MPa
CTE_pcb = 47.986/1e6             # in 1/°C
V_pcb = 0.18                     # poisson ratio of pcb
G_pcb = E_pcb / (2 * (1 + V_pcb))                    # shear modulus of pcb
E_copper = 113000                # in MPa
V_copper = 0.34                  # poisson ratio of copper pad
G_pad = E_copper / (2 * (1 + V_copper))              # Shear Modulus of copper pad
V_solder = 0.36                  # poisson ratio of SAC305 solder
E_solder_max = 1.072e4           # in MPa            # elastic modulus of SAC305 Solder at 125°C
E_solder_min = 4.799e4           # in MPa            # elastic modulus of SAC305 Solder at -40°C
G_soldermax = E_solder_max / (2 * (1 + V_solder))    # Shear Modulus of SAC305 solder at 125°C
G_soldermin = E_solder_min / (2 * (1 + V_solder))    # Shear Modulus of SAC305 solder at -40°C
W_pad = 0.87*(1/1000)            # in meter
L_pad = 0.59*(1/1000)            # in meter
A_pad = L_pad * W_pad            # in meter^2
T_pad = 35*(1e-6)                # in meter
```

```
W_solder = W_pad                # in meter
L_solder = L_pad                # in meter
A_solder = A_pad                # in meter^2
H_solder = 60*(1e-6)            # in meter
Temp_max = 125+273.15           # K
Temp_min = -40+273.15           # k
Temp_mean = 30.445 + 273.15     # Reference temperature considered to be 30.455°C
del_T_hot = Temp_max - Temp_mean
del_T_cold = Temp_mean - Temp_min
T_dwell_max = 29.8 * 60         # seconds
time_critical = 360 * 60        # seconds
activation_energy = 2185        # in eV
Temp_critical = 378             # K



W_thermalpad = 0.72 * (1/1000)                 # in meter
L_thermalpad = 1.44 * (1/1000)                 # in meter
A_thermalpad = L_thermalpad * W_thermalpad
T_thermalpad = 35 * (1e-6)                     # in meter

W_thermalpad_solder = W_thermalpad           # in meter
L_thermalpad_solder = L_thermalpad           # in meter
A_thermalpad_solder = W_thermalpad_solder * L_thermalpad_solder  # in meter^2
```

A.4.3 Samsung V 2 Chip LED Setting Block for Aluminum IMS

```
# IMPORTS
import math
from decimal import Decimal, getcontext

# Set the global precision for the entire script
getcontext().prec = 20

W_comp = 3.1*(1/1000)             # in meter
L_comp = 2.9*(1/1000)           # in meter
L_d = (math.sqrt(((W_comp)**2)+((L_comp)**2)))
T_comp = 0.4*(1/1000)           # in meter
E_comp = 300000                 # Young's Modulus of component material in MPa
CTE_comp = 4.5/1e6              # in 1/°C
W_pcb = 144*(1/1000)            # in meter
L_pcb = 144*(1/1000)            # in meter
T_pcb = 1.681 *(1/1000)         # in meter
E_pcb = 4432                    # in MPa
CTE_pcb = 49.021/1e6            # in 1/°C
V_pcb = 0.18                    # poisson ratio of pcb
G_pcb = E_pcb / (2 * (1 + V_pcb))                   # shear modulus of pcb
E_copper = 113000               # in MPa
V_copper = 0.34                 # poisson ratio of copper pad
G_pad = E_copper / (2 * (1 + V_copper))             # Shear Modulus of copper pad
V_solder = 0.36                 # poisson ratio of SAC305 solder
E_solder_max = 1.072e4          # in MPa            # elastic modulus of SAC305 Solder at 125°C
```

