


## Article

# Circuit-Based Rainflow Counting Algorithm in Application of Power Device Lifetime Estimation

Tian Cheng \*, Dylan Dah-Chuan Lu  and Yam P. Siwakoti

School of Electrical and Data Engineering, Faculty of Engineering and IT, University of Technology Sydney, Building 11, Ultimo, NSW 2007, Australia; dylan.lu@uts.edu.au (D.D.-C.L.); yam.siwakoti@uts.edu.au (Y.P.S.)  
\* Correspondence: tian.cheng@student.uts.edu.au

**Abstract:** The software-assisted reliability assessment of power electronic converters is increasingly important due to its multi-domain nature and extensive parametric calculations. The rainflow counting algorithm is gaining popularity for its low relative error in device lifetime estimation. Nevertheless, the offline operation of the algorithm prevents most simulation software packages considering other parameters for the device under study, such as aging and the current state of health in the estimation, as it requires a complete loading profile to run recursive comparison. This also brings difficulties in realization in circuit simulators such as SPICE. To tackle the issue, an in-the-loop circuit-based rainflow counting algorithm is proposed in this paper, and applied to estimate the consumed lifetime of the MOSFET in a boost converter for illustration. Instantaneous electrical and thermal performances, and the accumulated stress of the device can be monitored. Not only does this assist in evaluating the state of health of a device, but also allows the possibility of integrating the aging into the lifetime evaluation. The method follows the four-point rainflow counting algorithms, which continuously compares three adjacent temperature fluctuations  $\Delta T_j$  to select full cycles for two rounds, and the remaining cycles are counted as half cycles. To validate the performance, a comparative analysis in terms of counting accuracy and simulation speed was performed alongside the proposed method, MATLAB<sup>®</sup> and also with a well-accepted half-cycle counting method. Reported results show that the proposed method has an improved counting accuracy compared to the half-cycle counting from 24% to 3.5% on average under different load stresses and length conditions. The accuracy can be effectively improved by a further 1.3–2% by adding an extra comparison round.

**Keywords:** circuit-based rainflow counting algorithm; multi-domain simulation; lifetime estimation application



**Citation:** Cheng, T.; Lu, D.D.-C.; Siwakoti, Y.P. Circuit-Based Rainflow Counting Algorithm in Application of Power Device Lifetime Estimation. *Energies* **2022**, *15*, 5159. <https://doi.org/10.3390/en15145159>

Received: 15 June 2022

Accepted: 13 July 2022

Published: 16 July 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Stress cycle counting algorithms translate complicated and irregular loading profiles into a set of organized cycles to facilitate stress accumulative calculations, and are essential in reliability assessments and lifetime prediction for power devices, such as power semiconductors, or converters/inverters, especially when they operate with critical and varying loads.

Various counting approaches such as level-crossing, peak counting and rainflow were introduced in [1–5]. Comparative discussions on peak counting algorithms (e.g., half-cycle, maximum edge and rising edge) and rainflow counting are given in [6] through thermo-mechanical FEM analysis, and the rainflow counting is reported to have the lowest relative error with 11% while the others are between 19–27% when running a certain mission profile. Rainflow counting was first introduced by Matsuishi and Endo in fatigue analysis in 1968 [7] by assuming the reversals as rain drops and drips off the pagoda roofs. This method incorporates closed hysteresis loops in stress–strain plots and considers them to be full cycles with corresponding range and mean value. It has also gained popularity in

dealing with critical thermal profiles and calculating lifetimes. For example, the lifetime of grid-connected PV inverters in [8,9], multi-level converters utilized in wind turbine systems [10–12] are evaluated with the help of this counting algorithm. In most of these studies, the counting is completed by inputting the thermal profile into the MATLAB<sup>®</sup> rainflow counting command for ease of implementation and fast computation speed. To the best of our knowledge, there are very few existing solutions which realize rainflow counting in a circuit simulation platform. Possible reasons may be that the traditional rainflow method is applied offline, while the circuit simulator follows the time sequence, which brings difficulties in comparing different reversals, especially when they are far away from each other.

There is a growing trend at present for implementing multi-domain or multi-discipline simulations; for instance, an increasing number of integrated electro-thermal models of power MOSFETs, IGBTs, etc., are provided by leading manufacturers such as Infineon and STMicroelectronics. It could be beneficial to include lifetime estimation into the simulation, such that users are allowed to monitor the device's electrical performance, thermal properties, accumulated damage and lifetime simultaneously and only in one simulator. Apart from this, studies have demonstrated the great impacts of aging and the current health state of a device on lifetime estimation [13–17]. To take them into consideration, it is necessary to know the instantaneous accumulated stress. In other words, an online counting method is required. Although the MATLAB<sup>®</sup> rainflow counting function provides a convenient implementation, it is an offline method which means a complete loading profile is required. To tackle this, online rainflow counting methods were proposed in [18,19]. All are stack-based recursive algorithms, which allow users to easily select and discard counted cycles in programming. These methods are well-suited to a real-time environment but have difficulty in being applied in a circuit simulator. As circuit simulators follow the time sequence, it is not possible to make any change on those already plotted points.

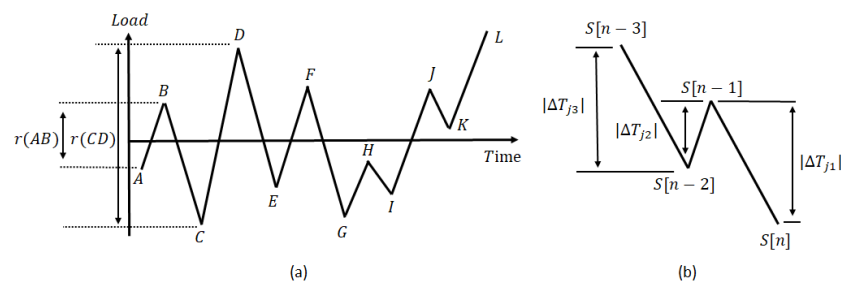
This paper proposes an easy-to-use and in-the-loop rainflow counting approach for power devices and power electronic circuits. The novelty of the proposed work is the implementation of the algorithm in a circuit simulator, which makes possible the in-the-loop monitoring of the electrical and thermal performances, thermal cycles counting and instantaneous evaluation of aging of a device. This work also addresses the issue in the conventional lifetime estimation that multi-domain models and stages are essential but provides a solution to integrate them together. In addition, to the best of our knowledge, there is no existing solution of using circuits for realizing the rainflow counting algorithm. Therefore, this work is proposed is to fill the gap of missing functionality and the integrated approach in existing circuit simulators, as well as to facilitate multi-discipline simulation in SPICE. Additionally, since the rainflow counting has the smallest relative error, the estimated lifetime evaluation is more reliable with this method. The realization of this algorithm is assisted by sample-and-hold function blocks, buffers, logic gates, behavioral set–reset flipflop and behavioral models. Since it is not possible for the circuit simulator to run the recursive method such as the stack-based implementation, simplification is made by only sorting full cycles for limited times. Although it cannot pick out all full cycles, it still has an improved counting accuracy as compared to other simple approaches, such as the half-cycle counting method. Additional comparison rounds can be added to improve the counting accuracy further.

