FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Development of an Advanced Protection Concept for Automotive Wire Harnesses

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Technische Informatik

eingereicht von

Christopher Gabriel, BSc
Matrikelnummer 0725226

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dipl.-Ing. Dr.rer.nat. Radu Grosu
Mitwirkung: Univ.Ass. Dipl.-Ing. Denise Ratasich
Dr. techn. Hans-Peter Kreuter (Infineon Technologies Austria AG)

Wien, 22. August 2016

Christopher Gabriel	Radu Grosu

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Development of an Advanced Protection Concept for Automotive Wire Harnesses

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Computer Engineering

by

Christopher Gabriel, BSc
Registration Number 0725226

to the Faculty of Informatics

at the Vienna University of Technology

Advisor: Univ.Prof. Dipl.-Ing. Dr.rer.nat. Radu Grosu
Assistance: Univ.Ass. Dipl.-Ing. Denise Ratasich
Dr. techn. Hans-Peter Kreuter (Infineon Technologies Austria AG)

Vienna, 22nd August, 2016

Christopher Gabriel | Radu Grosu

Erklärung zur Verfassung der Arbeit

Christopher Gabriel, BSc
Kaiserin Elisabethstraße 5/3
2344 Maria Enzersdorf

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 22. August 2016

Christopher Gabriel

Danksagung

Ich möchte mich bei allen Personen der Infineon Austria AG dafür bedanken, dass ich bei Ihnen an diesem Thema arbeiten durfte. Speziell möchte ich hierbei Hardy Stoelben und Hans-Peter Kreuter aus der Application Engineering Abteilung und Luca Petruzzi, Christian Djelassi und dem ehemaligen Kollegen David Jansson aus der Design Abteilung danken. Es war mir ein Vergnügen mit ihnen zu arbeiten.

An der TU Wien möchte ich meinen Dank an Prof. Radu Grosu und seinen Assistenten widmen, allen voran Denise Ratasich.

Schlußendlich gilt mein besonderer Dank meinen Eltern, welche mich stets in allen Lebenslagen unterstützten.

Acknowledgements

I want to thank all the people at Infineon Technologies Austria AG for being able to work on this topic with them. In particular Hardy Stoelben and Hans-Peter Kreuter from the application engineering group and Luca Petruzzi, Christian Djelassi and former employee David Jansson from the design group. It was a pleasure to work with them.

From the Vienna University of Technology I want to thank Prof. Radu Grosu and his assistants, especially Denise Ratasich.

Finally I want to thank my parents for always supporting me in every aspect of life.

Kurzfassung

Die heutzutage in der Automobilindustrie eingebauten Kabelbäume haben eine Gesamtkabellänge von mehreren Kilometern. Durch Unterschiede im thermischen Verhalten zwischen Kabel und den verwendeten Schmelzsicherungen sind die verwendeten Kabel überdimensioniert um den Sicherheitsvorschriften in der Automobilindustrie in Hinblick auf Kabelsicherung zu entsprechen. Dies hat zur Folge, dass die Kabel nicht optimal genutzt werden können, wodurch sich ein großes Kosteneinsparungspotential ergibt.

In dieser Diplomarbeit wird die effiziente Implementierung einer elektronischen Sicherung basierend auf dem thermischen Modell des verwendeten Kabels behandelt. Wenn die Temperatur zu hoch ist unterbricht die Sicherung den Stromfluss um einen Schaden zu vermeiden. Nachdem das generelle Konzept vorgestellt wird, werden verschiedene Optimierungsmethoden erläutert. Eine optimierte Temperaturberechnung ist wichtig im Himblick auf den Ressourcenverbrauch und damit den Herstellungskosten. Das vorgeschlagene Konzept erlaubt eine optimale Ausnutzung der Kabel, was eine signifikante Gewichtsreduktion des Kabelbaums ermöglicht. Verschiedene optimierte Berechnungsmethoden werden vorgestellt und ein funktionsfähiger Demonstrator wurde mit einem Microcontroller realisiert. Zusätzlich wurden Hardware Module entwickelt um den Ressourcenverbrauch einer Hardwarelösung zu untersuchen.

Eine abschließende Evaluierung, bei der die Ergebnisse der verschiedenen Methoden mit der tatsächlich gemessenen Kabeltemperatur verglichen werden, zeigt, dass die verschiedenen vorgestellten Methoden sich zur Temperaturberechnung eignen und damit eine effiziente Sicherungsimplementierung ermöglichen.

Abstract

Wire harnesses in the automotive area contain a multitude of wires nowadays with a combined length of several kilometres. In order to fulfill the safety requirements in the automotive area and considering the different thermal properties the used wires and fuses are often oversized. This yields a tremendous potential for reducing costs since the wires are not being used to capacity.

This thesis covers the efficient implementation of an electronic fuse based on the thermal model of the used wire. The thermal model is used to calculate the temperature difference between the ambient temperature and the wire temperature. If the temperature of the wire is too high the fuse can open the circuit and prevent damage to the wire. After explaining the general concept, methods of optimization are introduced. An efficient temperature calculation is important to reduce resource consumption and manufacturing costs. The proposed concept allows to use wires up to capacity and therefore significant weight reduction is possible. Several optimized calculation options are presented and a functional demonstrator was built as a proof of concept based on a micro controller. Additionally, hardware modules were developed in order to compare resource consumption of the different methods in the case of a hardware solution.

A final evaluation comparing the optimized methods with actual temperature measurements shows that the different optimizations are all suitable to calculate the wire temperature and thus allow an efficient fuse design.

Contents

CHAPTER 1

Introduction

While the technology used within a car has changed significantly over the last decades, the way wires, electronic and electrical components are protected hasn't really changed at all. Considering that more and more electrical components are found in modern cars, the problem of protected wiring grows, as „point-to-point connections along several electromechanical fuse and relay distribution levels often result in supply paths of more than 10 meters. Resistance, voltage drop and power loss all increase with path length and wires with bigger cross-sections are frequently used to compensate this." [SFWLR14, 1] As an example the break lights in the rear of the car require a connection from the fuse box, which is usually located somewhere underneath the dashboard, all the way to the break lights at the back of the car. Because of the larger distance, with today's car wiring a larger wire cross-section is used than actually required. A larger cross-section might also be required because the characteristics of a conventional meltdown fuse would not be able to fully protect the wire across the whole operating current range, since the isothermals of a meltdown fuse are typically of different shape than those of wires, as can be seen in figure 1.1. In the figure one can see the isothermals of a $0.75\,mm^2$ wire together with the minimum and maximum opening time for a 10 A automotive meltdown fuse taken out of the manufacturer's catalog. As one can see, the minimum and maximum opening times vary tremendously and for currents below 19 A the fuse is not guaranteed to open before the wire is operated out of specification risking damage.

The practice of increasing the wire cross-section ensures that it is protected but it comes at a cost. Not only are the actual costs of the thicker wire higher but the overall weight of the wire harness is increased as well. This extra weight has a negative effect on the fuel efficiency of a car.

Additional care has to be taken when moving towards an 48V system in the automotive industry. With this increased voltage the possibility of arcs cannot be neglected. As [EMW+14] states „Conventionally, electric lines are protected by fuses. However, since an arc represents an additional resistance in the line, the short-circuit current is limited

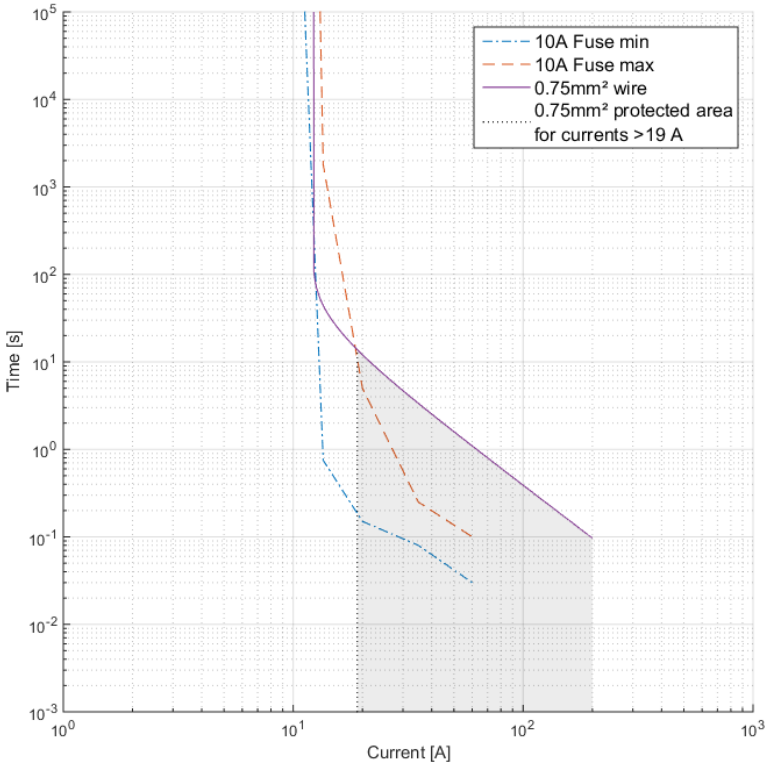


Figure 1.1: Fuse and wire mismatch.
The 0.75 mm^2 wire is only guaranteed to be protected in it's dynamic section for currents larger than 19 A with the selected 10 A fuse

such that the fuse may not be triggered.“ Additionally [otDPM05, 53] states that „nearly three-quarters of accidental car fires are due to vehicle defects, with defects in wiring and batteries being the biggest single cause.“

It is therefore necessary to develop new advanced protection systems which allow new wiring topologies resulting in shorter wire lengths and also having the possibility to detect arcing.

The goal of this thesis is to provide a proof of concept that an advanced wire protection can be implemented without a floating-point unit (FPU) with a very simple microcontroller unit (MCU) or as a hardware solution within an field-programmable gate array (FPGA) or as an application specific integrated circuit (ASIC), resulting in an economically viable digital fuse consisting of very few or even just one component.

1.1 Structure

Regarding the structure of this document there will first be an introduction into state of the art wire harness protection and load connection in chapter 2. Following that is a collection of related work with a short summary for every piece of work mentioned. The related work further shows how and why this topic is of importance for the automotive industry when considering the specifications and regulations which are present in this business area.

Chapter 3 will first deal with the principal goal of a fuse and then give an overview of concepts and models being used in the following chapters regarding design and implementation.

The actual design decisions and optimizations made are discussed in chapter 4. Since the goal is to achieve an implementation using the least resources possible multiple implementation designs are proposed. The implementation of these designs both in software using the C programming language and in hardware using Very High Speed Integrated Circuit Hardware Description Language (VHDL) targeting a MCU or FPGA / ASIC solution, respectively, is dealt with in chapter 5.

An evaluation of the different designs follows in chapter 6. It will show results concerning resource consumption and accuracy of the implemented versions. A prototype fuse which has been developed as a demonstrator is presented and compared to the simulated results.

Finally chapter 7 gives a conclusion and briefly explains what challenges remain which are not covered in this document.

CHAPTER 2

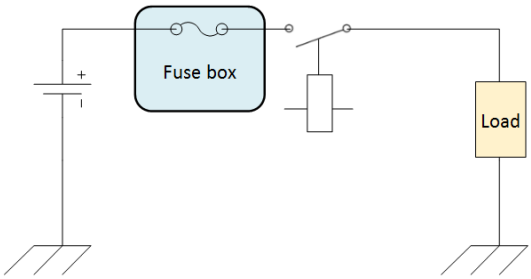
State of the art

Before going into details about the method presented in this thesis, a short overview of the state of the art wiring and wiring protection is given.

2.1 Current wiring and protection

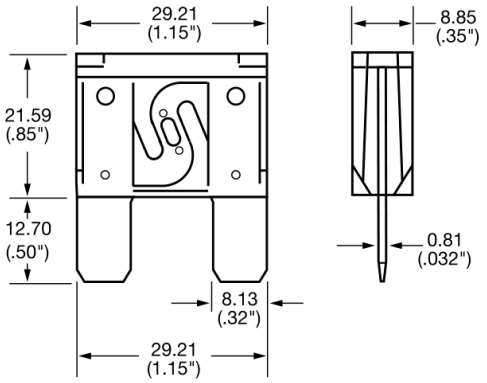
The current state of the art concerning load wiring is show in figure 2.1a. Between the battery and the load, there are some first stage fuses (omitted in the figure), then a load specific fuse, which is resident in the fuse box, and in case of a switchable load an additional relay. For every load there exists a dedicated fuse within the fuse box protecting the wiring used to connect the load and the load itself. Since modern cars do not only have electric loads required for basic operation of the vehicle itself but also for comfort and entertainment, the amount of fuses in modern cars is very large. Figure 2.1b shows one of two fuse boxes in the passenger compartment of a VW Multivan.

Conventional automotive fuses, as depicted in figure 2.1c, are meltdown fuses. They contain a fusible link in some kind of housing and are designed such that the metal strip connecting the two terminals melts and thus breaks the connection if the operating conditions are exceeded. The current-time characteristics which describe the time until the fusible link melts and interrupts the circuit, however, are not constant but vary between a documented minimum and maximum. When selecting a fuse for a given wire, one has to make sure that the fusible link breaks before the wire takes any damage. Due to the different thermal characteristics of fuse and wire, as shown in figure 1.1, wires are usually over dimensioned to ensure this. As a result the wire harness in all modern cars is to a certain degree over dimensioned regarding the requirements of the load. Being able to use wires of smaller diameter and using them to capacity by designing a fuse based on the wires' properties the mass of the wire harness can be reduced to a certain degree resulting in better fuel efficiency.



(a) State of the art: the fuse box contains separate fuses for different loads from where they are connected to the battery via an optional relay

(b) One of two fuse boxes inside the passenger compartment of a VW Multivan



(c) Typical automotive fuse used in cars. Source: Littelfuse, Inc. catalog

Figure 2.1

2.2 Related work

The concept of having a digital fuse is not new but so far no economically viable solution outperforming conventional meltdown fuses is available. All digital solutions are either based on measuring the RMS current for one or more predefined time periods or estimating the wire temperature by evaluating a thermal model of the wire. In either case, these existing or proposed solutions consist either of multiple discrete components or state formulas requiring an FPU to carry out the required mathematical operations such as squaring, taking the square root or evaluating the exponential function. While these are technically suitable solutions, the hardware cost and PCB area consumption are too high since either the multitude of discrete components or the advanced MCU and its

supportive circuitry result in high part cost and require a significant amount of PCB area. The consequence is that the whole system, while providing better protective functions than meltdown or PTC fuses, is economically inferior to standard automotive meltdown fuses.