```
E_solder_min = 4.799e4          # in MPa          # elastic modulus of SAC305 Solder at -40°C
G_soldermax = E_solder_max / (2 * (1 + V_solder))  # Shear Modulus of SAC305 solder at 125°C
G_soldermin = E_solder_min / (2 * (1 + V_solder))  # Shear Modulus of SAC305 solder at -40°C
W_pad = 1.22*(1/1000)           # in meter
L_pad = 1.32*(1/1000)           # in meter
A_pad = L_pad * W_pad           # in meter^2
T_pad = 35*(1e-6)               # in meter
W_solder = W_pad                # in meter
L_solder = L_pad                # in meter
A_solder = A_pad                # in meter^2
H_solder = 60*(1e-6)            # in meter
Temp_max = 125+273.15           # K
Temp_min = -40+273.15           # k
Temp_mean = 30.455 + 273.15     # Reference temperature considered to be 30.455°C
del_T_hot = Temp_max - Temp_mean
del_T_cold = Temp_mean - Temp_min
T_dwell_max = 29.8 * 60         # seconds
time_critical = 360 * 60        # seconds
activation_energy = 2185        # in eV
Temp_critical = 378             # K



W_thermalpad = 1.32 * (1/1000)                 # in meter
L_thermalpad = 2.74 * (1/1000)                 # in meter
A_thermalpad = L_thermalpad * W_thermalpad
T_thermalpad = 35 * (1e-6)                     # in meter

W_thermalpad_solder = W_thermalpad           # in meter
L_thermalpad_solder = L_thermalpad           # in meter
A_thermalpad_solder = W_thermalpad_solder * L_thermalpad_solder  # in meter^2
```

A.4.4 Samsung V 2 Chip LED Setting Block for Copper IMS

```
# IMPORTS
import math
from decimal import Decimal, getcontext

# Set the global precision for the entire script
getcontext().prec = 20

W_comp = 3.1*(1/1000)             # in meter
L_comp = 2.9*(1/1000)           # in meter
L_d = (math.sqrt(((W_comp)**2)+((L_comp)**2)))
T_comp = 0.4*(1/1000)           # in meter
E_comp = 300000                 # Young's Modulus of component material in MPa
CTE_comp = 4.5/1e6              # in 1/°C
W_pcb = 144*(1/1000)            # in meter
L_pcb = 144*(1/1000)            # in meter
T_pcb = 1.681 *(1/1000)         # in meter
E_pcb = 6379                    # in MPa
CTE_pcb = 47.986/1e6            # in 1/°C
```

```
V_pcb = 0.18                    # poisson ratio of pcb
G_pcb = E_pcb / (2 * (1 + V_pcb))                    # shear modulus of pcb
E_copper = 113000               # in MPa
V_copper = 0.34                 # poisson ratio of copper pad
G_pad = E_copper / (2 * (1 + V_copper))              # Shear Modulus of copper pad
V_solder = 0.36                 # poisson ratio of SAC305 solder
E_solder_max = 1.072e4          # in MPa            # elastic modulus of SAC305 Solder at 125°C
E_solder_min = 4.799e4          # in MPa            # elastic modulus of SAC305 Solder at -40°C
G_soldermax = E_solder_max / (2 * (1 + V_solder))   # Shear Modulus of SAC305 solder at 125°C
G_soldermin = E_solder_min / (2 * (1 + V_solder))   # Shear Modulus of SAC305 solder at -40°C
W_pad = 1.22*(1/1000)           # in meter
L_pad = 1.32*(1/1000)           # in meter
A_pad = L_pad * W_pad           # in meter^2
T_pad = 35*(1e-6)               # in meter
W_solder = W_pad                # in meter
L_solder = L_pad                # in meter
A_solder = A_pad                # in meter^2
H_solder = 60*(1e-6)            # in meter
Temp_max = 125+273.15           # K
Temp_min = -40+273.15           # k
Temp_mean = 30.455 + 273.15     # Reference temperature considered to be 30.455°C
del_T_hot = Temp_max - Temp_mean
del_T_cold = Temp_mean - Temp_min
T_dwell_max = 29.8 * 60         # seconds
time_critical = 360 * 60        # seconds
activation_energy = 2185        # in eV
Temp_critical = 378             # K



W_thermalpad = 1.32 * (1/1000)                 # in meter
L_thermalpad = 2.74 * (1/1000)                 # in meter
A_thermalpad = L_thermalpad * W_thermalpad
T_thermalpad = 35 * (1e-6)                     # in meter

W_thermalpad_solder = W_thermalpad           # in meter
L_thermalpad_solder = L_thermalpad           # in meter
A_thermalpad_solder = W_thermalpad_solder * L_thermalpad_solder  # in meter^2
```

A.4.5 Samsung V 3 Chip LED Setting Block for Aluminum IMS