## 2. Realization of Rainflow Counting Algorithm in SPICE

### 2.1. Principle of Rainflow Counting Algorithm

Different methodologies are proposed to achieve the rainflow counting in [2–5]. For the sake of simplicity without losing effectiveness, the principle of the four-point rainflow counting algorithm is briefly explained here and is also employed in realizing in SPICE. The load profile is first turned into a sequence of reversals as shown in Figure 1a. The reversals denote as the local minima and maxima with a change of sign, e.g., points A–L, while the

difference between the two reversals is described as range, e.g.,  $r(AB)$ . Then, three adjacent ranges are continuously compared to determine the counting action following the criteria that  $X_2 \leq X_3$  and  $X_2 \leq X_1$ . If satisfied, a closed loop is formed, and the range  $X_2$  is counted as a full cycle. Finally, reversals involved in the counted range will be discarded, and a new range between uncounted reversals will be similarly formed, and the comparisons keep on. For instance, once  $r(EF)$  in the loop of D–E–F–G in Figure 1a is counted, both E and F will be discarded, and the new range becomes  $r(DG)$ . By doing so, it is possible to pick out all of the full cycles in the load profile, and only a few uncounted half cycles remain. In summary, the principle of the four-point algorithm is simple, and since it requires recursive implementation, it is a more programming friendly algorithm.



**Figure 1.** (a) Example of reversal sequence; and (b) four-point rainflow algorithm.

## 2.2. Implementation Method in SPICE

Following the four-point rainflow counting principle, and applying it in counting thermal cycles as shown in Figure 1b, the ranges  $X_1$ ,  $X_2$  and  $X_3$  in the criteria represent the temperature swings  $|\Delta T_{j1}|$ ,  $|\Delta T_{j2}|$  and  $|\Delta T_{j3}|$ , respectively. Although the principle is straightforward, when adopted in SPICE, it still needs to address the recursive comparison issue. Here, the proposed method runs two comparison rounds to sort full cycles, and count all of the remaining cycles as half cycles. The detailed operation algorithm is shown in Figure 2 and the derivation steps and explanation of implementing the method in SPICE are described below.

1. The one round starts here: sample the data points  $T_j$  in the thermal profile, and obtain one sampling period  $T_s$  delay  $T_j[Zn^{-1}]$  by using the delay function. Subtract these two values to generate the temperature swing waveform  $\Delta T_j$  and calculate the swing amplitude  $|\Delta T_j|$ —in other words, the absolute value of  $\Delta T_j$ . Generate the required  $T_s$  delay waveforms for both of these values.
2. Compare the first-order swing amplitude  $|\Delta T_j[Zn^{-1}]|$  with the current value  $|\Delta T_j[Zn]|$  and the second-order  $|\Delta T_j[Zn^{-2}]|$ . Convert the comparison into a Boolean expression, and if it satisfies the criteria that  $X_2 \leq X_3$  and  $X_2 \leq X_1$ , then the counting waveform 1st generates a pulse valued at 1 V, indicating that it counts the  $|\Delta T_j[Zn^{-1}]|$  as one cycle. Calculate the corresponding  $T_m$  and  $Q$  of the counted cycles.
3. Generate a new sampling pulse which is made by inverting the 1st + 1st $[Zn^{-1}]$ , to sample  $T_j$  and construct a new  $T_{j(n+1)}$  waveform. This can effectively skip sampling the two points involved in the counted  $|\Delta T_j[Zn^{-1}]|$ , for example,  $S[n-2]$  and  $S[n-1]$ , while keeping and holding the uncounted values  $S[n-3]$  and  $S[n]$  in Figure 1b. Since the circuit simulator follows the time sequence, it is not possible to discard the already generated points such as the stack-based implementation. In addition, waiting until the previous round to finish counting is also unwise. As the action is simultaneously proceeding with the load changing, the overall number of points that should be discarded—or in other words the delay time—is unknown until the simulation is complete. Hence, the proposed method is used to mimic the discard function.
4. The two rounds starts here: same as step 1 with the newly generated  $T_{j(n+1)}$ .
5. At this time,  $|\Delta T_{j(n+1)}[Zn]|$  is no longer consecutive, however, zeros appear due to the discarded points in the previous round. This results in difficulties by simply

adopting the same method as in step 2. Thus, another solution is given here. Determine the growing trend of the temperature swing by comparing  $|\Delta T_{j(n+1)}[Zn]|$  with  $|\Delta T_{j(n+1)}[Zn^{-1}]|$ . A falling trend is indicated if the current  $|\Delta T_{j(n+1)}[Zn]|$  is smaller or equal to  $|\Delta T_{j(n+1)}[Zn^{-1}]|$ . Otherwise, it is following a rising trend. Based on the criteria that  $X_2 \leq X_3$  and  $X_2 \leq X_1$ ,  $X_2$  must have the smallest range. Thus, full cycles at the end of falling segments will always exist, just next to the following rising segment. As the one round discards even numbers of points, for instance, two points for 1 counted cycle, four points for two consecutive counted cycles, etc., the full cycles are most likely to occur in odd delay times, e.g.,  $[Zn^{-1/-3/-5}]$ . Therefore, the solution is to pick out full cycles by shifting the  $|\Delta T_{j(n+1)}|$  five times, from  $[Zn^{-1}]$  to  $[Zn^{-5}]$ , to determine the first cycle that overlaps with the rising segment. Within five  $T_s$  periods, it is capable of filtering out most of the full cycles in this round.

6. Unify the counted cycles to the same delay times  $[Zn^{-5}]$  and calculate their sum. Same as step 3 and generate with a new sampling pulse made by inverting the aforementioned pulses, and output a new  $T_{j(n+2)}$  waveform.
7. The three rounds starts here: same as step 1 with the newly generated  $T_{j(n+2)}$ .
8. Same as step 2, and the loop stops here. The remaining  $\Delta T_{j(n+2)}$  values are counted as half cycles. Calculate the corresponding  $T_m$  and  $Q$  of half cycles.

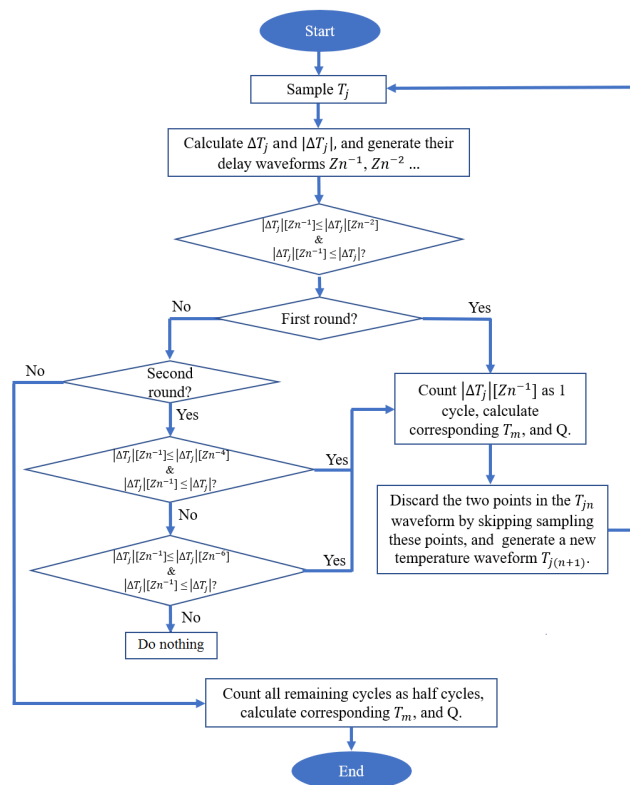


Figure 2. Operation of the proposed rainflow counting algorithm in LTspice®.