Related work to this thesis ranges from patents, describing the principles behind an electronic fuse, to publications showing the necessity due to an expected change in a car's power grid system and the benefits compared to solutions currently used in the automotive industry. The remainder of this section gives a short summary of related work found concerning this topic.

[WKM15] introduces an interesting concept combining both a conventional fuse, electronic integration circuits and an electronic switch. The idea is to have the fuse protect the wire but also make use of the integration circuits for special cases where the fuse is too slow. The patent mentions arcing as one of these scenarios. While the conventional fuse may be too slow to detect a parallel arc or may not be able to detect arcing in case of a serial short at all, evaluation of the integrator outputs can be used to detect these cases. For a parallel electrical arc this is done by detecting short high-current spikes with the integrators by integrating over a short time period and opening the switch once a threshold is passed. For a serial arc, where the load has a seemingly higher impedance and thus current flow is reduced, the problem is transferred back to the case of a parallel arc by setting a capacitor in parallel to the load. The capacitor provides power to the load during the serial arc and causes high current spikes once it reloads. These can be observed and handled as with the parallel arcing case. Drawbacks of this solution, however, are the large number of parts involved and since a conventional fuse is being used physical access to the module by the customer is necessary in order to change a blown fuse. The benefit of being able to hide the module somewhere beneficial for the car manufacturer is therefore void.

[Wor12] is similar to [WKM15] as it uses a conventional meltdown fuse as well. This time, however, the fuse chosen is not capable of protecting the wire for the whole operating range as it is chosen to be underrated. This way the fuse is well suited for low currents but would melt much earlier than required for higher currents. An integration circuit observes the current and opens a metal–oxide–semiconductor field-effect transistor (MOSFET) in parallel to the fuse if the wire is capable of handling the higher current before the fuse melts. Once the current is too high or remains for too long the electronic circuit closes the MOSFET again and the fuse blows within a short time. The disadvantages of this approach are again the number of component required and physical access due to a meltdown fuse being used.

[HT10] deals with evaluating a thermal model of a wire. The electrical resistance of the wire changes with its temperature and the evaluation used in this patent incorporates this change in resistance when calculating the wire temperature delta. The choice for the initial temperature, however, is not trivial. The wire temperature can vary depending on its location within the vehicle. Another possibility would be a residue temperature due to preceding use which is not taken into account due to a power reset, for example. In

addition, while being mathematically completely correct, the operations and formulas used are not optimized in any way and offer a large potential for optimization.

[Bad14] describes how to build an electronic fuse which triggers once a specific energy per time unit is exceeded. This is accomplished using discrete components such as resistors, capacitors and operational amplifiers. The thesis describes how to dimension these components in order to simulate the behaviour of an automotive meltdown fuse and also shows an actual prototype.

[Est98] shows the possibility of using Infineon PROFET high-side MOSFET switches as a fuse. The protection functions of the PROFET itself can be used to protect a wire. In the case of an overcurrent the device limits the current to a specified level protecting the attached wire and load from high current peaks. If the overcurrent lasts longer and is not a transient peak then the PROFET heats itself up beyond a certain threshold and switches itself off in order to protect itself from thermal destruction. This behaviour offers sufficient protection for all wires who's isothermal lines are completely above those of a specific PROFET. In addition the PROFET offers a sense pin which delivers a current relative to the current being forwarded. It is mentioned that the information on this sense pin can be used with additional circuitry, such as an MCU to provide additional protection functions. An example given is to have a fuse with a low current limit, e.g. $5\,A$, which would not trigger the internal protection functions of the PROFET.

[JB15] describes the benefits of an intelligent protection system. Additionally, it states that wire diameter reduction can also be achieved by actively preventing parallel activation of loads. Low priority loads can be selectively deactivated in order to prevent overloading a wire's capacity. With conventional meltdown fuses one has to be prepared for the worst case where all loads connected to a wire are active at once. Finally a brief description of a prototype built out of discrete components by ams AG is described. The prototype calculates the temperature or measures the current flowing through the wire and triggers its MOSFET drivers to interrupt the current flow. The article touches upon the topic of functional safety whereby it solely states that the company is capable of building a reliable product.

[FGO+14] introduces a joint project by BMW Group, HELLA KGaA Hueck & Co concerning current sense and switching modules (CSSMs). It proposes a change from a cascaded power supply network to a power bus network with attached CSSM modules. This step, according to the article, is the key to reduce complexity and the weight of the wiring harness and to improve functionality and reliability of modern vehicles. The CSSM modules have multiple channels and can be used for low and very high currents. The protection function is implemented as an over current protection by switching off channels if a specific current is exceeded for a specified time. The modules have been implemented in a BMW 7-series vehicle for trial runs.

[MJKF13] gives a short overview concerning temperature calculation using a thermal model and it's benefits. It mentions transitions to an energy backbone in the future and highlights the benefits of an electronic fuse. Tests together with Volkswagen AG showed

that a significant weight reduction of up to 50 % is possible with a decentralized power distribution and electronic fuses.

[Dru14] focuses on the requirements for and economical feasibility of a future energy backbone design. The motivation is a switch to an Ethernet communication network and corresponding EMV requirements, to lower CO_2 emissions and to deal with the fact that modern composite car bodies can't conduct electricity. For the introduced multi-voltage energy backbone the necessary higher degree of freedom for the power supply network can only be achieved with intelligent fuses and switching elements. For future energy backbone designs conventional fuses used today are therefore not sufficient.

[SFWLR14] deals with the challenges of developing electrical systems for modern cars as well. Passenger exposure to electromagnetic fields as well as issues regarding powering and connecting electrical loads are topics addressed. A central busbar design with multiple voltages is proposed as a main power backbone with „short multi-drops feeding to distributed, smart current distributors. The distributors electronically safeguard the supply lines to the consumers and may be installed in spaces that are not readily accessible to the customer due to their small dimensions and their freedom from maintenance. The semiconductor switches in the distributors are reset by the software, thus fuses and/or relays do no longer have to be replaced.“ The benefit of electronic fuses is a better wire utilization and therefore wire cross sections can be reduced, reducing the weight of the wire loom and thus reducing CO_2 emissions. The article concludes that „The present concept with busbars and smart current distributors opens up new solution possibilities for future-generation vehicles.“ highlighting the importance and necessity of electronic fuses in the future.

The two latest articles from 2016, [Dru16] and [SD16], show a prototype built by the Dräxlmaier Group involving not only the use of electronic fuses but going a step further and changing the conventional hierarchical cabling system to an energy backbone system providing both 12 V and 48 V in addition to ground. Figure 2.2 shows the backbone wiring inside the passenger cabin. Benefits from this system include weight and space reduction of the wiring, EMI reduction and all the benefits from electrical fuses.

Figure 2.2: Dräxlmaier Group prototype. Source: [SD16]

CHAPTER 3

Basic concepts

The main purpose of a fuse in a vehicle is to prevent the wires from being operated outside their operational range. In order to understand what that means, we will have a quick look at an electrical wire as used in the automotive industry. As can be seen in figure 3.1 an electrical wire consists of two parts. In the center there is a conducting element which is commonly made out of copper although there is a trend towards switching to aluminum in the automotive industry due to the cheaper price and reduced weight. Surrounding the conductor is a layer of insulation. The insulation material most widely used is polyvinyl chloride (PVC), which allows working temperatures up to 100°C although nowadays working temperatures of up to 150°C and higher are possible with thermoplastic elastomers and modern manufacturing processes.

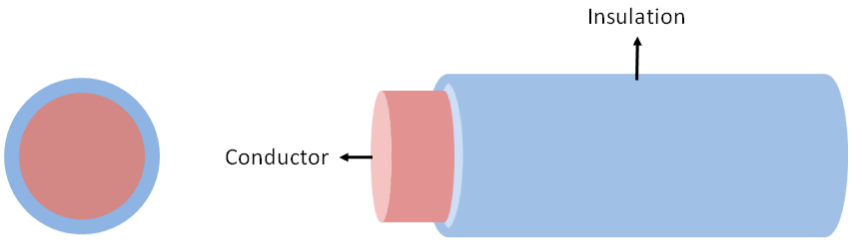


Figure 3.1: Typical wire used in the automotive area

A wire has two main operational areas, as shown in figure 3.2. In the static area it can hold a current up to a certain value infinitely long without taking any damage. In

it's dynamic area it can be used to deliver much higher currents but only for a limited amount of time. If a high current flows through the wire for too long it will take damage. A lot of loads within a vehicle can be wired using wires having a small cross section since the loads will be operated within the dynamic area of the wire. Typical loads falling into this category are for example the motors driving the door locks, windows, mirrors and seats, the horn and other loads which are only active for a short time.

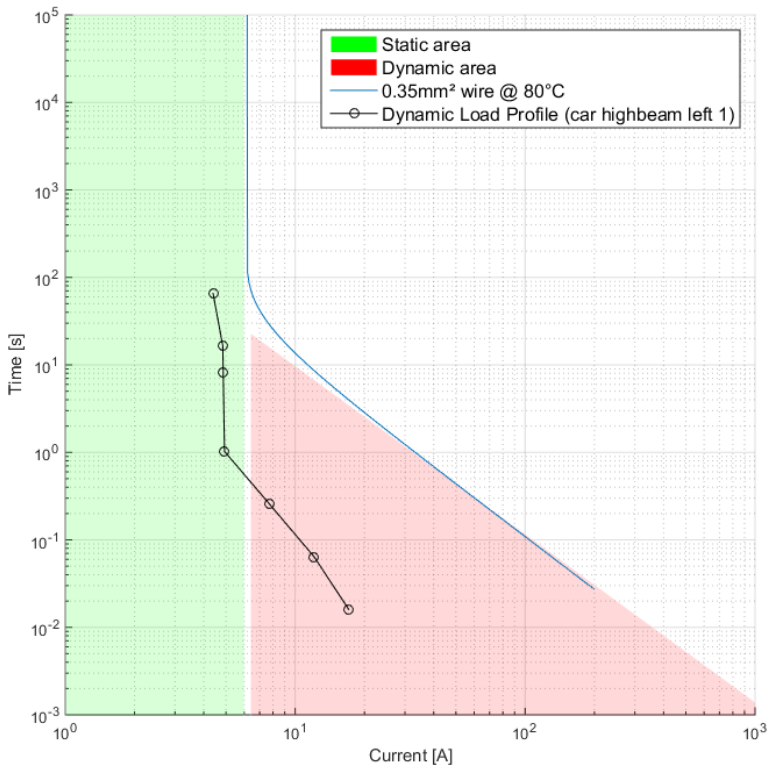


Figure 3.2: Static and dynamic usage area of a wire

In order to prevent any damage to the wire's insulation which could lead to arcing and potentially fire, the temperature of the conductor has to be kept within the working temperature range of the wire. When using meltdown fuses this is done by selecting a fuse and wire combination where the fusible link always breaks before the wire's temperature exceeds operating conditions for the entire operating range of the wire. This holds true for all loads regardless of the fact if they are driven only in the wire's static or dynamic operational area. Finding a good wire and fuse pair can thus be somewhat tricky and often a wire with a larger cross section has to be chosen to guarantee operational safety. Looking at figure 3.3 we can see that in order to use the 10 A fuse one has to increase the wire cross section from $0.75\,mm^2$ to a larger one such as $1.5\,mm^2$ to have the wire

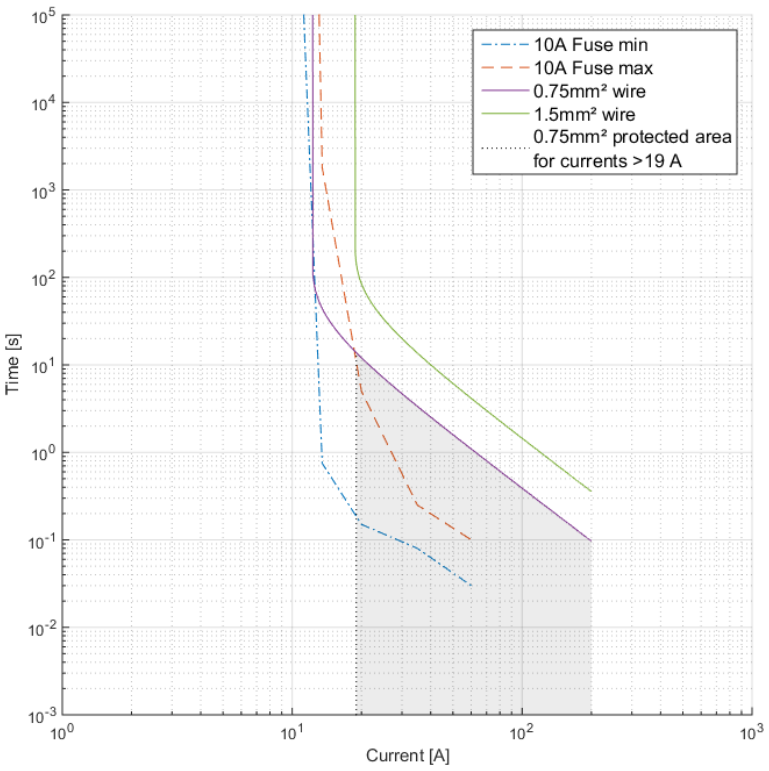


Figure 3.3: In order to be fully protected the $0.75\,mm^2$ wire has to be replaced with a larger cross section wire

protected at all times.

Using a different approach one can remove the meltdown fuse altogether and thus facilitate the design of the wire harness within the vehicle. This can be achieved by periodically evaluating the thermal model of a wire and constantly keeping track of the wire's temperature or by averaging the current over different time periods and interrupting the current either when the temperature rises too high or one of the average currents is too high, respectively.