```
# IMPORTS
import math
from decimal import Decimal, getcontext

# Set the global precision for the entire script
getcontext().prec = 20

W_comp = 3.1*(1/1000)             # in meter
L_comp = 4.13*(1/1000)           # in meter
L_d = (math.sqrt(((W_comp)**2)+((L_comp)**2)))
T_comp = 0.4*(1/1000)           # in meter
```

```
E_comp = 300000                # Young's Modulus of component material in MPa
CTE_comp = 4.5/1e6             # in 1/°C
W_pcb = 144*(1/1000)           # in meter
L_pcb = 144*(1/1000)           # in meter
T_pcb = 1.681 *(1/1000)        # in meter
E_pcb = 4432                   # in MPa
CTE_pcb = 49.021/1e6           # in 1/°C
V_pcb = 0.18                   # poisson ratio of pcb
G_pcb = E_pcb / (2 * (1 + V_pcb))                   # shear modulus of pcb
E_copper = 113000              # in MPa
V_copper = 0.34                # poisson ratio of copper pad
G_pad = E_copper / (2 * (1 + V_copper))             # Shear Modulus of copper pad
V_solder = 0.36                # poisson ratio of SAC305 solder
E_solder_max = 1.072e4         # in MPa             # elastic modulus of SAC305 Solder at 125°C
E_solder_min = 4.799e4         # in MPa             # elastic modulus of SAC305 Solder at -40°C
G_soldermax = E_solder_max / (2 * (1 + V_solder))   # Shear Modulus of SAC305 solder at 125°C
G_soldermin = E_solder_min / (2 * (1 + V_solder))   # Shear Modulus of SAC305 solder at -40°C
W_pad = 1.32*(1/1000)          # in meter
L_pad = 1.84*(1/1000)          # in meter
A_pad = L_pad * W_pad          # in meter^2
T_pad = 35*(1e-6)              # in meter
W_solder = W_pad               # in meter
L_solder = L_pad               # in meter
A_solder = A_pad               # in meter^2
H_solder = 60*(1e-6)           # in meter
Temp_max = 125+273.15          # K
Temp_min = -40+273.15          # k
Temp_mean = 30.455 + 273.15    # Reference temperature considered to be 30.455°C
del_T_hot = Temp_max - Temp_mean
del_T_cold = Temp_mean - Temp_min
T_dwell_max = 29.8 * 60        # seconds
time_critical = 360 * 60       # seconds
activation_energy = 2185       # in eV
Temp_critical = 378            # K




W_thermalpad = 0.72 * (1/1000)                # in meter
L_thermalpad = 1.44 * (1/1000)                # in meter
A_thermalpad = L_thermalpad * W_thermalpad
T_thermalpad = 35 * (1e-6)                    # in meter

W_thermalpad_solder = W_thermalpad          # in meter
L_thermalpad_solder = L_thermalpad          # in meter
A_thermalpad_solder = W_thermalpad_solder * L_thermalpad_solder  # in meter^2
```

A.4.6 Samsung V 3 Chip LED Setting Block for Copper IMS