In summary, the proposed implementation of rainflow counting continuously compares the values of three adjacent temperature swings  $|\Delta T_{jn(n=1,2,3)}|$  for the first two rounds. Counting pulses will be generated during the comparison, and its inverted waveform will be used as the sampling pulse for producing the next round  $T_{j(n+1)(n=1,2)}$ . By doing so, the counted points can be skipped sampling, and the next round  $T_{j(n+1)(n=1,2)}$  waveforms are simplified. The remaining cycles in the round 3 are all counted as half cycles.

It should be noted that two rounds cannot filter out all full cycles. In fact, in theory, it requires numerous rounds to achieve it. However, the difficulty, complexity and simulation time of running this are expected to increase rapidly. Hence, the proposed method will

inevitably have small errors. Nevertheless, it is possible to modularize the two-round operation and add it as an additional round to improve the counting accuracy. As a trade-off, the simulation time will be increased. For illustration purposes, this paper only explains the three key steps of the proposed rainflow counting. The detailed implementation of this algorithm in a circuit simulator will be introduced in the following section.

### 2.3. Circuitry Analysis

The implementation of the algorithm is carried out in LTspice<sup>®</sup>, which is a user-friendly and free circuit simulator. In order to verify the effectiveness of the proposed model, it is applied to estimate the lifetime of the MOSFET in a free-running boost converter according to random load-profiles. The complete circuitry of the proposed counting method and the electro-thermal averaged boost converter model are shown in Figure A1 in Appendix A. Since LTspice<sup>®</sup> focuses on electrical domain simulation, the temperature (°C) and power loss (W) in the thermal domain are modeled by behavioral voltage (V) and current (A) sources in the electrical domain, respectively.

#### 2.3.1. Electro-Thermal Averaged Model

To count the thermal cycles, it is first necessary to have the thermal profile. An averaged boost converter is constructed for its advantage of fast operation speed over the switching converter. Modification is made to replace the fixed MOSFET on-state resistance  $R_{ds,on}$  in the conventional averaged model with a temperature-dependent variable resistance. The value is evaluated using (1), where  $R_{init}$ ,  $T_j$ ,  $\Delta R_{ds,on}$  and  $\alpha$  denote the initial  $R_{ds,on}$  value at 25 °C, MOSFET junction temperature  $T_j$  and the coefficient, respectively. In addition, since the averaged model is frequency independent, while the switching losses significantly contribute to the total power losses at high frequency, an extra voltage source  $V_{sw}$  which represents the voltage drop caused by the switching is added. A general estimation of switching losses is by using (2), where  $f_s$ ,  $T_r$ ,  $T_f$ ,  $I_{in}$  and  $V_{out}$  are namely the converter operation frequency, MOSFET turn-on rise time, turn-off fall time, input current, and the output voltage. Thus, a simple method to represent  $V_{sw}$  is by using (3) as a result of (2). Regarding the conduction losses, it can be easily calculated using (4), where  $i_m$  and  $D$  are MOSFET averaged current and duty cycle, respectively.

In the thermal model, a resistance network comprised of two thermal resistances: namely the junction to case  $R_{jc}$  and the case to ambient  $R_{ca}$  thermal resistances, and a voltage source represented by ambient temperature  $T_a$  is constructed. The inputs of this model are the conduction and switching losses described by behavioral current sources.

These two models are linked by feeding back the estimated MOSFET junction temperature  $T_j$  from thermal model to the electrical model, while the calculated losses  $P_{loss}$  from the electrical to thermal model. The detailed derivation steps can be found in previous works in [15,20].

$$R_{ds,on} = R_{init} \cdot \left(1 + \frac{\alpha}{100}\right)^{T_j - 25^\circ\text{C}} \quad (1)$$

$$P_{sw} = 0.5 \cdot f_s \cdot (T_r + T_f) \cdot V_{out} \cdot I_{in} \quad (2)$$

$$V_{sw} = 0.5 \cdot f_s \cdot (T_r + T_f) \cdot V_{out} \quad (3)$$

$$P_{cond} = i_m^2 \cdot R_{ds,on} \cdot D \quad (4)$$

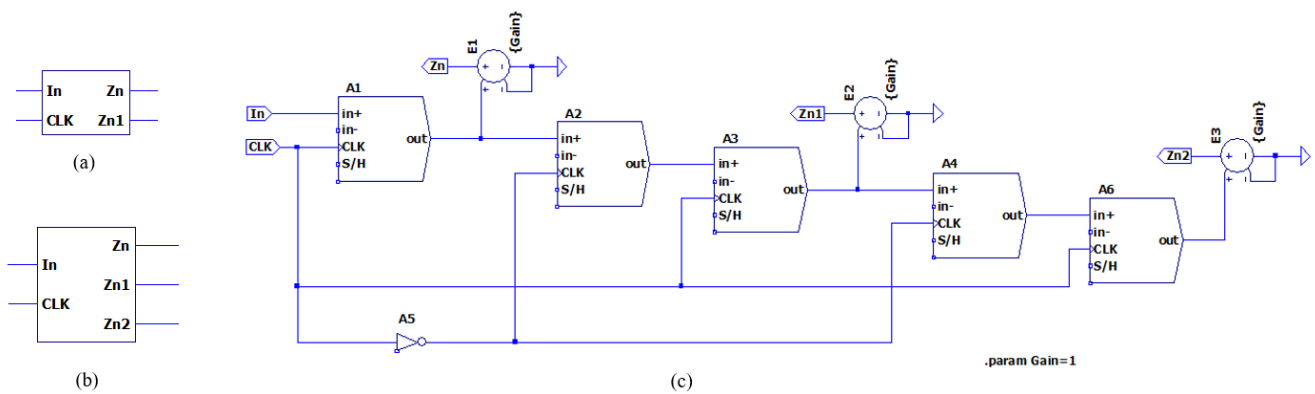
#### 2.3.2. Rainflow Counting Circuits

Before explaining the counting circuitries, sub-function blocks frequently employed in circuitry and the purposes of using them are introduced first.