3.1 Electrical fuse via thermal model

Instead of having a fuse which has its own thermal characteristics the idea is to build an electronic fuse with time-current characteristics matched to the characteristics of the wire to be protected. This can be done by continuously evaluating the thermal model of a given wire. The fuse is designed such that it interrupts the current flow before the wire

is used outside specification, i.e. before its temperature rises to a point where it could take damage.

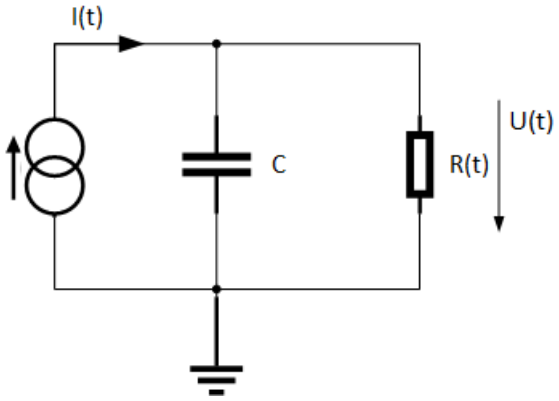


Figure 3.4: Equivalent electrical system of a wire's thermal model

An in the automotive industry accepted thermal model of a wire, as seen in figure 3.4, consists of two main components. The first is the thermal resistance R_{th}[1], R in figure 3.4, and the second is the thermal capacity C_{th}[2], labeled C in the figure. The higher a material's thermal resistance is the more it resists heat flow. Materials with low thermal resistance, such as metals, are therefore commonly used as heat sinks while materials with high thermal resistance are used for thermal insulation, such as glass and gases for multi-pane glazing in buildings. The thermal capacity is a property describing how much heat an object or material can absorb until it's temperature rises by 1°K. These two properties together with the electrical resistance of the wire and the current flowing through the wire are sufficient to evaluate the thermal model. Further details on how this can be done and what kind of optimizations are possible will follow in chapter 4.

3.2 Anti-aliasing

The protection system presented in this document will be based on periodic current samples and as a consequence the issue of aliasing has to be addressed. The Nyquist–Shannon sampling theorem states that the sampling frequency f_s of a signal has to be greater than twice the highest frequency one wants to capture. Frequencies higher than $f_s/2$ will distort the captured information, an effect called aliasing, and thus these frequencies have to be blocked out before the signal sampling stage. Figure 3.5 shows how a signal f_a close to the sampling frequency f_s can falsely be interpreted as a signal of frequency $f_s - f_a$. Having a look at the frequency domain in figure 3.6 one can see that frequencies higher than $f_s/2$ will have repeated images of themselves and cannot be distinguished from actual frequencies in the range of interest if they are not attenuated adequately.

[1]Thermal resistance is measured in Kelvin per Watt $\frac{K}{W}$
[2]Thermal capacity is measured in Joule per Kelvin $\frac{J}{K} = \frac{Ws}{K}$

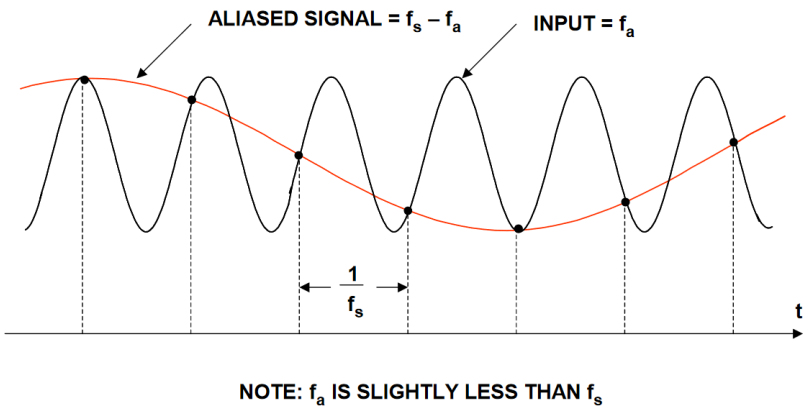


Figure 3.5: Aliasing effect in the time domain. Source: [Kes05]

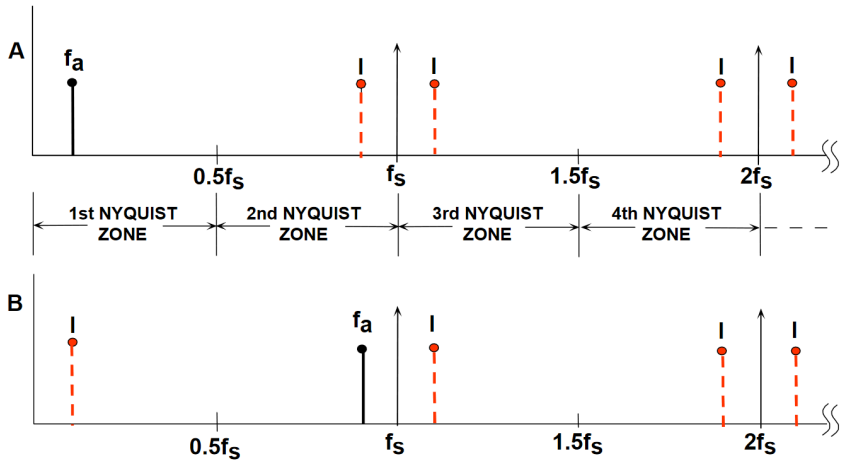


Figure 3.6: Aliasing effect in the frequency domain. Source: [Kes05]

We will have a very brief look at the idea behind three different methods to prevent aliasing. Further information can be found in corresponding literature.

- Analog RC-circuit
- Oversampling, digital filtering and decimation
- Stochastic sampling

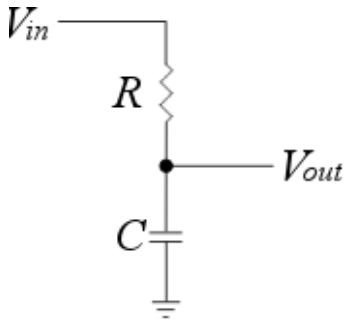


Figure 3.7: RC low-pass filter suitable as anti-aliasing mechanism

3.2.1 Analog RC-circuit

Using resistors and capacitors one can build an analog low-pass filter as depicted in figure 3.7. When the input signal is of low frequency, the capacitor has enough time to charge and the input signal can be read at the filter output. At high frequencies, the capacitor hasn't got enough time to charge and thus the signal will not be available at the output. The cutoff frequency, which is defined as the frequency where the power gain is equal to $-3dB$, is given by $f_c = \frac{1}{2\pi RC}$. At this point the power of the output signal is about half the power of the input signal. Concerning voltage amplitude it is the point where the output voltage is scaled by a factor $1/\sqrt{2}$. Frequencies higher than f_c are in the so called stop band are being attenuated.

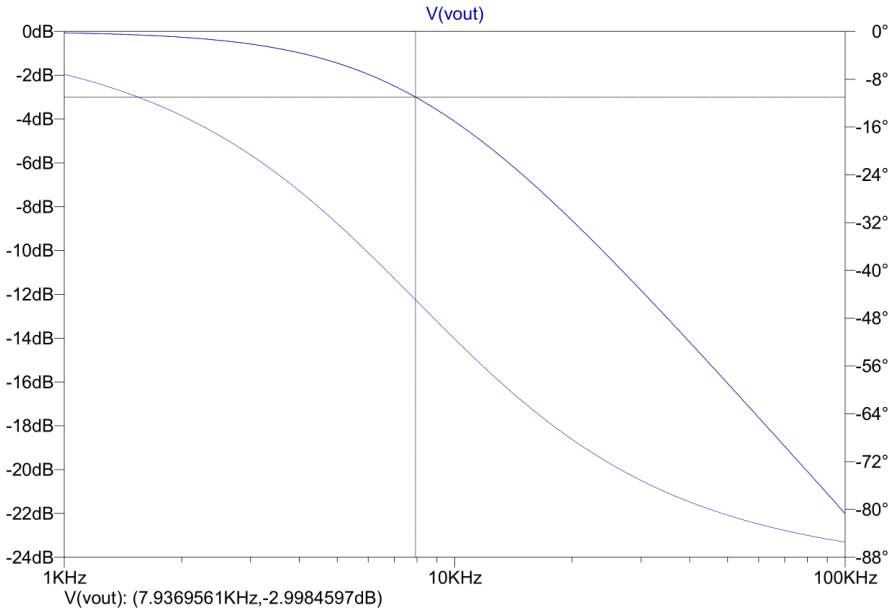


Figure 3.8: Bode plot for simple RC low-pass filter. $R = 1k\Omega$, $C = 20nF$

Figure 3.8 shows the so called Bode plot visualizing both the magnitude and the phase

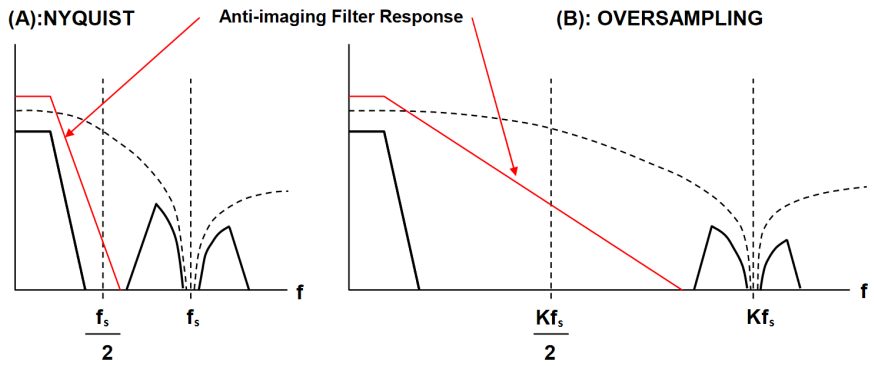


Figure 3.9: Oversampling can be utilized to use lower order analog filters. Source: [Ana09]

shift of the output signal depending on the signal frequency. For a simple RC circuit as depicted in figure 3.7 with $R = 1k\Omega$ and $C = 20nF$ the corner frequency f_c is $\frac{1}{2\pi RC} = \frac{1}{2\pi \cdot 1k \cdot 20n} \approx 7960Hz$. When using an analog RC circuit as an anti-aliasing filter, one has to make sure that the frequencies higher than half the sample frequency are sufficiently attenuated. Depending on the usage scenario this will require higher order and thus more complicated and therefore more expensive filters.

3.2.2 Oversampling, digital filtering and decimation

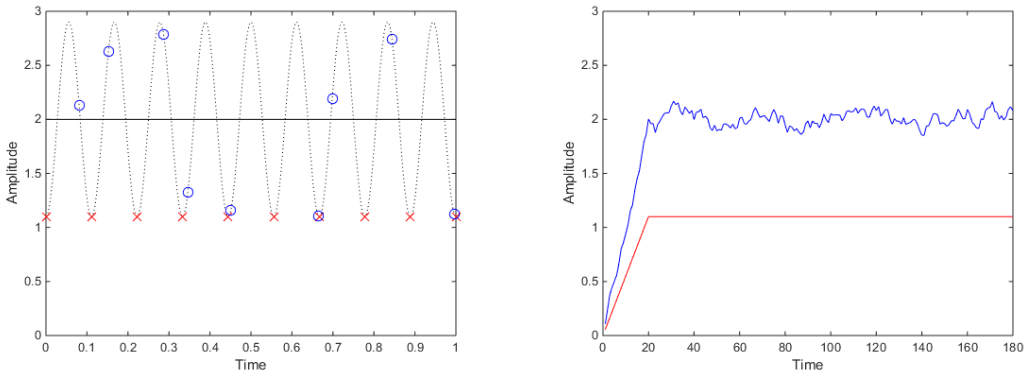
While the practice of oversampling, digital filtering and decimation cannot replace an analog anti-aliasing filter, it can be used to drastically reduce the complexity of the required analog filter. The closer the frequencies of interest are to $f_s/2$, the steeper the frequency response has to be considering the transition from pass band to stopband. This means that complex and expensive high order analog filters have to be implemented. A way to lessen the burden on the analog filter is to use oversampling. Hereby the signal is sampled with a multiple of the original sampling frequency. Once the signal is sampled it is filtered by a digital low-pass filter and then decimated, for example by choosing every K^{th} sample for an oversampling factor of K, to receive the originally intended sampling frequency.

This process enables the use of a lower order first stage analog anti-aliasing filter. Figure 3.9 shows how the conditions regarding the frequency response of the analog anti-aliasing filter are influenced by oversampling.

3.2.3 Stochastic sampling

Used frequently when dealing with images, „Stochastic sampling scatters high frequency information into broadband noise rather than generating the false patterns produced by regular sampling.“ [DW85]. Essentially the sampling points are not spaced equidistantly but based on a stochastic function.

Figure 3.10a shows a DC signal (black solid line) of amplitude 2 with a 9 Hz noise signal of amplitude 0.9. The combined signal is given by the black dotted line. Since the signal is sampled at 9 Hz as well, this scenario depicts a worst case. Upon every sampling instant, depicted by red cross markers, the sample is much lower than the signal of interest at that time. Using a stochastic sampling approach, depicted by blue circle markers, the sampling instant is varied uniformly across the sampling period. After passing the samples through a moving average filter to compensate for the introduced noise, as shown in figure 3.10b, one can see that in the case of equidistant sampling the output is of amplitude 1.1, due to the effect of aliasing. Using stochastic sampling there is no aliasing effect but the noise signal is converted into broadband noise resulting in a noisy signal. However, this signal is fluctuating around the amplitude of 2, which is the amplitude of the signal of interest. Thus with stochastic sampling a worst case scenario as with equidistant sampling is not possible.



(a) The sampling points are randomly distributed with stochastic sampling in contrast to conventional equidistant sampling

(b) Aliasing is converted into high frequency noise when using stochastic sampling

Figure 3.10

3.3 FXP calculation

The normal approach to calculate the thermal model would be to calculate it using floating-point data types. Every central processing unit (CPU) used in notebooks, personal computers and mobile phones does this with the help of the FPU. The FPU, often referred to as a math coprocessor, is a hardware unit specifically designed to execute floating-point operations. Once a CPU reaches special floating-point instructions it passes control to the FPU which executes the required operations and presents the result to the CPU. This ensures fast execution but comes at a cost. The ARM VFP9-S FPU coprocessor, for example, has a gate count of up to „130K gates“[ARM]. A solution which does not require a FPU can therefore reduce the production costs directly by not

having to include a FPU for a hardware solution or indirectly by being able to choose a MCU that does not have one.