```
# IMPORTS
import math
from decimal import Decimal, getcontext
```

```
# Set the global precision for the entire script
getcontext().prec = 20

W_comp = 3.1*(1/1000)             # in meter
L_comp = 4.13*(1/1000)           # in meter
L_d = (math.sqrt(((W_comp)**2)+((L_comp)**2)))
T_comp = 0.4*(1/1000)           # in meter
E_comp = 300000                 # Young's Modulus of component material in MPa
CTE_comp = 4.5/1e6              # in 1/°C
W_pcb = 144*(1/1000)            # in meter
L_pcb = 144*(1/1000)            # in meter
T_pcb = 1.681 *(1/1000)         # in meter
E_pcb = 6379                    # in MPa
CTE_pcb = 47.986/1e6            # in 1/°C
V_pcb = 0.18                    # poisson ratio of pcb
G_pcb = E_pcb / (2 * (1 + V_pcb))                   # shear modulus of pcb
E_copper = 113000               # in MPa
V_copper = 0.34                 # poisson ratio of copper pad
G_pad = E_copper / (2 * (1 + V_copper))             # Shear Modulus of copper pad
V_solder = 0.36                 # poisson ratio of SAC305 solder
E_solder_max = 1.072e4          # in MPa            # elastic modulus of SAC305 Solder at 125°C
E_solder_min = 4.799e4          # in MPa            # elastic modulus of SAC305 Solder at -40°C
G_soldermax = E_solder_max / (2 * (1 + V_solder))   # Shear Modulus of SAC305 solder at 125°C
G_soldermin = E_solder_min / (2 * (1 + V_solder))   # Shear Modulus of SAC305 solder at -40°C
W_pad = 1.32*(1/1000)           # in meter
L_pad = 1.84*(1/1000)           # in meter
A_pad = L_pad * W_pad           # in meter^2
T_pad = 35*(1e-6)               # in meter
W_solder = W_pad                # in meter
L_solder = L_pad                # in meter
A_solder = A_pad                # in meter^2
H_solder = 60*(1e-6)            # in meter
Temp_max = 125+273.15           # K
Temp_min = -40+273.15           # k
Temp_mean = 30.455 + 273.15     # Reference temperature considered to be 30.455°C
del_T_hot = Temp_max - Temp_mean
del_T_cold = Temp_mean - Temp_min
T_dwell_max = 29.8 * 60         # seconds
time_critical = 360 * 60        # seconds
activation_energy = 2185        # in eV
Temp_critical = 378             # K



W_thermalpad = 0.72 * (1/1000)                 # in meter
L_thermalpad = 1.44 * (1/1000)                 # in meter
A_thermalpad = L_thermalpad * W_thermalpad
T_thermalpad = 35 * (1e-6)                     # in meter

W_thermalpad_solder = W_thermalpad           # in meter
L_thermalpad_solder = L_thermalpad           # in meter
A_thermalpad_solder = W_thermalpad_solder * L_thermalpad_solder  # in meter^2
```

A.4.7 Calculation of Mean Fatigue Life N_f

A.4.7.1 Stiffness Calculation (in the x-direction)

```
def calculate_stiffness(temperature_type):
    K_pcb = E_pcb * W_pcb * T_pcb / L_pcb
    K_foundation = (9 * G_pcb * (L_pad)) / (2 - V_pcb)
    K_foundation_thermal = (9 * G_pcb * (L_thermalpad)) / (2 - V_pcb)
    K_copperpad = G_pad * A_pad / T_pad
    K_thermalpad = G_pad * A_thermalpad / T_pad
    K_component = E_comp * W_comp * T_comp / L_comp
    if temperature_type == 'hot':
        K_solder = G_soldermax * A_solder / H_solder
        K_solder_thermal = G_soldermax * A_thermalpad_solder / H_solder
    elif temperature_type == 'cold':
        K_solder = G_soldermin * A_solder / H_solder
        K_solder_thermal = G_soldermin *A_thermalpad_solder / H_solder
    else:
        raise ValueError("Invalid temperature_type. Use 'hot' or 'cold'.")
    a = 1/((1/K_foundation)+(1/K_copperpad)+(1/K_solder))
    b = a
    c = 1/((1/K_foundation_thermal)+(1/K_thermalpad)+(1/K_solder_thermal))
    d = a+b+c
    return 1/((1/d)+(1/K_component)+(1/K_pcb))


K_hot = calculate_stiffness('hot')
K_cold = calculate_stiffness('cold')
```

A.4.7.2 Stiffness Calculation (in the y-direction)

```
def calculate_stiffness(temperature_type):
    K_pcb = E_pcb * L_pcb * T_pcb / W_pcb
    K_foundation = (9 * G_pcb * (W_pad)) / (2 - V_pcb)
    K_foundation_thermal = (9 * G_pcb * (W_thermalpad)) / (2 - V_pcb)
    K_copperpad = G_pad * A_pad / T_pad
    K_thermalpad = G_pad * A_thermalpad / T_pad
    K_component = E_comp * L_comp * T_comp / W_comp
    if temperature_type == 'hot':
        K_solder = G_soldermax * A_solder / H_solder
        K_solder_thermal = G_soldermax * A_thermalpad_solder / H_solder
    elif temperature_type == 'cold':
        K_solder = G_soldermin * A_solder / H_solder
        K_solder_thermal = G_soldermin * A_thermalpad_solder / H_solder
    else:
        raise ValueError("Invalid temperature_type. Use 'hot' or 'cold'.")
    a = 1/((1/K_foundation)+(1/K_copperpad)+(1/K_solder))
    b = a
    c = 1/((1/K_foundation_thermal)+(1/K_thermalpad)+(1/K_solder_thermal))
    d = a+b+c
    return 1/((1/d)+(1/K_component)+(1/K_pcb))
```