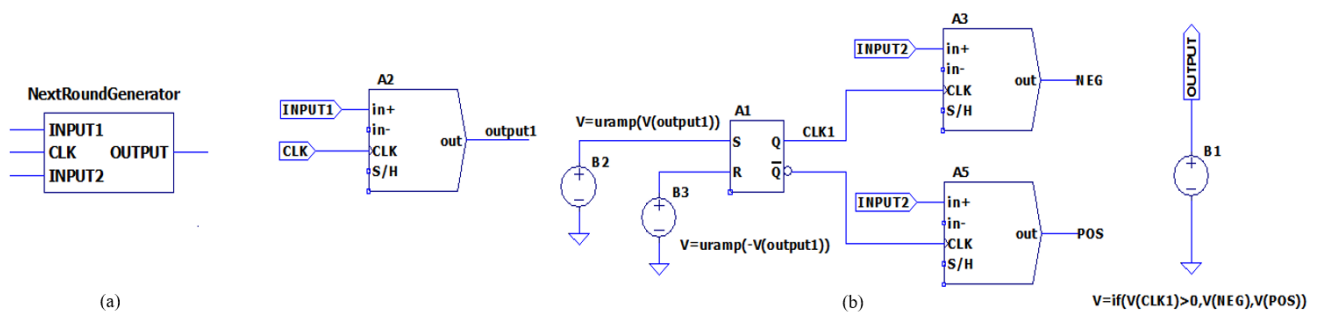
- Sample-and-hold with delay function block is shown in Figure 3. This submodule is composed of several sample-and-hold function and logic gates, aiming to sample the data and generate the 1, and 2  $T_s$  delay. The input  $in+$  and  $CLK$  are the data that need to be sampled and the sampling frequency, respectively. Due to the maximum allowed sample voltage being 10 V, the thermal profile needs to be scaled down before

being input to this function. It can be converted back to its original value by using a voltage-dependent voltage source  $E$  to provide a gain. The pre-defined value is 1.

- Behavioral Schmitt-triggered buffer with differential inputs is utilized in comparing the input amplitudes, and outputs a Boolean result.
- Logic gates including AND, NOT and OR gates are used in making decisions if a cycle should be counted or not.
- Behavioral models, which include behavioral current or voltage sources, allow the user to define the required functions. A few functions employed in this algorithm and also in the electrical circuits are explained in Table 1, and further description can be found in [21].
- Next round  $T_{j(n+1)}$  waveform generator, as shown in Figure 4, where input 1, input 2 and CLK are the uncounted  $\Delta T_{j(n)}$ ,  $T_{j(n)}$  and sampling pulse, respectively, while the output is the  $T_{j(n+1)}$  waveform. After sampling the  $\Delta T_{j(n)}$  values, it will be split into two groups which contain pure positive and negative values, respectively. Using the behavioral set–reset flipflop, the positive pulse can be extended until it meets the negative value, and vice versa. By doing so, the uncounted  $T_{j(n)}$  will be kept and extended until meeting the next uncounted value. The achievement of the next round waveform is by adding the positive and negative outputs together through an IF function.



**Figure 3.** Sample-and-hold with delay function block: (a) symbol of the block with  $1 T_s$  delay; (b) symbol of the block with two-order delay; and (c) schematic of (b).



**Figure 4.** Next round  $T_{j(n+1)}$  waveform generator: (a) symbol; and (b) schematic.

**Table 1.** Description and explanation of LTspice<sup>®</sup>.functions used in the proposed model.

Function	Description [21]	Purposes of These Functions in the Model
if(x,y,z)	If x is true, do y else z	Conditional statement
idt(x)	Integrate x	Accumulate stresses Q
delay(x,t)	x delayed by t	Generate a waveform with t cycles delayed for comparison
abs(x)	Absolute value of x	Calculate amplitude $\Delta T_j$
uramp(x)	If $x > 0$ , output x, else 0	Split positive and negative $\Delta T_j$

As can be seen from the rainflow counting circuits in Figure A1, the counting step in one round is simple. It only contains sampling  $T_j$ , calculating the temperature swing and its amplitude, and generating their delays waveforms by employing sample-and-hold with delay function blocks and behavioral voltage sources. The determination of a full cycle is by using the behavioral Schmitt-triggered buffer with differential inputs. Since it can only output the true value when the positive side is larger than the negative side, hence, an inverting comparison is made to achieve the  $X_2 \leq X_3$  and  $X_2 \leq X_1$  criteria.

Using two rounds is slightly complicated as zeros appear in  $|\Delta T_{j2}|$ , as explained in Section 2.2 step (5). A reference waveform  $|\Delta T'_{j2}[Zn^{-1}]|$  is generated to facilitate the comparison between two non-zero adjacent values, as they may possibly become separated and far away from each other in the original  $|\Delta T_{j2}|$ . The reference waveform is firstly achieved by detecting all zeros in  $|\Delta T_{j2}|$ , and utilized it as the sampling pulse to sample the  $|\Delta T_{j2}|$  again. By doing so, a new waveform, which samples the previous value of all zeros in  $|\Delta T_{j2}|$  are constructed. This action will result in  $1 T_s$  delay from the  $|\Delta T_{j2}|$ . Hence, to sum the  $|\Delta T_{j2}[Zn^{-1}]|$  and the aforementioned pulses together, the reference waveform  $|\Delta T'_{j2}[Zn^{-1}]|$  is formed. For instance, the original  $|\Delta T_{j2}[Zn^{-1}]|$  is  $[X_3 X_2 0 0 X_1]$ , while in reference  $|\Delta T'_{j2}[Zn^{-1}]|$ , it becomes  $[X_3 X_2 X_2 X_2 X_1]$ . With the help of the reference waveform, the falling and rising cycles can be distinguished.

Shifting the falling cycles  $V(\text{falling})$  by one  $T_s$  produces the  $V(F\_ZN1)$ , which is used to meet the rising cycles five times. After each time, cycles that overlap with the rising segment are deleted, and a new truncated falling cycles waveform will be generated and employed in the next shifting step. Comparisons are mainly made in  $1, 3$  and  $5 T_s$ , to search the full cycles that when  $X_2$  is  $1, 3, 5 T_s$  away from  $X_1$ , indicating cases that, no cycles, one cycle, and two consecutive cycles are counted between  $X_2$  and  $X_1$  in a previous round, respectively. Since one counted cycle contains two reversals, discarding them will result in two zeros in  $|\Delta T_{j2}|$ . With cycles in even numbers of shifting orders,  $2$  or  $4 T_s$  will be straightforwardly deleted as they will either be zeros or cycles  $X_3$  before the full cycles  $X_2$ .

The third round is the simplest one, which counts all of the remaining temperature swings' cycles as half cycles. Therefore, only a next-round generator, and a sample-and-hold with delay function block to produce the  $T_{j3}$  and its one  $T_s$  delay  $T_{j3}[Zn^{-1}]$  waveform are required. As they provide enough information for calculating the  $|\Delta T_j|$  and  $T_m$ .