If the full benefits of floating-point calculation are not required and one simply needs a couple of extra bits to store a fractional part, fixed-point (FXP) calculation offers a good trade off between precision and ease of computation. The idea is to have all the values scaled by a specific value and therefore gaining space for a fictional fractional part, since the computations are executed on integer numbers. To interpret the integer result correctly one has to know with what factor it has to be scaled.

In this document we will use the same $(S/I/F)$ notation system as in [Has10]. The $(S/I/F)$ notation uses S bits for the sign, I bits for the integer part and F bits for the fractional part of a corresponding FXP number, resulting in a total of $S + I + F$ bits. In our model we will be using unsigned numbers only, so all numbers will be of the kind $(0/I/F)$.

3.3.1 FXP addition and subtraction

For FXP addition and subtraction no special attention is required. These operations work as usual in the unsigned or signed domain with the only difference being that the lowest F bits of a $(S/I/F)$ number represent the fractional part. One can simply treat the result as the result of two $I + F$ bit numbers and then multiply the number by 2^{-F}. The following example adds two numbers $a = 2.125$ and $b = 1.5$ with format $(0/2/3)$:

$$\begin{gathered} a + b = c \\ 2.125 + 1.5 = 3.625 \\ 10.001 + 01.100 = 11.101 \end{gathered} \tag{3.1}$$

$$\begin{gathered} 17 + 12 = 29 \\ 10001 + 01100 = 11101 \\ 29 \cdot 2^{-3} = 3.625 \end{gathered}$$

3.3.2 FXP multiplication

With FXP multiplication there is not really a difference as well but one has to keep in mind that there is a fractional part. Assuming we square the largest possible value of a $(0/I/F)$ number, we receive the following result:

$$\left(\frac{2^{I+F} - 1}{2^F}\right)^2 = \frac{2^{2(I+F)} - 2^{I+F+1} + 1}{2^{2F}} \tag{3.2}$$

The resulting number is therefore of the format $(0/2I/2F)$. Let us have a look at an example with actual numbers. Number a is 3.5 and number b is 2.25, whereby both numbers are represented in the (0/3/2) FXP format. Our calculation looks therefore like this:

$$\begin{aligned} a \cdot b &= c \\ 3.5 \cdot 2.25 &= 7.875 \\ 011.10 \cdot 010.01 &= 000111.1110 \end{aligned} \tag{3.3}$$

As one can see in equation 3.3, the calculation is essentially the same as normal binary multiplication, except that our result c is now represented in the format (0/6/4). In general, when multiplying two FXP numbers a and b with formats $(0/I_a/F_a)$ and $(0/I_b/F_b)$ the result will have the format $(0/I_a + I_b/F_a + F_b)$.

CHAPTER 4

Design of Advanced Wire Protection

4.1 Overview

This chapter will give some details regarding the design of an actual wire protection system based on the evaluation of the wire's thermal model. The wire temperature is calculated and current flow is interrupted if it gets too high. With this concept one can use the wire up to capacity compared to solutions averaging the current.

Figure 4.1 is a conceptual figure showing the main building blocks required. As mentioned in chapter 3, the required information for calculating the wire temperature are its electrical resistance R, thermal resistance R_{th}, thermal capacity C_{th} and the amount of current flowing through the wire. Section 4.2 will go further into detail as how this information is used and also shows areas of optimization.

4.1.1 Current measurement

Of the four parameters of the wire temperature, the current flowing is the only one that has to be actually measured periodically during operation while the others can be evaluated beforehand. The first computational block is therefore one representing current measurement. Depending on the actual implementation of the fuse, using multiple components or combining as much as possible into a single ASIC, this may be done in several ways. A common approach is to use an external shunt resistor, or as proposed in related work using the wire itself as a shunt. Another possibility would be a hall sensor or the usage of other components possibly used in other building blocks. It should be noted here that the possibility of combining several blocks into a single component is given for the entire design and is solely a matter of cost and performance.

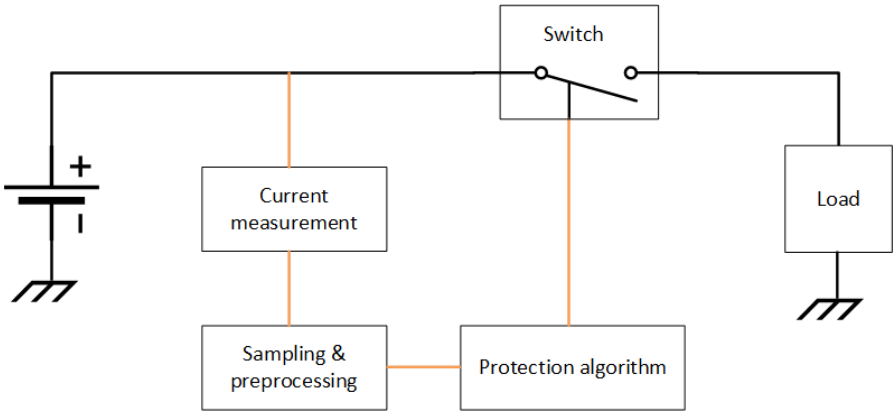


Figure 4.1: Fuse concept block diagram

4.1.2 Sampling & preprocessing

Once the current is able to be measured via a voltage differential it has to be sampled. An analog-to-digital converter (ADC) converts the voltage level into a digital representation ready for use in the following processing steps. An anti-aliasing technique has to be used to prevent signal distortion which can lead to a false temperature calculation. The algorithm evaluating the thermal model actually requires the squared current and not the current itself. This squaring can be achieved via several methods and can be performed before sampling, e.g. by using an analog multiplier. Since the main focus of this thesis lies on the efficient evaluation of the thermal model we will assume that the squared current value is available for calculation.

4.1.3 Protection algorithm

The next building block is the main one, as it is responsible for the temperature calculation by evaluating the thermal model. As shown in the upcoming section 4.4 there are different ways given on how to perform the actual computation. While the different methods of execution may not have any performance difference when executed on an MCU, for a hardware solution the different methods do have varying resource requirements. Chapter 5 will show how the four implementation options presented in section 4.4 perform as a hardware solution using an FPGA. Depending on the temperature calculated in this block the switch block is controlled.

4.1.4 Switch

As a final building block we have the switch. This will most likely be a high-power MOSFET, for example from the Infineon PROFET family. During normal operation the switch is closed and current can flow through the load. Once the temperature calculation yields a temperature exceeding a predefined threshold, a control signal is sent to the MOSFET or a MOSFET driver in order to open the switch and interrupt the current

flow. Once the wire temperature has gone down and reached a safe level the switch can be closed again in order to allow normal operation.

4.2 Simplified thermal model

In chapter 3 we showed the idea of evaluating a wire's thermal model in order to implement the protection function. The thermal model used, depicted as its equivalent electrical model, can be seen again in figure 4.2. We will first derive a formula and then validate it against similar formulas found in the literature.

4.2.1 Derivation

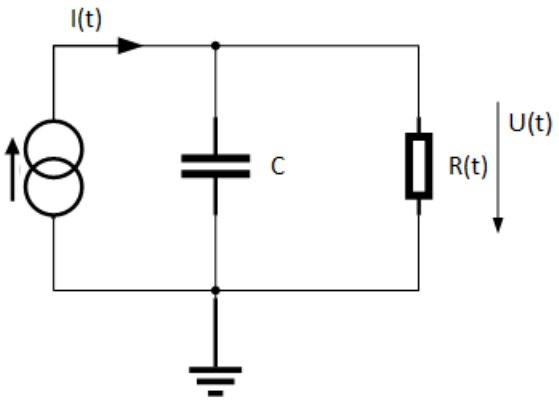


Figure 4.2: Equivalent electrical model of thermal model

The heat flow through the system corresponds to the electric power $I(t)^2R(t)$, whereby both the current and the specific electrical resistance of the wire are time dependent. The specific electrical resistance of a copper wire is actually not time dependent but temperature dependent but since the conductor temperature is a function of time, the specific resistance can also be expressed as a function of time. [Ilg04] uses an approximation for the specific resistance R with a linear temperature coefficient α and square temperature coefficient β.

$$R = R_0 \left[1 + \alpha(T - T_0) + \beta(T - T_0)^2\right]$$

Here T represents the actual conductor temperature and T_0 a reference temperature, usually taken as 20°C. However, [HT10] and others use an approximation with a linear term only since this is accurate enough. The problem with this temperature dependent resistance is that in a car the wire is routed through several areas of a car. These areas are for example the engine compartment, the cabin and along the chassis. It is obvious that the temperature in these areas can not be assumed to be the same. [Ed 04] shows that the temperatures in the engine compartment can exceed 300°C for certain parts,

whereby the temperature measured in the centre of the hood is approximately 100°C. The overall ambient temperature in the engine compartment can thus be assumed to be close to 80°C. In the passenger cabin temperatures are normally between 20°C to 30°C, while zones outside of the passenger cabin except for the engine compartment have a temperature close to the outside ambient temperature, which can even go below 0°C. As one can see, it is not easy to choose what temperature should be taken as a reference temperature in order to calculate the specific electrical resistance of the conductor.

As a consequence we will assume the worst case temperature of 80°C to ensure that the wire is protected in any scenario regardless of the point of installation. This assumption leads to a simplified formula and algorithm, since the electrical resistance is no longer time dependent but rather a constant.

Instead of evaluating the thermal model directly we can instead convert it into an equivalent electrical model and evaluate this one instead. This enables us to make use of all the sophisticated operations available in the electrical domain.

We make use of the following equivalences. Heat Q corresponds to electrical charge q, heat transfer rate $\dot{Q}$ to electrical current I and temperature T corresponds to voltage V. This gives us equivalent formulas for resistance and capacitance as shown in equation 4.1.

$$\begin{aligned} R_{th} = \frac{\Delta T}{\dot{Q}} \quad &\Leftrightarrow \quad R = \frac{U}{I} \\ C_{th} = \frac{Q}{\Delta T} \quad &\Leftrightarrow \quad C = \frac{q}{U} \end{aligned} \tag{4.1}$$

Using the Laplace transform to facilitate calculation and using the initial condition $\Delta T = 0$, since as initial condition everything is at ambient temperature, we receive equation 4.2.

$$I(s) = U(s) \cdot sC + \frac{U(s)}{R} \tag{4.2}$$

The Laplace domain is used when dealing with continuous-time systems. Since the algorithm will be executed on an MCU, FPGA or ASIC, where the time domain is discrete, we have to transform the formula. We can achieve this by using the bilinear transform. This transformation can not only be used to switch from the continuous-time Laplace domain to discrete-time Z-domain but also vice versa. All we have to do is to perform the following substitution $s = \frac{2}{T}\frac{z-1}{z+1}$ in equation 4.2. As a result we receive equation 4.3.

$$
\begin{aligned}
I(z) &= U(z)\cdot\frac{2}{T}\frac{z-1}{z+1}C + \frac{U(z)}{R} \\
I(z)\cdot R &= U(z)\cdot\frac{2RC}{T}\frac{z-1}{z+1} + U(z) \\
I(z)\cdot R(z+1) &= U(z)\cdot(z-1)\frac{2RC}{T} + U(z)\cdot(z+1) \\
I(z)\cdot R(1+z^{-1}) &= U(z)\cdot(1-z^{-1})\frac{2RC}{T} + U(z)\cdot(1+z^{-1})
\end{aligned}
\tag{4.3}
$$

Now that we've got our equation in the discrete-time Z-domain all that remains is to convert it back into the regular discrete-time domain. Using the inverse Z-transform on equation 4.3 we can receive the desired result.

$$
\begin{aligned}
R(I[n] + I[n-1]) &= \frac{2RC}{T}(U[n] - U[n-1]) + U[n] + U[n-1] \\
R(I[n] + I[n-1]) &= U[n]\left(\frac{2RC}{T} + 1\right) - U[n-1]\left(\frac{2RC}{T} - 1\right) \\
U[n] &= \frac{R(I[n] + I[n-1])}{\frac{2RC}{T} + 1} - U[n-1]\frac{\frac{2RC}{T} - 1}{\frac{2RC}{T} + 1}
\end{aligned}
\tag{4.4}
$$

As we can see in the transformed formula, for any $I[n]$ it is true that it is used twice. Once in the n^{th} iteration and once in the $(n+1)^{th}$. In general there will be no large difference between two consecutive current samples, due to the high sampling frequency, and thus the electrical power, in the formula $I[n]$, will not change significantly between two iterations. The wire temperature $U[n]$ also won't change significantly from one iteration to another, since the time constant of the wire model, see equation 4.6, is much larger than the iteration period. As a result, one can simplify equation 4.4 by using $2I[n]$ instead of $I[n] + I[n-1]$ without introducing any significant error.

$$
\begin{aligned}
U[n] &= R\cdot I[n]\frac{2}{\frac{2RC}{T} + 1} + U[n-1]\frac{\frac{2RC}{T} - 1}{\frac{2RC}{T} + 1} \\
U[n] &= R\cdot I[n]\alpha + U[n-1](1-\alpha) \qquad , \ \alpha = \frac{2}{\frac{2RC}{T} + 1}
\end{aligned}
\tag{4.5}
$$

4.2.2 Validation

Now that we derived an optimized formula we can validate it against other formulas concerning temperature estimation. In order to compare equation 4.5 with equation 4.8

listed in [Bad14, Est98, HT10] we have to compare the coefficients of $U[n-1]$ and $R{\cdot}I[n]$ to an exponential term.

$$\begin{aligned} e^{-\frac{T}{\tau}} &= \frac{\frac{2RC}{T}-1}{\frac{2RC}{T}+1} = \frac{1-\frac{T}{2RC}}{1+\frac{T}{2RC}} \\ -\frac{T}{\tau} &= log\left(\frac{1-\frac{T}{2RC}}{1+\frac{T}{2RC}}\right) \\ &= log\left(1-\frac{T}{2RC}\right) - log\left(1+\frac{T}{2RC}\right) \approx -\frac{T}{2RC} - \frac{T}{2RC} = -\frac{T}{RC} \\ \tau &\approx RC \end{aligned} \tag{4.6}$$

Equation 4.6 makes use of the approximation $log(1+x) \approx x$ for $|x| \ll 1$ resulting from the Taylor series expansion, in particular the Mercator series, of the natural logarithm, given by equation 4.7.

$$log(1+x) = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n} x^n \qquad , -1 < x \leq 1 \tag{4.7}$$

Since $\frac{2}{\frac{2RC}{T}+1} = 1 - \frac{\frac{2RC}{T}-1}{\frac{2RC}{T}+1}$ and in our equivalent electrical model for our thermal network $U[n]$ is equal to $T[n]$, equation 4.5 is equivalent to equation 4.8, which corresponds more to the ones generally found in the literature.

$$T_{wire}[n] = T_{wire}[n-1]{\cdot}e^{-\frac{\Delta t}{\tau}} + R_{th}{\cdot}R_{el}{\cdot}I_{el}[n]^2{\cdot}\left(1-e^{-\frac{\Delta t}{\tau}}\right) \tag{4.8}$$

The parameter T in equation 4.5 and Δt in equation 4.8, respectively, correspond to the sampling period. This parameter cannot be chosen at will but has to be chosen such that all relevant frequencies of the continuous time system are included. If this is not possible, one should chose a sampling period resulting in the error between the discrete time and continuous time system being sufficiently small.