```
K_hot = calculate_stiffness('hot')
K_cold = calculate_stiffness('cold')
```

A.4.7.3 Mean Fatigue Life N_f Calculation

The mean solder fatigue life is determined by separately evaluating the total stiffness in the x and y directions. The final mean fatigue life is then determined by selecting the lower value between the calculated fatigue lives in each direction.

```
def calculate_ineleastic_strain_energy(temperature_types):
    inelastic_strain_energy = 0
    for temperature_type in temperature_types:
        # Select parameters based on temperature type
        K, G_solder, del_T = (K_hot, G_soldermax, del_T_hot) if temperature_type == 'hot'
        else (K_cold, G_soldermin, del_T_cold)

        # Common calculations
        U_total_T = abs(CTE_comp - CTE_pcb) * del_T *(L_d/2)*0.4
        elastic_strain = ((K * U_total_T) / A_solder) / G_solder
        plastic_strain = ((U_total_T) / H_solder) - elastic_strain
        max_strain_range = plastic_strain * ((T_dwell_max / time_critical) ** 0.136) * math.exp(
            activation_energy * ((1 / Temp_critical) - (1 / Temp_max if temperature_type == 'hot'
            else 1 / Temp_min)))

        inelastic_strain_energy += max_strain_range * ((K * U_total_T) / A_solder)

    return inelastic_strain_energy

ineleastic_strain_energy = 0.5*calculate_ineleastic_strain_energy(['hot', 'cold'])