The corresponding  $|\Delta T_j|$  and  $T_m$  of all cases above are calculated by using (5) and (6), respectively, where C and G indicate the counted cycles of each round, and the gain value, as the temperature is scaled down before inputting into the sample and hold function. For  $n$  and  $x$ , they are the corresponding round, and delay  $T_s$ .

$$\Delta T_j = C \cdot |\Delta T_{j(n)}| \cdot G \quad (5)$$

$$T_m = \frac{1}{2} \cdot C \cdot (T_{j(n)}[Zn^{-x}] + T_{j(n)}[Zn^{-x-1}]) \cdot G \quad (6)$$

The stress accumulator is utilized to evaluate the accumulated damage of a device after a number of thermal cycles, in other words, the consumed lifetime. It is made by connecting behavioral current sources in parallel which represent stresses from each round, and injecting them into a 1 ohm resistor to sum the stresses up. To calculate the stress of a device, it is necessary to know two terms, namely the device failure cycles under different operation conditions, and the number of cycles that a device has performed. Widely adopted modified Coffin–Manson model (7) can be used to evaluate the number of cycles to fail, in which  $N_f$ ,  $\Delta T_j$ ,  $T_m$ ,  $E_a$ ,  $k$  are the number of cycles before a device generates a fault under certain thermal stress, junction temperature swing, mean temperature, thermal activation energy and Boltzman constant, respectively. Regarding  $\delta$  and  $A_1$ , they are empirical coefficients. The MOSFET model under estimation is the IRFP340, and the coefficient  $\delta$  and  $A_1$  are  $-5.2776$  and  $4.9283 \times 10^{13}$ , respectively, given in [22]. The accumulated stress  $Q$  is determined by Miner’s rule (8), which is a commonly used model of describing fatigue-related failures. The rule evaluates the stress that a device undergoes after a number of thermal cycles. In the equation,  $N_i$  indicates the number of cycles that a device has performed and  $N_f$  indicates the corresponding number of cycles to fail under certain stress. The  $Q$  is continuously accumulated, and once it reaches one, it reaches its threshold of the end of its useful lifetime.

$$N_f = \Delta T_j^\delta A_1 e^{\frac{E_a}{kT_m}} \quad (7)$$

$$Q = \sum_{i=1}^n \frac{N_i}{N_f} \quad (8)$$

### 3. Simulation Results

#### 3.1. Simulation Waveform

The simulation results of applying the proposed method to a DC/DC boost converter with a 700 s (s) random load are given in Figures 5–7 as an illustrative example. The load  $V(\text{load})$  changes every 5 s, and only contains reversal points. The  $V(\text{tj\_mos})$  and  $V(\text{rds\_ins})$ , namely the simulated MOSFET junction temperature  $T_j$ , and the instantaneous  $R_{ds,on}$ , share the same varying trend. As they follow the (1) that high operation temperature will result a high on-state resistance, and it in turn will cause heavy power losses.  $V(\text{tj\_zn})$ ,  $V(\text{tj2\_zn})$ , and  $V(\text{tj3\_zn})$  are the sampled waveforms of  $V(\text{tj\_mos})$ , and the generated 2 and 3 rounds  $T_j$  after skipping sampling counted points. As  $V(\text{tj\_zn})$  and  $V(\text{tj2\_zn})$  will be used in the next round sample-and-hold function, the gain is 1 for simplicity. The true value needs to be multiplied by a gain of 25. A fairly clear observation on the simplification of the latter round as compared to the previous one can be found.

Key waveforms utilized in 1 and 2 rounds are shown in Figure 6 to better explain the counting actions.  $V(\text{tj\_zn})$ ,  $V(\text{adt\_zn1})$  and  $V(\text{1st})$  indicate the 1 round sampled  $T_j$ , the absolute value of the temperature swing  $|\Delta T_{j(n+1)}[Zn^{-1}]|$  and the counting waveform. It is easy for this round to determine the full cycles due to its ease of comparison. The counted cycles can either appear at the middle value  $X_2$ , which corresponds to the  $|\Delta T_{j(n+1)}[Zn^{-1}]|$ , or alternatively, at the third point  $X_3$ , which is  $|\Delta T_{j(n+1)}[Zn]|$ . To facilitate the later stress calculation, the second point corresponding waveform is always selected.

While with two round,  $V(\text{adt2\_zn1})$  and  $V(\text{adt2\_zn1\_ref})$  are the  $|\Delta T_{j2}[Zn^{-1}]|$ , and its corresponding reference waveform  $|\Delta T_{j2'}[Zn^{-1}]|$ . With the help of reference waveform,  $|\Delta T_{j2}[Zn^{-1}]|$  can be divided into two groups, namely  $V(\text{falling})$  and  $V(\text{rising})$  with cycles only in a declining/increasing in it. The  $V(\text{2nd})$  displays all of the counted cycles in two rounds, with  $5 T_s$  delay time away from the  $|\Delta T_{j2}[Zn^{-1}]|$ .



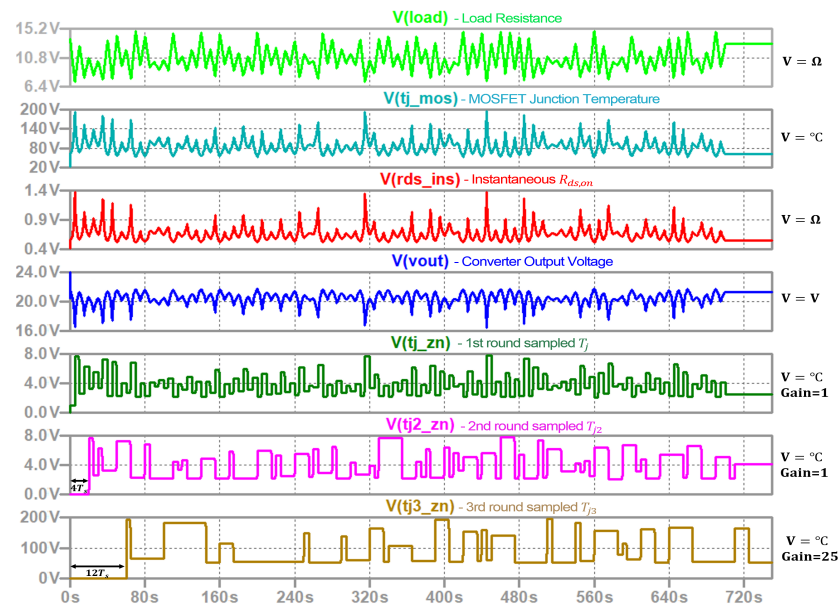


Figure 5. Waveforms of the proposed method of running a 700 s randomly generated load as an illustrative example.

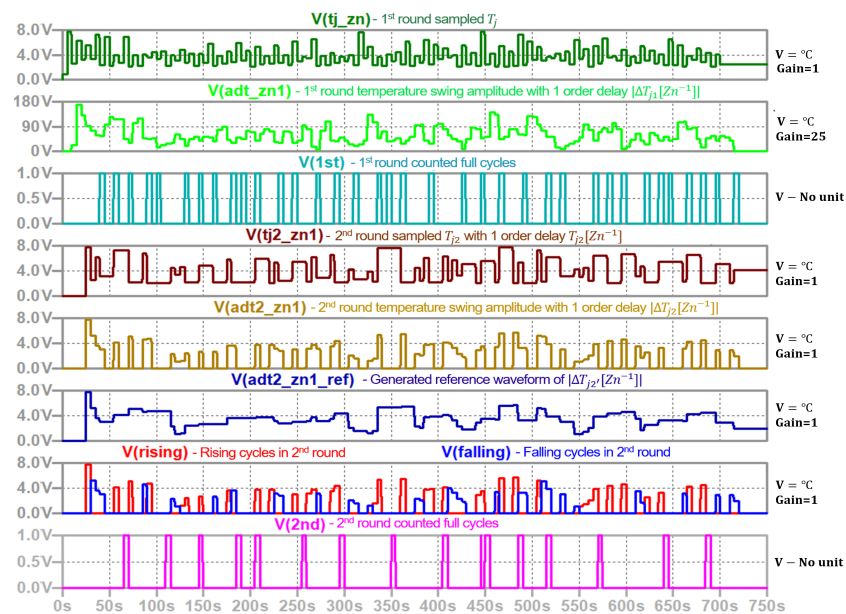


Figure 6. Example waveforms of depicting the counting actions of the 1 and 2 rounds for explanation.