The thermal models presented just now work perfectly fine in a system with sufficient calculation resources such as a personal computer (PC) or a MCU with FPU, however, for a hardware solution or MCU solution without FPU the calculation has to be optimized. Section 4.3 will cover the optimizations that we will incorporate. Section 4.5 will deal with a different concept of designing the fuse current-time characteristics by using a load dependent implementation which can also be used to detect or prevent arcing.

4.3 Formula optimization

The first step towards the target solution, optimized for simple CPUs or a hardware solution, is to do the calculations in FXP arithmetic instead of floating point arithmetic.

This tremendously decreases the amount of logic for a hardware solution and in case of a software solution it can avoid the necessity of an FPU.

There are, however, a couple more optimizations which can be utilized, as shown in the following subsections.

4.3.1 Limit scaling

Our current formula is now as given in equation 4.8. We will use the substitution $\alpha = 1 - e^{\frac{\Delta t}{\tau}}$ similar to the one in equation 4.5 and receive equation 4.9.

$$T_{wire}[n] = T_{wire}[n-1]\cdot(1-\alpha) + R_{th}\cdot R_{el}\cdot I_{el}[n]^2\cdot\alpha \tag{4.9}$$

With a first order recursion of the form $y[n] = \beta\cdot y[n-1] + \alpha\cdot x[n]$ one can omit the multiplication with α entirely, if it is constant. In order to prove this, we first do an unrolling of the recursion. Equation 4.10 shows the first step.

$$\begin{aligned} y[1] &= \alpha\cdot x[1] + \beta\cdot y[0] \\ y[2] &= \alpha\cdot x[2] + \beta\cdot y[1] = \alpha\cdot x[2] + \beta\cdot(\alpha\cdot x[1] + \beta\cdot y[0]) \\ &= \alpha\cdot x[2] + \alpha\cdot\beta\cdot x[1] + \beta^2\cdot y[0] \\ y[n] &= \beta^n\cdot y[0] + \sum_{i=1}^{n} \alpha\cdot\beta^{n-i}\cdot x[i] \end{aligned} \tag{4.10}$$

Setting $y[0] = 0$ in equation 4.10, which corresponds to the default initial condition for any recursion, we receive equation 4.11.

$$\begin{aligned} y[n] &= \sum_{i=1}^{n} \alpha\cdot\beta^{n-i}\cdot x[i] \\ y[n] &= \alpha\sum_{i=1}^{n} \beta^{n-i}\cdot x[i] \\ \tilde{y}[n] &= \frac{y[n]}{\alpha} = \sum_{i=1}^{n} \beta^{n-i}\cdot x[i] \end{aligned} \tag{4.11}$$

As can be seen in equation 4.11, we can save the multiplication by α and instead calculate $\tilde{y}[n]$, which is equal to $\frac{y[n]}{\alpha}$, instead of $y[n]$. The result of our calculation will be correct but in order to use the original intended value we either have to multiply the result

by α or keep this factor in mind for future operations. In the case that we only need to compare our result to thresholds or limits, one can simply defer the mathematical calculation and scale the limits. If, for example, we want to check that $y[n] < L$, we can simply check $\tilde{y}[n] = \frac{y[n]}{\alpha} < \frac{L}{\alpha}$. Thus we can simplify equation 4.9 by omitting the multiplication of $I[n]$ with α, R_{th} and R_{el}, which we use as constants, and instead scale all our thresholds and limits we compare the wire temperature to accordingly. For the rest of this document T_{wire} will actually stand for $\tilde{T}_{wire}$, i.e. $\frac{T_{wire}}{\alpha \cdot R_{th} \cdot R_{el}}$.

$$T_{wire}[n] = T_{wire}[n-1] \cdot (1-\alpha) + I_{el}[n]^2 \tag{4.12}$$

4.3.2 Optimizing by approximation

In this section we will see why we used the substitution $\alpha = 1 - e^{-\frac{\Delta t}{\tau}}$. For the values we are using, with small Δt and large τ, $e^{-\frac{\Delta t}{\tau}}$ is a number close to but smaller than 1. Therefore $1 - \alpha$ is close to 1. If we want to calculate the wire temperature in hardware, a multiplication with a number close to 1 is not ideal when considering resource consumption. It would be much better if we could instead multiply by one and subtract a small fraction to achieve the same solution. If the small fraction can be obtained with sufficient accuracy by multiplying with 2^{-n} then the multiplication is simply a bit shift to the right n times. A multiplication can thus be avoided and replaced with a bit shift and subtraction. We therefore expand our equation 4.12 to receive our final optimized equation 4.13, which can be implemented as an infinite impulse response (IIR) filter as represented in figure 4.3.

$$T_{wire}[n] = T_{wire}[n-1] - \alpha \cdot T_{wire}[n-1] + I_{el}[n]^2 \tag{4.13}$$

4.4 Implementation options

Four different methods for an efficient implementation of equation 4.13 are examined. The first two, sections 4.4.1 and 4.4.2 merely require addition, subtraction and bit shift whereas the latter two make use of FXP multiplication. For a MCU all four methods can be easily implemented even without a dedicated multiplication operation within the CPU. For a hardware solution, however, the different methods have different resource requirements, which, depending on the actual hardware solution, might cause one specific method to be preferred. Chapter 5 will cover the resource consumption of the different methods presented in the following sections.

4.4.1 Denominator approximation

For this method α is approximated by a fraction with numerator 1 and denominator of an integer multiple of a power of 2, i.e. $\alpha \approx \frac{1}{k \cdot 2^N}$. Using simple enumeration for n and k using a table a suitable pair is chosen so that the error is within $\pm 3\%$. Looking back at equation 4.13 our current formula is equation 4.14.

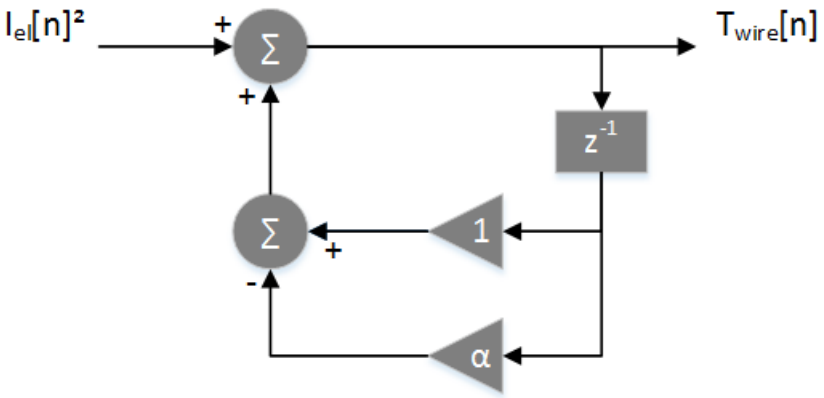


Figure 4.3: Direct form 1 representation of temperature calculation

$$T_{wire}[n] = T_{wire}[n-1] - \frac{1}{k \cdot 2^N} \cdot T_{wire}[n-1] + I_{el}[n]^2 \tag{4.14}$$

The division by 2^N can be efficiently implemented as a bit shift to the right N times but the division by k has to be treated separately. The values for k tend to be quite small and are uneven numbers, since for an even number one could simply use the pair $\frac{k}{2}$ and $N+1$. Instead of trying to implement the division by k efficiently, the division can be transformed into a different problem. The idea behind this is as follows. The division by $k2^N$ is done in order to subtract a tiny fraction of $T_{wire}[n-1]$ from itself in order to achieve a multiplication of $T_{wire}[n-1]$ with a number close to one. Instead of subtracting a tiny amount of $T_{wire}[n-1]$ in every iteration, however, a larger amount is subtracted every k^{th} iteration and otherwise nothing is subtracted. This process introduces a ripple on the output but since the wire temperature does not change significantly within k iterations and the omitted subtracted term is very small no significant error is introduced. Section 6.3 will show the introduced error. Thus, rather than dividing $T_{wire}[n-1]$ every iteration by $k2^N$, it is divided every k^{th} iteration by 2^N and set to zero in other iterations. The resulting formula is shown in figure 4.4 and is visualized by running through the branches of the graph with different frequencies F_1 and F_2 whereby $F_1 = kF_2$.

4.4.2 Numerator approximation

For this method, we use the approximation $\alpha \approx \frac{k}{2^N}$, as shown in equation 4.15. As with the method described in section 4.4.1 the division by 2^N can be implemented as a bit shift. For the multiplication with k we want to prevent the execution of an actual multiplication operation. Using a two dimensional enumeration again, appropriate values for k and N can be found. As within the previous section, the values for k tend to be quite small, i.e. smaller than 16, for the wires examined with cross sections below $1\,mm^2$. The multiplication complexity can simply be reduced by summing up $T_{wire}[n-1]$ k times

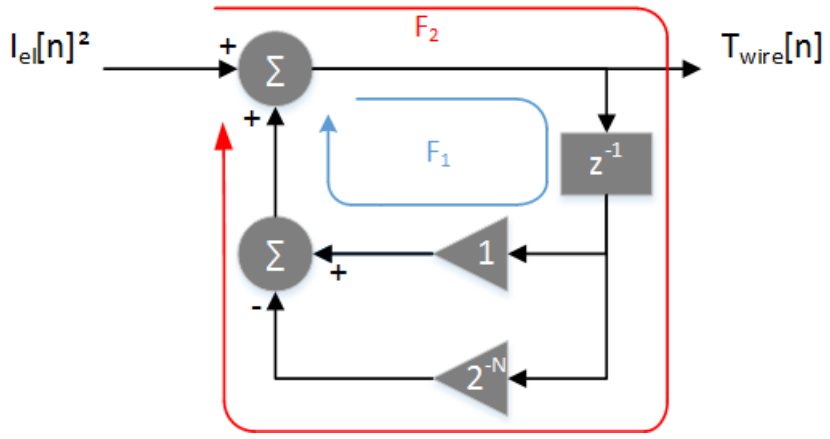


Figure 4.4: Direct form 1 representation of denominator approximation method

within a loop. In order to maintain the iteration frequency, however, the system clock has to be increased by at least a factor of k.

$$T_{wire}[n] = T_{wire}[n-1] - \frac{k}{2^N} \cdot T_{wire}[n-1] + I_{el}[n]^2 \tag{4.15}$$

4.4.3 FXP numerator approximation

This method uses the same approximation as the one in section 4.4.2, namely $\alpha \approx \frac{k}{2^N}$, except that we are going to use an actual multiplication operation for the multiplication with k. This gets rid of the loop overhead, does not require an increased iteration frequency and also no additional variables. For a MCU solution without multiplication operation, however, this will be the same method as in section 4.4.2 if no optimized multiplication algorithm is used.

4.4.4 FXP multiplication

This final method makes use of FXP multiplication to calculate the term $\beta \cdot T_{wire}[n-1]$, whereby $\beta = 1 - \alpha$, see equation 4.16. This is the most intuitive and direct implementation option but may not be the best option in terms of resource consumption in case of a hardware solution. On the other hand, while the multiplication will consume more resources than the multiplication in section 4.4.3 we do not have to execute the subtraction and bit shift. Figure 4.5 shows the modified IIR structure.

$$T_{wire}[n] = \beta \cdot T_{wire}[n-1] + I_{el}[n]^2 \tag{4.16}$$

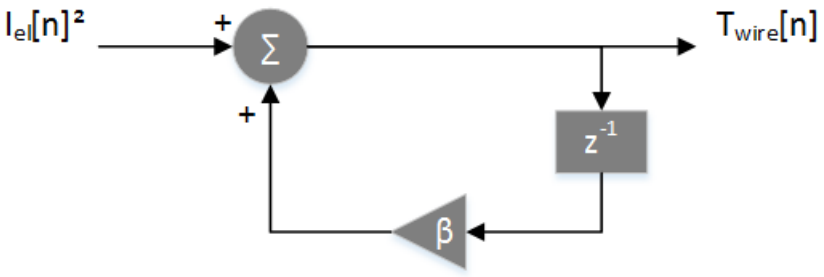


Figure 4.5: Direct form 1 representation of FXP multiplication method

4.5 Alternate trip line design method

Using multiple equations such as equation 4.17 one can implement multiple root mean square (RMS) current measurements for different time windows. The coefficient α defines the averaging period. The equation corresponds to an exponential averaging filter to measure the RMS current for a specific period, as in [Ana03], but omitting the final square root operation. The implementation of this method is very similar to the temperature calculation but requires more calculation due to the multiple filter instances. Therefore it is not covered thoroughly but is still mentioned as a possible solution.

$$I_{ms}[n] = I_{ms}[n-1] - \alpha \cdot I_{ms}[n-1] + I_{el}[n]^2 \tag{4.17}$$

Using multiple instances of this equation which can be implemented as IIR filters, one can design the current-time characteristics visually in the double-logarithmic plot. The goal is to have certain reference points in the double-logarithmic isothermal plot which are below the wire's curve. These points correspond to the maximally allowed time for a specific current to flow through the wire. The digital RMS filters are then configured in such a way that the filter characteristics for the specified current lies entirely between the wire curve or until a different filter curve is reached, which would open within a shorter time, and the load profile of a specific load. This is a similar approach as in [Wor12] but in this case the averaging process is done with a digital circuit. If one would choose to use a MCU to implement the averaging function, the number of filters and therefore the degree of freedom to design the current-time characteristics is only limited by the available memory, thus providing a highly customizable protection system.