SE_coeff = 0.001

C3 = 1

N_f1 = round(1 / ( SE_coeff * C3 * ineleastic_strain_energy))
print("Mean TTF : " + str(N_f1))
```

Bibliography

[1] Ansys Sherlock. *Solder Fatigue: Causes & Prevention. url: https://www.ansys.com/de-de/blog/solder-fatigue-causes-prevention*. Accessed on March 16, 2024.

[2] *Ansys Theory Reference provided by ANSYS Sherlock authority*. Accessed from November 1, 2023, to March 16, 2024.

[3] Y. H. Ao et al. "Experimental and Modeling Study of Thermal Cyclic Behavior of Sn-Cu and Sn-Pb Solder Joints." In: *Materials Research Society Symposium Proceedings* 323 (1994), pp. 153–158.

[4] Márcia Baptista et al. "Improved Time-Based Maintenance in Aeronautics with Regressive Support Vector Machines." In: Sept. 2016.

[5] Nathan Blattau and Craig Hillman. "An Engelmaier Model for Leadless Ceramic Chip Devices with Pb-free Solder." In: (2006).

[6] R. Darveaux and K. Banerji. "Constitutive relations for tin-based solder joints." In: *IEEE Transactions on Components, Hybrids, and Manufacturing Technology* 15.6 (1992), pp. 1013–1024. DOI: 10.1109/33.206925.

[7] R. Darveaux et al. "Reliability of Plastic Ball Grid Array Assemblies." In: *Ball Grid Array Technology*. Ed. by J. H. Lau. McGraw-Hill, 1995. Chap. 13, pp. 379–442.

[8] W. Engelmaier. "The Use Environments of Electronic Assemblies and Their Impact of Surface Mount Solder Attachment Reliability." In: *IEEE Transactions of Components, Hybrids, and Manufacturing Technology* 13 (1990), pp. 903–908.

[9] Jiajie Fan et al. "Fatigue Damage Assessment of LED Chip Scale Packages with Finite Element Simulation." In: *2018 19th International Conference on Electronic Packaging Technology (ICEPT)*. 2018, pp. 1642–1648. DOI: 10.1109/ICEPT.2018.8480748.

[10] H. J. Frost and M. F. Ashby. *Deformation Mechanism Maps*. New York: Pergamon, 1982. Chap. 2.

[11] M.I. Gorbunov-Posadov and R.V. Serebrjanyi. "Design of Structures on Elastic Foundation." In: *Proc. 5th Int. Conf. Soil Mech. Found. Eng.* Vol. I. 1961, pp. 643–648.

[12] Arthur J. Hallinan. "A Review of the Weibull Distribution." In: *Journal of Quality Technology* 25.2 (1993), pp. 85–93. DOI: 10.1080/00224065.1993.11979431. eprint: https://doi.org/10.1080/00224065.1993.11979431. URL: https://doi.org/10.1080/00224065.1993.11979431.

[13] Rudi Hechfellner et al. "A Solder Joint Reliability Model for the Philips Lumileds LUXEON Rebel LED Carrier Using Physics of Failure Methodology." In: *International Symposium on Microelectronics* 2013 (Jan. 2013), pp. 000887–000892. DOI: 10.4071/isom-2013-THP44.

[14] Craig Hillman, Nathan Blattau, and M. Lacy. "Predicting fatigue of solder joints subjected to high number of power cycles." In: *IPC APEX EXPO 2014* (2014).

[15] F. Hou et al. "Thermo-mechanical Reliability Study for 3D Package Module Based on Flexible Substrate." In: *International Conference on Electronic Packaging Technology*. 2014, pp. 1296–1300.

[16] *Image from Hella GmbH & Co. KGaA internal database/Documents*.

[17] IPC (Association Connecting Electronics Industries). *Design and Assembly Process Implementation for Bottom Termination SMT Components*. Tech. rep. IPC, Mar. 2011.

[18] Y. Kariya and W. J. Plumbridge. "Mechanical Properties of Sn-3.0%Ag-0.5%Cu Alloy." In: *Proceedings, 7th Symposium on Micro-joining and Assembly Technology in Electronics*. Yokohama, Japan, 2001, pp. 383–388.

[19] John Lau, Walter Dauksher, and Paul Vianco. "Acceleration models, constitutive equations, and reliability of lead-free solders and joints." In: Feb. 2003, pp. 229–236. ISBN: 0-7803-7791-5. DOI: 10.1109/ECTC.2003.1216281.

[20] Wendy Luiten. "Solder joint lifetime of rapid cycled LED components." In: Sept. 2013, pp. 98–103. ISBN: 978-1-4799-2271-0. DOI: 10.1109/THERMINIC.2013.6675232.

[21] M. A. Meyers and K. K. Chawla. *Mechanical Metallurgy: Principles and Applications*. Englewood Cliffs, NJ: Prentice Hall, 1984.

[22] National Institute of Standards and Technology. *Metallurgy Division - Solder Database*. http://www.metallurgy.nist.gov/solder. Accessed on: March 16, 2024.

[23] National Institute of Standards and Technology. *NIST Website*. http://www.nist.gov/. Accessed on: March 16, 2024.

[24] R. W. Neu, D. T. Scott, and M. W. Woodmansee. "Thermomechanical Behavior of 96Sn-4Ag and Castin Alloy." In: *ASME Transactions, Journal of Electronic Packaging* 123.3 (Sept. 2001), pp. 238–246.