Stress accumulation performance is shown in Figure 7. Since the proposed rainfall counting is an online method, a continuous accumulation of stresses from one, two and three rounds with the simulation can be seen. Both steep and flat increases occur in all of the stresses. In addition, the stresses from half cycles take the highest portion in the overall accumulated one. The reason of these can be found from (7), where the  $\Delta T_j$  is the dominant term. Hence, the steepness is highly dependent on the temperature swing  $\Delta T_j$ . As the first and second round filtered out small full cycles while the large  $\Delta T_j$  cycles are left, the half cycles contribute the most.

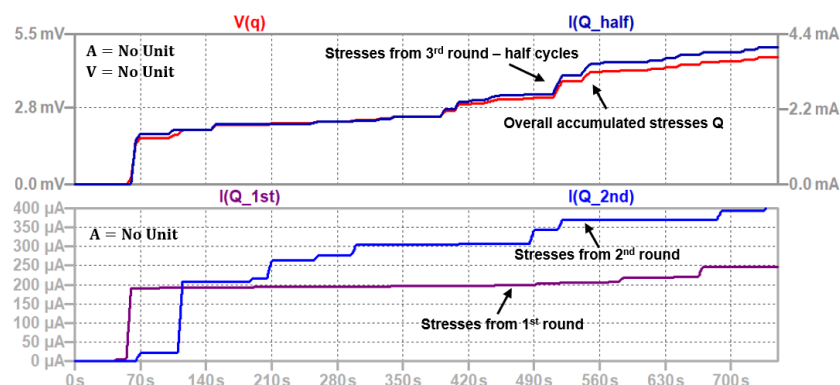


Figure 7. Accumulated stress by 1, 2, and 3 rounds, respectively.

### 3.2. Evaluation of the Proposed Method

To evaluate the proposed method in terms of counting accuracy and simulation speed, several groups of long-term loads with different stresses and lengths are simulated. All of the loads change every 5 s, and only contain reversal points. The simulated accumulated stresses are compared with both MATLAB® and half-cycle peak-through counting method simulation results for their wide acceptance in lifetime evaluation. Furthermore, the operation principle of three round in the proposed method is the same as the half-cycle counting approach. The detailed results are shown in Table 2, in which the accumulated stress Q has a factor of  $10^{-3}$ , while the proposed (3) and (4) indicate the simulation results of the proposed method with three and four rounds operation, respectively. An extra two round is added in the proposed (4), to evaluate the improvement on counting accuracy as compared to the three rounds. The MATLAB® results are adopted as the reference.

Table 2. Comparison of the simulation accuracy and speed among half-cycle counting, proposed rainflow counting (three and four rounds), and MATLAB® rainflow counting algorithm with different stresses’ long-term loads.

	High Stress (50–200 °C)			Medium Stress Load (50–160 °C)			Small Stress Load (50–110 °C)								
	20 ks Load			20 ks Load			50 ks Load			20 ks Load			100 ks Load		
	Q	Error	Time	Q	Error	Time	Q	Error	Time	Q	Error	Time	Q	Error	Time
MATLAB®	161	-	-	18.6	-	-	59.5	-	-	1.02	-	-	5.1	-	-
Half-cycle	122.5	23.7%	5 s	15.4	17.2%	5.7 s	49.8	16.3%	13 s	0.792	25.5%	7.2 s	3.49	31.6%	21 s
Proposed (3)	155.3	3.54%	91 s	17.85	4.03%	86 s	57.4	3.5%	198 s	0.99	3%	77 s	4.91	3.7%	376 s
Proposed (4)	158.3	1.7%	141 s	18.1	2.7%	137 s	58.17	2.2%	362 s	1.01	1%	144 s	5.01	1.8%	632 s

Note that: Q has a factor of  $10^{-3}$ . Proposed (3) and (4) indicate the result of the proposed method with 3 rounds and 4 rounds (run twice), respectively.

#### 3.2.1. Accuracy

Three conclusions can be obtained from Table 1. Firstly, the proposed method (result of the three-round operation for illustration) has a higher counting accuracy than the half-cycle counting approach for all of the tested loads. The error in the former is between 3% and 4%, while the latter is approximately 16.3–31.6%. Since the rainflow counting takes large temperature swing cycles into account, while the half-cycle counting does not, it is expected that higher counting accuracy will be obtained using the rainflow method.

Secondly, the counting method (result of the three-round operation for illustration) is effective regardless of the load stress and length. Two groups of simulation are carried out here. (1) Three load profiles with the same 20 kiloseconds (ks) length but different stress levels, namely high, medium, and low stress loads, with temperature swings of 50–200 °C, 50–160 °C, and 50–110 °C are tested. The calculated errors of these loads give similar results, between 3% and 4%. Thus, the stress level has little impact on the counting accuracy. (2) The impact of the load length on counting accuracy is also tested, aiming to check whether the errors will be accumulated with the increase in load length. Two

cases are tested here: (a) Medium stress load profiles with 20 ks and 50 ks; and (b) Small stress load profiles with 20 ks and 100 ks length are simulated, respectively. No significant accumulation of errors can be observed. Hence, the counting accuracy is independent of the load length.

Thirdly, it can also be observed that adding an extra round can effectively improve the accuracy by 1.3–2% further, generally from 3 to 4% in three rounds to 1–3%.

### 3.2.2. Simulation Time

The simulation is carried out on a laptop computer, with Inter(R) Core(TM) i7-7600U CPU. The reported elapse time of running the half-cycle algorithm with these 20–100 ks loads is between 5 s and 21 s, while for the proposed (3), it is between 77 s and 367 s. The half cycle counting has the advantage of its fast simulation speed, as it does not require any comparison; hence, less circuitries are used. For the proposed method, as the trade-off for the improved accuracy, the simulation speed is sacrificed as compared to the half-cycle counting method. Moreover, an increase of more than 50% in simulation time can be achieved by adding an additional round in proposed (4). Nevertheless, given the 100 ks load profile with 20,000 points and finishing both electro-thermal modeling and thermal cycles counting within less than 7 min is not unacceptable. A high performance computational system can also be used if a high speed is required. Since the MATLAB<sup>®</sup> rainflow counting is offline, its operation speed is not considered here.