Figure 4.6 shows the load profile of a Xenon high beam together with the isothermal of a 0.35 mm^2 wire and the current-time characteristics of a composite multi-filter digital fuse. The green boxes depict the filter window size and their current limits. With this approach, one can implement a primitive arc detection functionality, as apposed to a separate solution, such as [EMW+14, KY10].

In case of a parallel arc, as shown in figure 4.7a, the electrical resistance is lower than the actual load resistance and thus a higher current flow occurs. Since the current-time

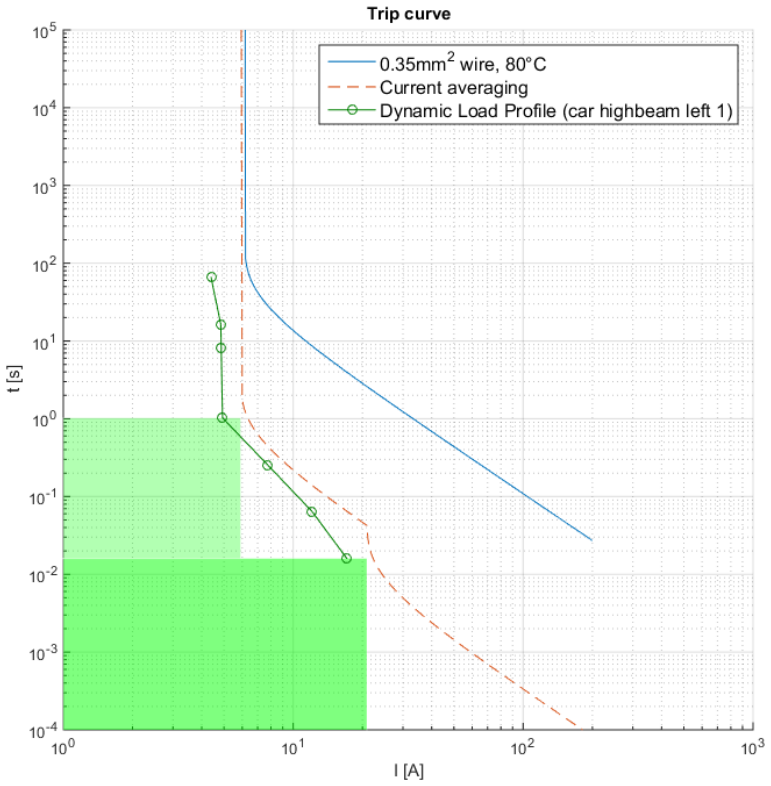
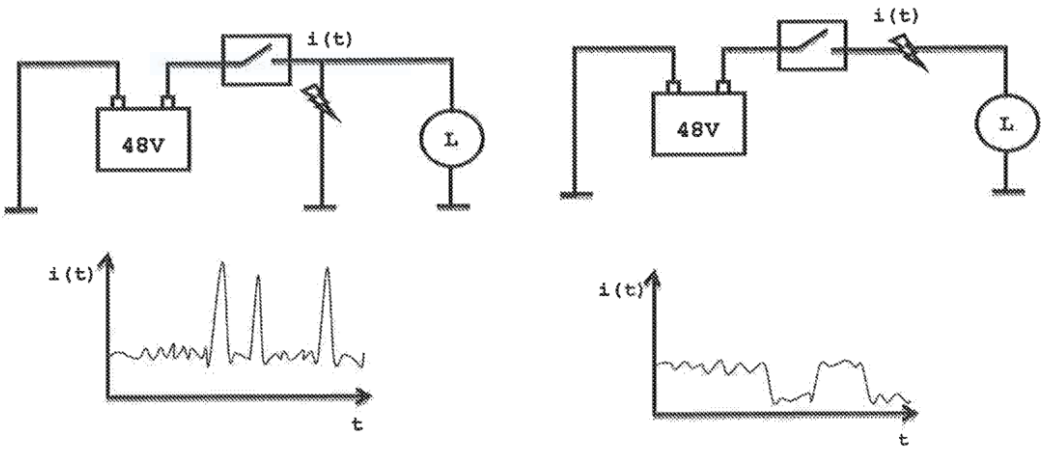


Figure 4.6: Current averaging based on dynamic load profile.
The current is averaged over two time periods with different thresholds in order to receive a fuse with time-current characteristics close to the load profile.

characteristics in figure 4.6 is specifically designed to be close to the load profile, a parallel arc can be detected if it occurs often enough or long enough for the load's dynamic profile to deviate from the reference load profile sufficiently to trip the fuse.

In the case of serial arcs the resistance is higher than the expected load resistance, as shown in figure 4.7b. In this case, on can use a second set of filters but treat the output differently. This means that if the resistance is too high for too long, deviating from the dynamic load profile, the average current will drop below a certain level. This can be detected by simply checking if the filter output is lower than a specific limit. In this case, however, the filter must only be allowed to process samples if the load is turned on, since the average current will obviously drop when the load is switched off and there is no current flow.



(a) Prallel arc whereby the load resistance drops and high current spikes occur. Source: [EMW+14]

(b) Serial arcs tend to increase the load resistance and therefore the current flow is reduced. Source: [EMW+14]

Figure 4.7

CHAPTER 5

Implementation

This chapter covers the implementation of the different design options introduced in the previous chapter. It covers both a software implementation using an MCU and a hardware implementation on an FPGA.

5.1 Software solution

Before being able to write the code for the fuse the specific hardware components had to be chosen. All the building blocks listed in section 4.1 had to be included, whereby one component can potentially contain multiple blocks. For the software solution, the following components were chosen.

5.1.1 Components

MCU As a computing component an Infineon XMC1100 MCU was selected. This MCU contains a lot of useful peripheral features such as timers, an ADC and multiple universal asynchronous receiver/transmitters (UARTs). With the CPU running at $32\,MHz$ and $64\,kB$ of built-in Flash memory it has more than enough processing power and memory for this application. In addition there is already a development kit with a remarkably small footprint of 14.0 by $38.5\,mm$, which was chosen to create a small prototype. This XMC2Go development kit and its block diagram can be seen in figure 5.1 and figure 5.2. The XMC1100 covers the *sampling & preprocessing* building block by using an external analog anti-aliasing filter together with the built-in ADC and some additional C code for preprocessing as well as the *protection algorithm* building block.

Switch An Infineon MOSFET from the PROFET family with current sense pin takes over two major roles. Not only is it an automotive grade high-side switch[1] but it also offers built-in current sense diagnostics making it an ideal choice for this application. The MOSFET itself offers a sense current via an output pin that is proportional to the current flowing through the power transistor. In addition it offers useful features such as short circuit and open load diagnostics. Figure 5.3 shows the block diagram of a PROFET switch. Looking back at the building blocks the PROFET incorporates the *current measurement* and *switch* blocks.

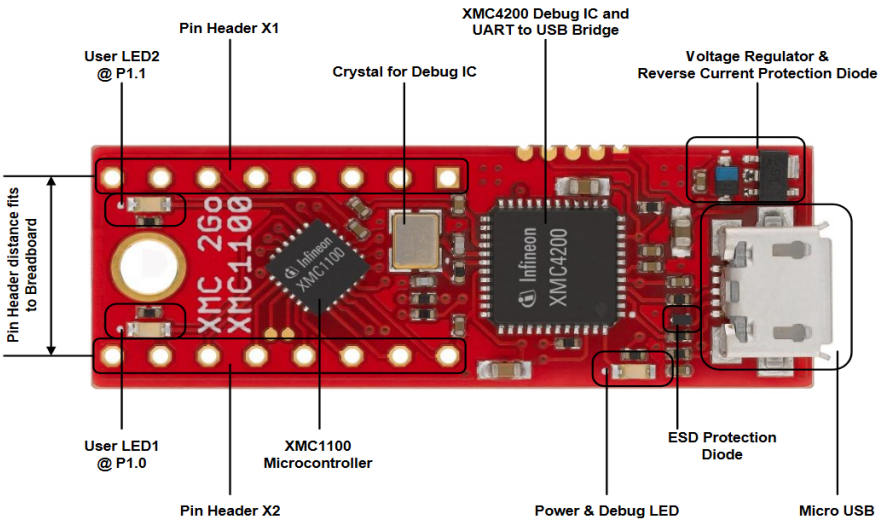


Figure 5.1: XMC2Go development kit. Source: XMC2Go Datasheet

Using these two components a prototype was built with the same dimensions as the XMC2Go board. In order to achieve this the PROFET high side switch together with two connectors used to connect a power supply and a load were mounted on a custom printed circuit board (PCB) designed to be placed upon the XMC2Go board. The finished solution can be seen in figure 5.4.

5.1.2 Algorithm implementation

Recalling the results of the previous chapter, the main equation to be continuously evaluated reads as follows:

$$T_{wire}[n] = T_{wire}[n-1]\cdot(1-\alpha) + I_{el}[n]^2 \qquad , \ \alpha = \frac{2}{\frac{2RC}{T}+1} \tag{5.1}$$

[1] A high-side switch is used to switch the supply voltage whilst sharing a common ground level.

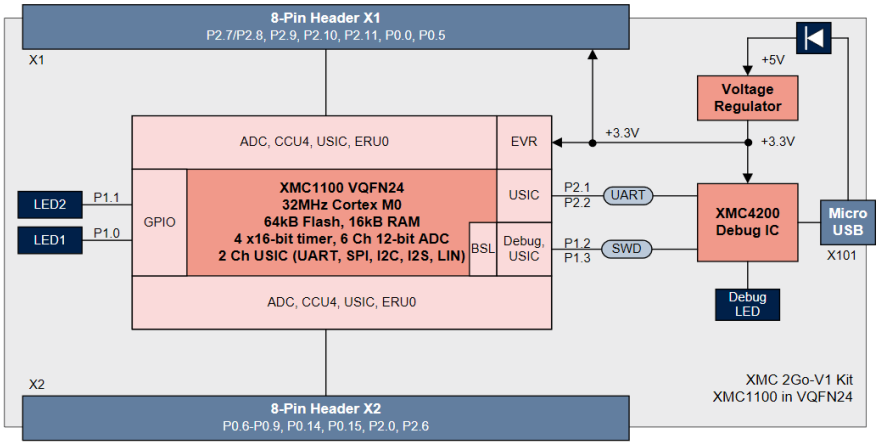


Figure 5.2: XMC2Go block diagram. Source: XMC2Go Datasheet

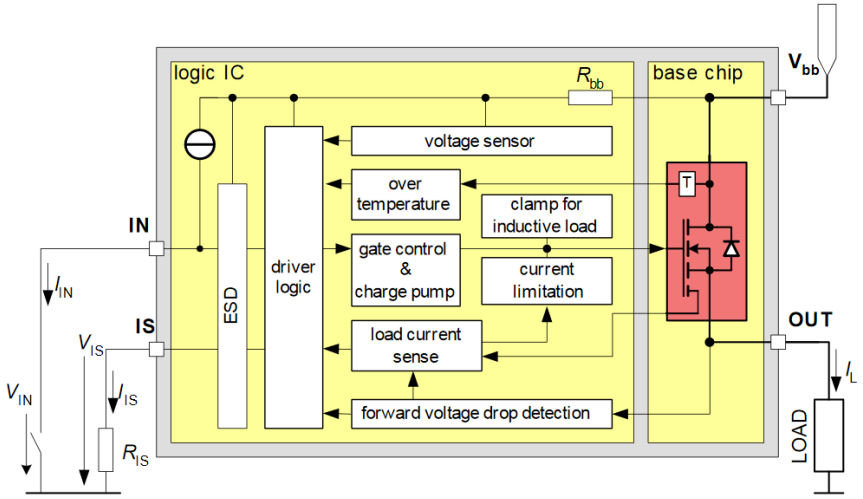


Figure 5.3: PROFET block diagram. Source: PROFET Datasheet

While R and C are values depending on the wire to be protected, the parameter T has to be chosen. T corresponds to the sampling period and the time between two iterations of the algorithm. It has to be small enough to ensure a fast response in case of an overload and to cover all the relevant frequencies. On the other hand it has to be large enough such that the calculations of two successive iterations do not overlap. A suitable value was found to be $1\,ms$ resulting in an execution frequency of $1\,kHz$, respectively.