[25] K. C. Norris and A. H. Landzberg. "Reliability of Controlled Collapse Interconnections." In: *IBM Journal of Research and Development* 13.3 (May 1969), pp. 266–271.

[26] S. Nurmi et al. "The effect of solder paste composition on the reliability of SnAgCu joints." In: *Microelectronics Reliability* 44 (Mar. 2004), pp. 485–494. DOI: 10.1016/j.microrel.2003.08.004.

[27] M. Osterman. "Strain range approximation for estimating fatigue life of lead-free solder interconnects under thermal cycle loading." In: (2006).

[28] Michael Osterman and Abhijit Dasgupta. "Life expectancies of Pb-free SAC solder interconnects in electronic hardware." In: *Journal of Materials Science: Materials in Electronics* 18 (Mar. 2007), pp. 229–236. DOI: 10.1007/s10854-006-9017-3.

[29] N. Pan et al. "An Acceleration Model for Sn-Ag-Cu Solder Joint Reliability under Various Thermal Cycle Conditions." In: *Proc. SMTA*. 2005, pp. 876–883.

[30] John Pang and B.S. Xiong. "Mechanical Properties for 95.5Sn–3.8Ag–0.7Cu Lead-Free Solder Alloy." In: *Components and Packaging Technologies, IEEE Transactions on* 28 (Jan. 2006), pp. 830–840. DOI: 10.1109/TCAPT.2005.848587.

[31] O. Salmela. "Acceleration factors for lead-free materials." In: (2007).

[32] *Samsung LED News*. URL: https://www.led.samsung.com/about-us/news-events/news/news-detail-32/ (visited on 03/15/2024).

[33] A. Schubertt et al. "Fatigue life models for SnAgCu and SnPb solder joints evaluated by experiments and simulation." In: Feb. 2003, pp. 603–610. ISBN: 0-7803-7791-5. DOI: 10.1109/ECTC.2003.1216343.

[34] Maxim Serebreni et al. "FATIGUE LIFE PREDICTION MODEL FOR LEDS ON METAL CORE PRINTED CIRCUIT BOARDS (MCPCBS) WITH PB-FREE SOLDER ALLOYS." In: Sept. 2017.

[35] M. Shaygi et al. "Finite element-based lifetime modelling of SAC solder joints in LED applications." In: *2021 22nd International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems (EuroSimE)*. 2021, pp. 1–11. DOI: 10.1109/EuroSimE52062.2021.9410847.

[36] H. Song, J. Morris, and Fay Hua. "The creep properties of lead-free solder joints." In: *JOM: the journal of the Minerals, Metals & Materials Society* 54 (June 2002), pp. 30–32. DOI: 10.1007/BF02701846.

[37] M. Spraul et al. "Reliability of SnPb and Pb-free flip-chips under different test conditions." In: *5th International Conference on Thermal and Mechanical Simulation and Experiments in Microelectronics and Microsystems, 2004. EuroSimE 2004. Proceedings of the*. 2004, pp. 437–442. DOI: 10.1109/ESIME.2004.1304075.

[38] Ahmer Syed. "Predicting solder joint reliability for thermal, power, & bend cycle within 25% accuracy." In: Feb. 2001, pp. 255–263. ISBN: 0-7803-7038-4. DOI: 10.1109/ECTC.2001.927732.

[39] Ahmer Syed. "Accumulated creep strain and energy density based thermal fatigue life prediction models for SnAgCu solder joints." In: vol. 1. July 2004, 737–746 Vol.1. ISBN: 0-7803-8365-6. DOI: 10.1109/ECTC.2004.1319419.

[40] Ahmer Syed. "Updated Life Prediction Models for Solder Joints with Removal of Modeling Assumptions and Effect of Constitutive Equations." In: May 2006, pp. 1–9. ISBN: 1-4244-0275-1. DOI: 10.1109/ESIME.2006.1644010.

[41] Vasu Vasudevan and Xuejun Fan. "An Acceleration Model for Lead-Free (SAC) Solder Joint Reliability under Thermal Cycling." In: *Proceedings - Electronic Components and Technology Conference*. 2008. DOI: 10.1109/ECTC.2008.4549960.

[42] Vasu Vasudevan and Xuejun Fan. "An acceleration model for lead-free (SAC) solder joint reliability under thermal cycling." In: *2008 58th Electronic Components and Technology Conference*. 2008, pp. 139–145. DOI: 10.1109/ECTC.2008.4549960.

[43] Waloddi Weibull. "A statistical distribution function of wide applicability." In: *Journal of applied mechanics* (1951).

[44] Waloddi Weibull. *Fatigue testing and analysis of results*. Elsevier, 2013.

[45] Steffen Wiese, Ekkehard Meusel, and Klaus-Juergen Wolter. "Microstructural dependence of constitutive properties of eutectic SnAg and SnAgCu solders." In: Feb. 2003, pp. 197–206. ISBN: 0-7803-7791-5. DOI: 10.1109/ECTC.2003.1216277.

[46] Q. Zhang, A. Dasgupta, and P. Haswell. "Viscoplastic Constitutive Properties and Energy-Partitioning Model of Lead-Free Sn3.9Ag0.6Cu Solder Alloy." In: *Electronic Components and Technology Conference*. IEEE. 2003.