## 4. Conclusions

This paper presents a circuit-based rainflow counting algorithm in device lifetime estimation application, aiming to achieve in-the-loop and high accuracy counting, whilst realizing multi-domain simulation in a circuit simulator. The proposed method follows the principle of the four-point rainflow counting algorithm, which continuously compares adjacent temperature swings in a thermal profile to figure out full cycles for two rounds. Counting pulses will be generated during the comparison, and its inverse will be used as the sampling pulse for producing a simplified thermal profile  $T_{j(n+1)}$  for the next round, where counted reversals are discarded. The remaining cycles will be counted as half cycles. To verify the performance of the proposed method, it is applied to estimate the lifetime of an operating MOSFET in a boost converter. The simulated accumulated stress is compared with the MATLAB<sup>®</sup> simulation results and also with the half-cycle peak-through counting method. Results show that the proposed method has an improved counting accuracy with 3–4% errors as compared to the half-cycle counting method which is 16.3–31.6% under different load conditions. It is also reported that a further refinement of the accuracy of 1.3–2% can be achieved by adding an extra comparison round. Simulation speed will be inevitably increased as a trade-off for the enhanced counting accuracy. Nevertheless, with 100 ks, the load profile finished simulating in less than 7 min is still acceptable. A high-speed computing system can also be utilized to compensate this effect if the simulation speed is a requirement.

**Author Contributions:** Conceptualization, methodology, software, validation and writing—original draft preparation, T.C.; writing—review and editing, D.D.-C.L. and Y.P.S.; supervision, D.D.-C.L. and Y.P.S. All authors have read and agreed to the published version of the manuscript.

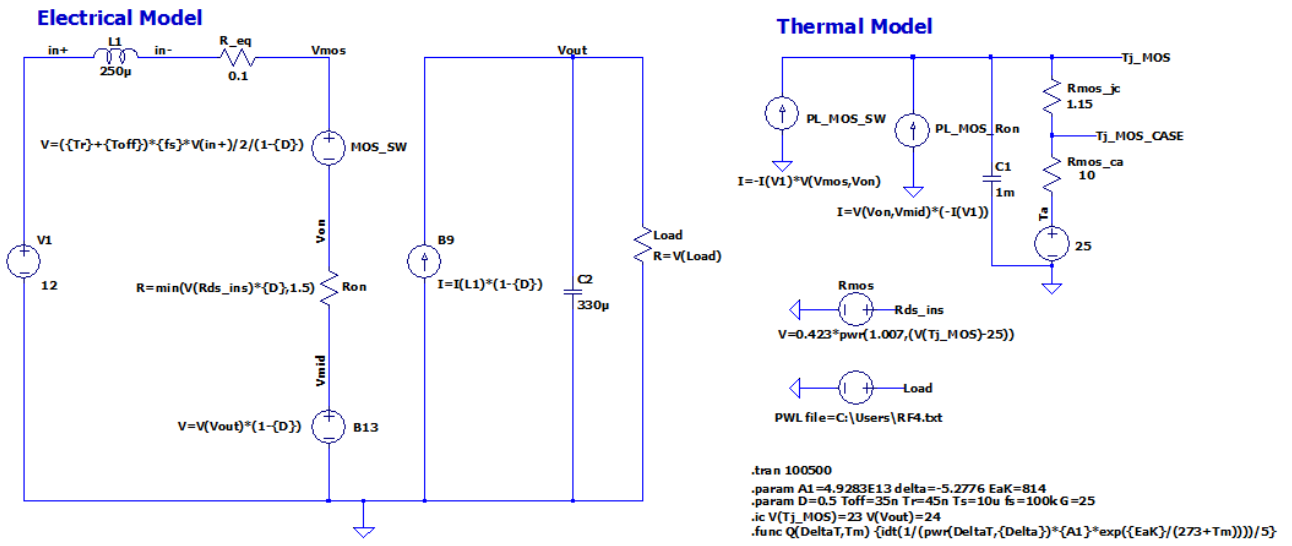
**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

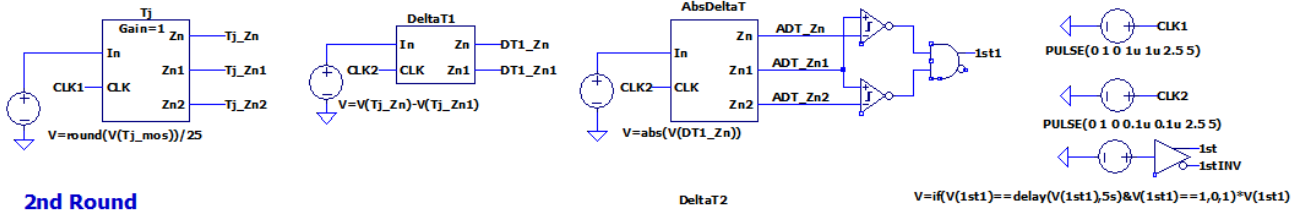
**Informed Consent Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

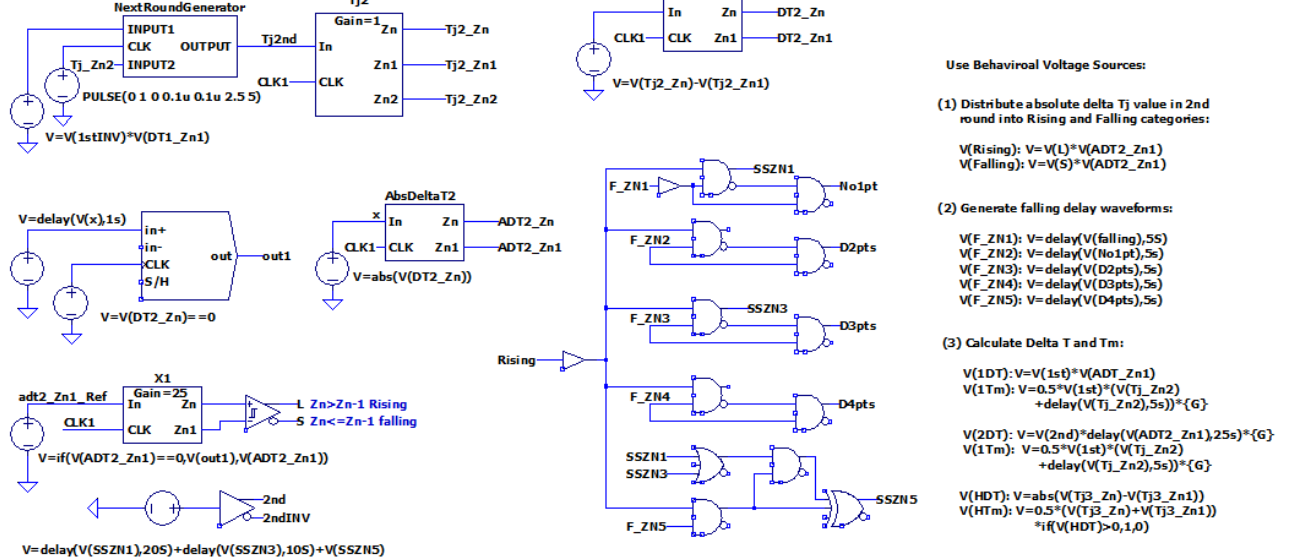
### Appendix A. Schematic Diagram



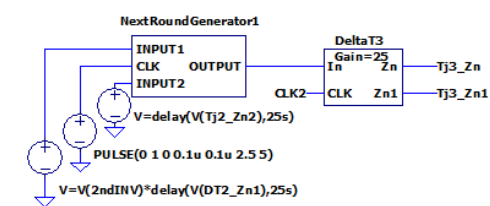
#### Rainflow Counting Circuit 1st Round