Figure 5.5 shows how the demonstrator is connected to a load. The execution flow is as follows. A timer is used to capture samples of the sense current via a sense resistor periodically with the ADC. From this measured voltage level one can obtain the current

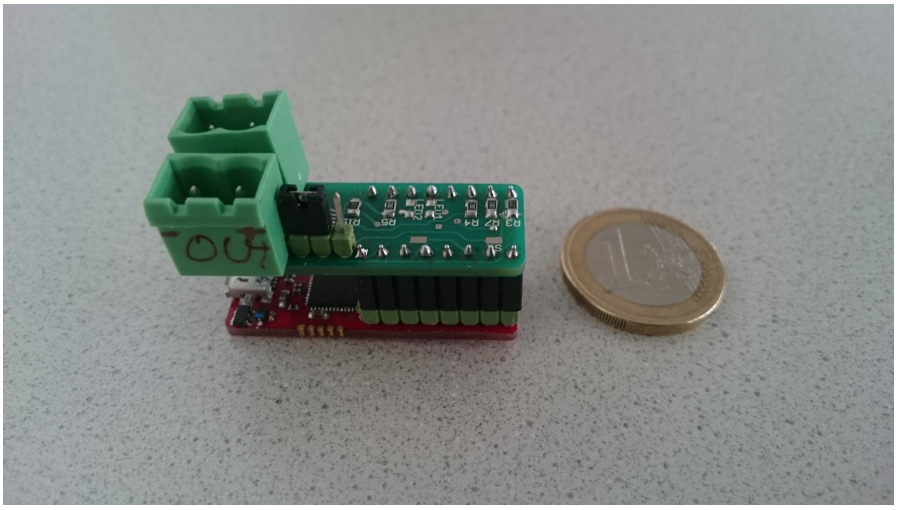


Figure 5.4: Software implementation prototype using an XMC1100 on an XMC2Go development kit and a PROFET high side switch. A one Euro coin is placed next to it for scale

flowing through the load as shown in equation 5.2.

$$
\begin{aligned}
k_{ILIS} &= \frac{I_L}{I_S} \\
U_S &= R_S \cdot I_S \\
B_S &= \frac{U_S \cdot 2^N}{U_{max}} \\
B_L &= \frac{B_S \cdot k_{ILIS} \cdot U_{max} \cdot scale}{R_S \cdot 2^N} = \frac{U_S \cdot 2^N \cdot k_{ILIS} \cdot U_{max} \cdot scale}{U_{max} \cdot R_S \cdot 2^N} \\
&= \frac{U_S \cdot k_{ILIS} \cdot scale}{R_S} = \frac{R_S \cdot I_S \cdot I_L \cdot scale}{R_S \cdot I_S} \\
B_L &= I_L \cdot scale
\end{aligned}
\tag{5.2}
$$

The factor k_{ILIS} is given in the datasheet of the PROFET as a conversion ratio between the sense current I_S and the load current I_L. U_S is the voltage across the sense resistor when the sense current I_S is flowing. When the ADC samples the voltage U_S it converts it into an N bit number B_S. One bit represents a voltage level of $U_{max}/2^N\,V$ whereby U_{max} is the maximal allowed voltage for an input signal to the ADC. B_L gives the load current whereby to avoid a value of 1 to be corresponding to $1\,A$, a custom *scale* factor

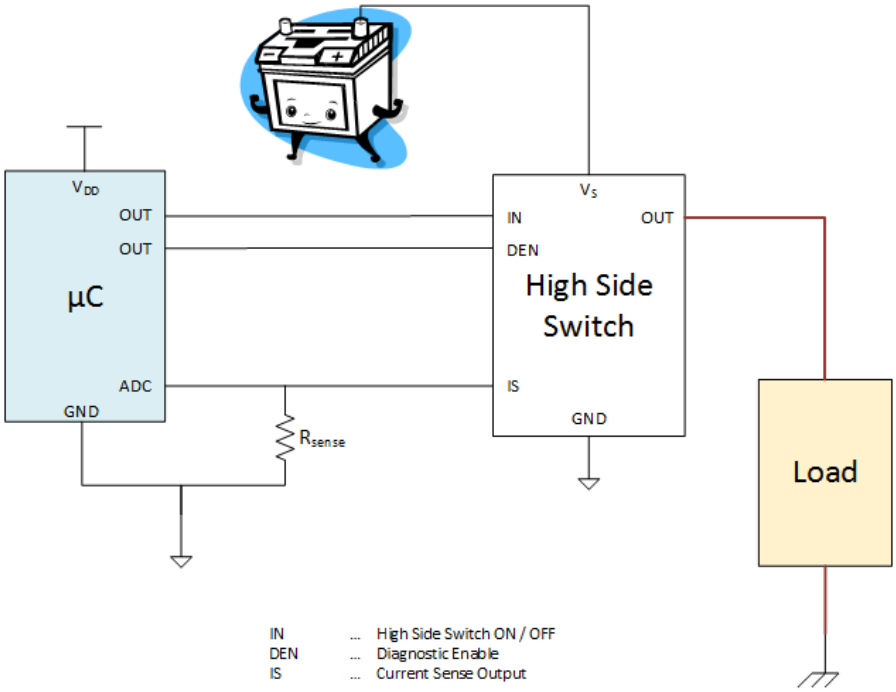


Figure 5.5: Demonstrator: The thermal model is continuously evaluated on the MCU and disconnects the load by turning off the high side switch if the temperature is too high

is used to gain higher resolution. Choosing $scale = 1000$, for example, results in B_L representing the load current as the amount of milliampere.

Whenever a new sample is available, this B_L is calculated according to equation 5.2 and used to calculate the wire temperature delta using the different calculation methods introduced in chapter 4 using FXP arithmetic. In addition, the temperature is also calculated using a floating-point data type for comparison. If the temperature delta exceeds a predefined threshold a control signal is sent to the PROFET and the load is disconnected from the power supply. Figure 5.6 is a flowchart showing the main steps of the protection algorithm.

For testing and diagnostic purposes a UART connection was also implemented together with a LabVIEW graphical user interface (GUI). The evaluation will follow in chapter 6.

5.2 Hardware solution

In contrast to the software implementation where a fully functional demonstrator was built the hardware implementation focused only on the *protection algorithm* building block. The reason is as follows: if the software implementation proves to be functional then the development of a hybrid or hardware only implementation is possible as well and merely an engineering effort. Since the different design options differ solely in

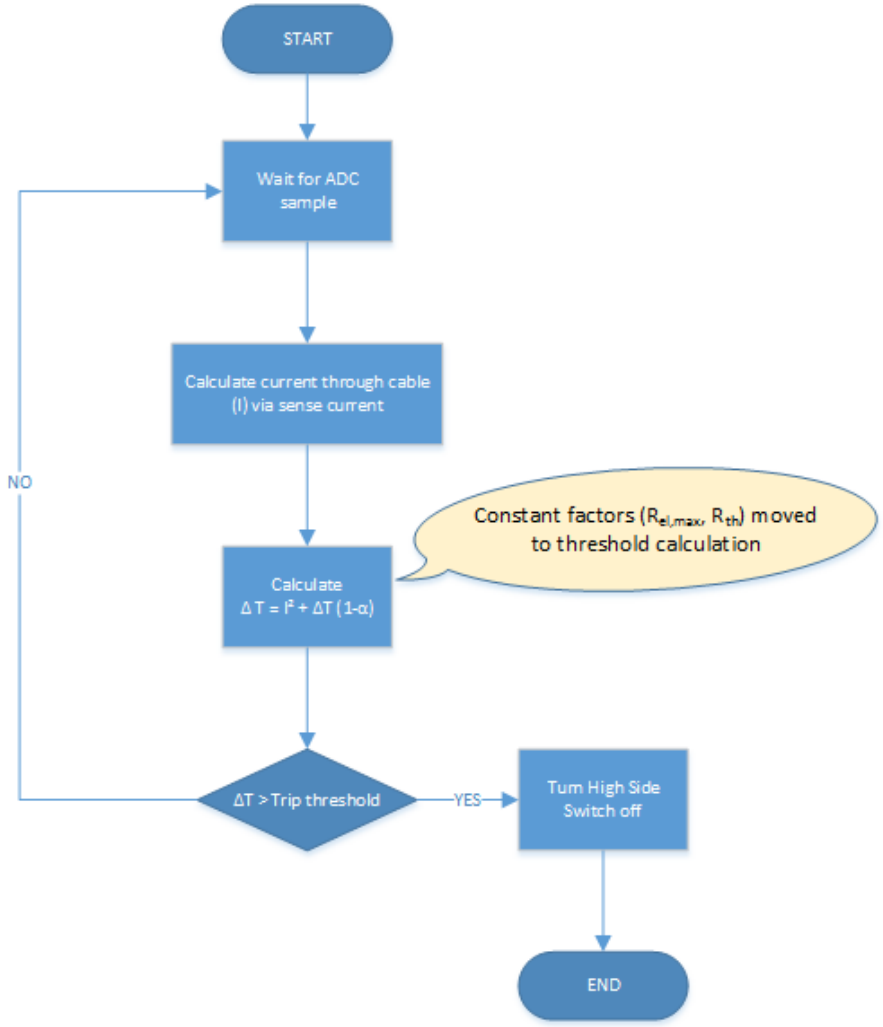


Figure 5.6: Flowchart of the implemented algorithm on the MCU

the *protection algorithm* block and resource consumption is the main criterion when developing hardware, it is sufficient to compare the resource requirements of that block. We will first have a look at the hardware technology used and then see how the *protection algorithm* block was implemented.

5.2.1 Components

For each design option presented in chapter 4 a hardware module was developed using VHDL. In order to measure the resource consumption, which is covered in chapter 6, a target technology or platform was required. For the sake of simplicity and easy to use toolchain the designs were synthesized for an FPGA. The target platform was a

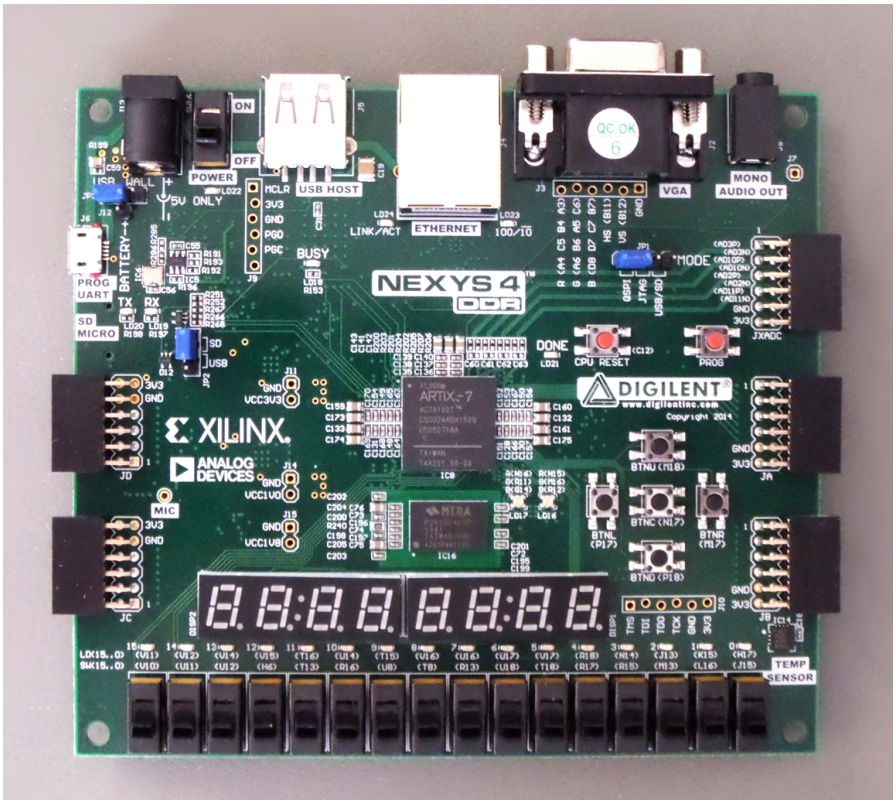


Figure 5.7: Nexys 4-DDR FPGA board used for the hardware implementation

Xilinx Artix-7 FPGA[2] by using a Digilent Nexys4-DDR development board as shown in figure 5.7. From the four processing blocks only the *protection algorithm* block is written in synthesizable code for the FPGA while the other blocks are handled by a test-bench.

5.2.2 Algorithm implementation

The program flow for the hardware solution is identical to that of the software solution shown in figure 5.6. In contrast to a software solution with a MCU executing instructions in sequence, a hardware solution is divided into modules which all operate in parallel. Program flow thus has to be enforced by connecting input and output ports of the specific hardware modules.

Figure 5.8 shows a block diagram of the hardware solution. Synthesizable VHDL modules were only written for the *protection algorithm* blocks. A non-synthesizable VHDL test-bench takes over the part of the remaining blocks by reading current values from a logfile which are used as an input for the different implementation options and also calculating a reference temperature using a floating-point data type.

[2]Xilinx XC7A100T-1CSG324C FPGA

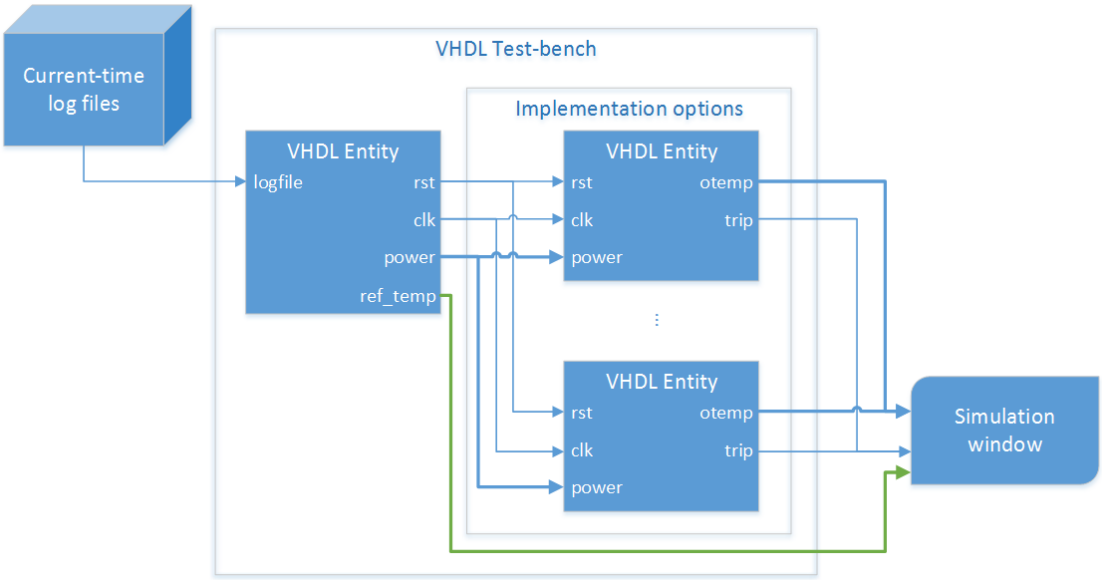


Figure 5.8: Block diagram of the hardware environment

Listing 5.1 VHDL entity of the processing blocks

```
entity protection_block is
  port (
    rst    : in  std_ulogic;
    clk    : in  std_ulogic;
    power  : in  unsigned(pow_bits - 1 downto 0);
    otemp  : out unsigned(temp_bits - 1 downto 0);
    trip   : out std_ulogic
  );
end protection_block;
```

The entity of the VHDL modules is shown in listing 5.1. The ports *rst* and *clk* are the reset and clock signals required for initialization and operation. The modules were developed using synchronous logic whereby all operations are synchronized by a clock signal.