#### 2nd Round



#### 3rd Round



#### Stress Accumulator

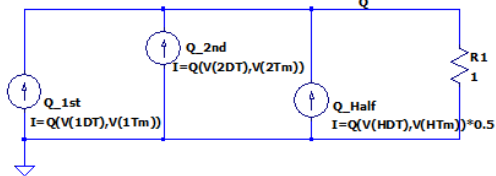


Figure A1. Schematic of the implementation of the proposed rainflow counting algorithm in counting the thermal cycles of a MOSFET in a running averaged boost converter.

## References

1. Ciappa, M.; Carbognani, F.; Fichtner, W. Lifetime prediction and design of reliability tests for high-power devices in automotive applications. *IEEE Trans. Device Mater. Reliab.* **2003**, *3*, 191–196. [CrossRef]
2. ASTM E1049-85; Standard Practices for Cycle Counting in Fatigue Analysis. ASTM: West Conshohocken, PA, USA, 2011.
3. Application Notes AN2019-05-“PC and TC Diagrams”; Infineon Technologies: Neubiberg, Germany, 2019.
4. Matlab, Rainflow Counts for Fatigue Analysis. Available online: <https://au.mathworks.com/help/signal/ref/rainflow.html> (accessed on 14 June 2022).
5. McInnes, C.H.; Meehan, P.A. Equivalence of four-point and three-point rainflow cycle counting algorithms. *Int. J. Fatigue* **2008**, *30*, 547–559. [CrossRef]
6. Mainka, K.; Thoben, M.; Schilling, O. Lifetime calculation for power modules application and theory of models and counting methods. In Proceedings of the 2011 14th European Conference on Power Electronics and Applications, Birmingham, UK, 30 August–1 September 2011; pp. 1–8.
7. Matsuishi, M.; Endo, T. Fatigue of metals subjected to varying stress. *Jpn. Soc. Mech. Eng.* **1968**, *68*, 37–40.
8. Sangwongwanich, A.; Yang, Y.; Sera, D.; Blaabjerg, F. Lifetime Evaluation of Grid-Connected PV Inverters Considering Panel Degradation Rates and Installation Sites. *IEEE Trans. Power Electron.* **2018**, *33*, 1225–1236. [CrossRef]
9. Shen, Y.; Chub, A.; Wang, H.; Vinnikov, D.; Liivik, E.; Blaabjerg, F. Wear-Out Failure Analysis of an Impedance-Source PV Microinverter Based on System-Level Electrothermal Modeling. *IEEE Trans. Ind. Power Electron.* **2019**, *66*, 3914–3927. [CrossRef]
10. Shipurkar, U.; Lyrakis, E.; Ma, K.; Polinder, H.; Ferreira, J.A. Lifetime Comparison of Power Semiconductors in Three-Level Converters for 10-MW Wind Turbine Systems. *IEEE J. Emerg. Sel. Top. Power Electron.* **2018**, *6*, 1366–1377. [CrossRef]
11. Liu, H.; Ma, K.; Qin, Z.; Loh, P.C.; Blaabjerg, F. Lifetime Estimation of MMC for Offshore Wind Power HVDC Application. *IEEE J. Emerg. Sel. Top. Power Electron.* **2016**, *4*, 504–511. [CrossRef]
12. Ma, K.; Liserre, M.; Blaabjerg, F.; Kerekes, T. Thermal Loading and Lifetime Estimation for Power Device Considering Mission Profiles in Wind Power Converter. *IEEE Trans. Power Electron.* **2015**, *30*, 590–602. [CrossRef]
13. Huang, H.; Mawby, P.A. A Lifetime Estimation Technique for Voltage Source Inverters. *IEEE Trans. Power Electron.* **2013**, *28*, 4113–4119. [CrossRef]
14. Sintamarean, N.; Blaabjerg, F.; Wang, H.; Iannuzzo, F.; de Place Rimmen, P. Reliability Oriented Design Tool For the New Generation of Grid Connected PV-Inverters. *IEEE Trans. Power Electron.* **2015**, *30*, 2635–2644. [CrossRef]
15. Cheng, T.; Lu, D.D.C.; Siwakoti, Y.P. A MOSFET SPICE Model With Integrated Electro-Thermal Averaged Modeling, Aging, and Lifetime Estimation. *IEEE Access* **2021**, *9*, 5545–5554. [CrossRef]
16. Lai, W.; Chen, M.; Ran, L.; Xu, S.; Jiang, N.; Wang, X.; Alatisse, O.; Mawby, P. Experimental Investigation on the Effects of Narrow Junction Temperature Cycles on Die-Attach Solder Layer in an IGBT Module. *IEEE Trans. Power Electron.* **2017**, *32*, 1431–1441. [CrossRef]
17. Lai, W.; Chen, M.; Ran, L.; Alatisse, O.; Xu, S.; Mawby, P. Low  $\Delta T_j$  Stress Cycle Effect in IGBT Power Module Die-Attach Lifetime Modeling. *IEEE Trans. Power Electron.* **2016**, *31*, 6575–6585. [CrossRef]
18. Musallam, M.; Johnson, C.M. An Efficient Implementation of the Rainflow Counting Algorithm for Life Consumption Estimation. *IEEE Trans. Reliab.* **2012**, *61*, 978–986. [CrossRef]
19. Chen, Z.; Gao, F.; Yang, C.; Peng, T.; Zhou, L.; Yang, C. Converter Lifetime Modeling Based on Online Rainflow Counting Algorithm. In Proceedings of the 2019 IEEE 28th International Symposium on Industrial Electronics (ISIE), Vancouver, BC, Canada, 12–14 June 2019; pp. 1743–1748.
20. Cheng, T.; Lu, D.D.C.; Siwakoti, Y.P. Electro-Thermal Average Modeling of a Boost Converter Considering Device Self-heating. In Proceedings of the 2020 IEEE Applied Power Electronics Conference and Exposition (APEC), New Orleans, LA, USA, 15–19 March 2020; pp. 2854–2859.
21. LTspice®B Sources (Complete Reference). Available online: [http://ltwiki.org/index.php?title=B\\_sources\\_\(complete\\_reference\)](http://ltwiki.org/index.php?title=B_sources_(complete_reference)) (accessed on 14 June 2022).
22. Dusmez, S.; Duran, H.; Akin, B. Remaining Useful Lifetime Estimation for Thermally Stressed Power MOSFETs Based on on-State Resistance Variation. *IEEE Trans. Ind. Appl.* **2016**, *52*, 2554–2563. [CrossRef]