The *power* port contains the squared current required to evaluate our thermal model. Its vector size *pow_bits* depends on the resolution one wants to achieve and the current range. This value comes from the test-bench and a vector size of 16 bits was chosen as a standard number.

While not required for the operation it was chosen to make the current wire temperature available via the *otemp* output port since it is plausible that this information might be useful for possible additional modules. In order to select the correct vector size *temp_bits* for the temperature the highest possible value has to be known. Recalling that we are

actually calculating $\frac{T_{wire}}{\alpha \cdot R_{th} \cdot R_{el}}$, the required amount of bits can be calculated according to equation 5.3, whereby T_{max} has to be sufficiently larger than the threshold where the current flow should be interrupted in order to prevent an overflow error when the wire temperature is very high.

$$temp\_bits = \left\lceil log_2 \left(\frac{T_{max}}{\alpha \cdot R_{th} \cdot R_{el}} \right) \right\rceil \tag{5.3}$$

Finally the *trip* port is a control signal in order to control the switch. It is 'low' during normal operation and switches to 'high' once the temperature threshold has been exceeded.

Chapter 6 covers the resource consumption of the different implementation options and shows results using the test-bench.

CHAPTER 6

Evaluation

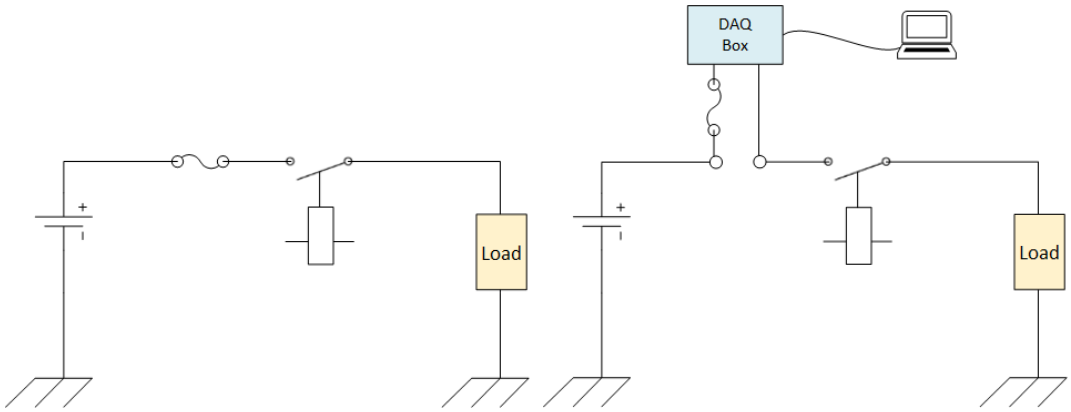
6.1 Load Profiles

The developed system should not only be lab tested with theoretical values and standardized pulses but actually make use of real world data. For this purpose it was necessary to have accurate load profiles, i.e. current samples over time. It quickly turned out that no load profiles recorded in the actual field environment, i.e. within the car, were publicly available. Hence, a data acquisition (DAQ) box was constructed in order to record the current consumption of different loads in the car.

The DAQ box consists of the following components:

- Allegro MicroSystems ACS770LCB-100U-PFF-T hall effect DC current sensor
- National Instruments NI USB-6001 DAQ box
- ATC fuse extension leads of different sizes

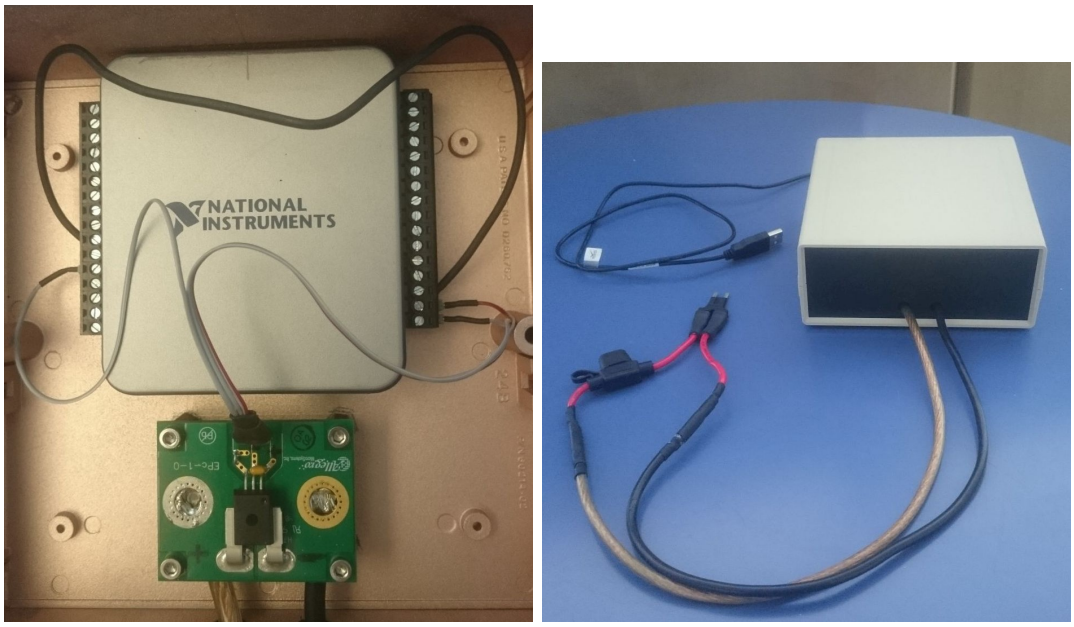
The Allegro hall effect current sensor can measure currents up to $100\,A$ and delivers a voltage between $0\,V$ and the supply voltage of $5\,V$ proportional to the current being measured. With the National Instruments DAQ box this voltage level can be measured and continuously streamed to a host computer via its USB connection. In order to tap into the electrical circuit of the loads the fuse extensions were required. As pictured in figure 6.1a the electrical circuit is protected by a fuse which is located in the fuse box. To tap into the electrical circuit without the need of any modifications to the car the fuse extensions were used, as shown in figure 6.1b. The fuse was removed from the fuse box and replaced with the fuse extension. The removed fuse was then inserted into a plug-in position on the extension leads thus keeping the protection intact. The finished setup can be seen in figure 6.2a and figure 6.2b.



(a) Typical load wiring. The load is connected to a fuse in the fuse box

(b) Connection of the DAQ box by tapping into the fuse box

Figure 6.1: DAQ box connection

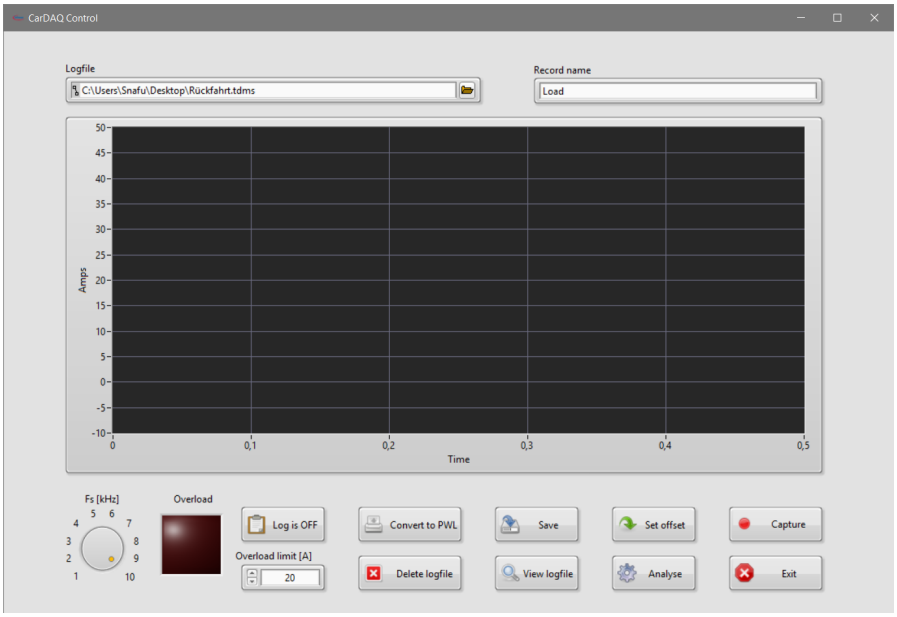


(a) DAQ box components

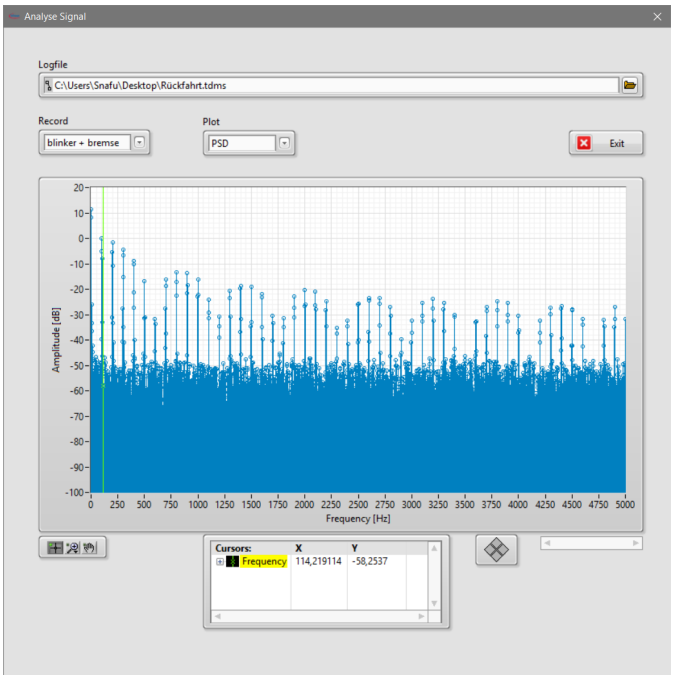
(b) DAQ box with enclosure

Figure 6.2: DAQ box

The box is USB powered and the measuring circuit is electrically isolated from the load circuit making it completely transparent for the vehicle. A LabVIEW software with GUI, see figure 6.3, was developed to continuously log data in LabVIEW's technical data management streaming (TDMS) file format with a sampling rate of up to $10\,kS/s$. It was used to record the actual current consumption of several loads in real time inside the car during several test drives. Additional features of the software included comma-separated



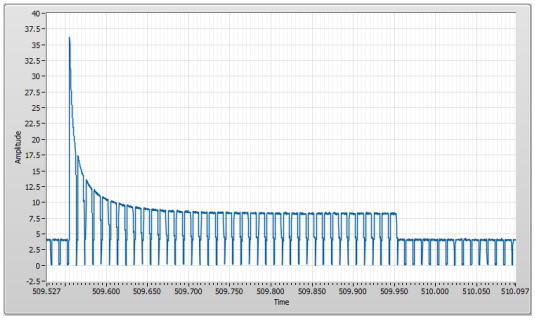
(a) LabVIEW GUI software to control the DAQ box



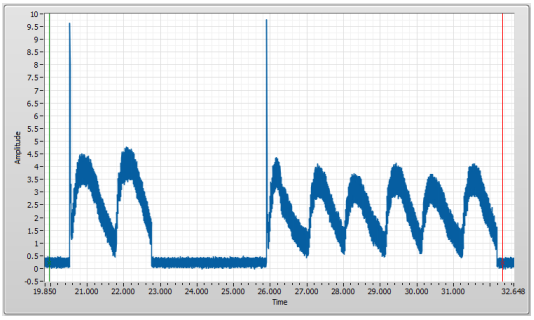
(b) Frequency analysis built into the software

Figure 6.3: DAQ GUI

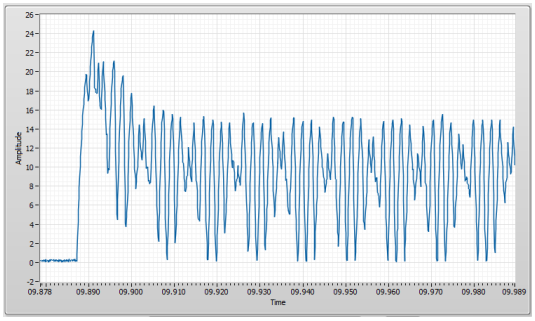
values (CSV) export and signal analysis features such as frequency analysis. Examples of recorded profiles can be seen in figure 6.4.



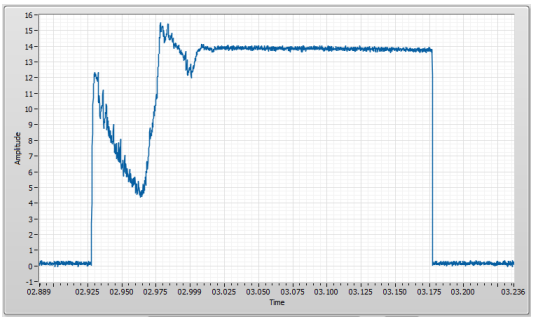
(a) Indicators and break lights are connected to the same fuse. A fixed phase shift prevents simultaneous inrush currents



(b) The rear windscreen wiper current consumption is similar to a sawtooth function



(c) This amplitude modulation resembling profile comes from the car's horn



(d) A door unlock process of the central locking system

Figure 6.4: Short sections from different load profiles

6.2 Software demonstrator evaluation

Resource consumption was of actual concern for the hardware implementation. For the software implementation, however, the focus was on having a working demonstrator which could be controlled via a host computer. Concerning the memory required for implementation one has to note that the different temperature calculation options themselves merely require a few bytes even if multiplications are replaced by summation loops. In times where MCUs have internal Flash memory of multiple kilobytes this amount of memory is insignificant. The final .hex file had a file size of 61 kB. This large amount, however, comes from the usage of so called apps in the DAVE integrated development environment (IDE) for peripheral features such as the timer, UART and the ADC. While they make addition of peripheral functions a matter of a few clicks with almost no programming effort required, they create rather bloated code that could be

written by hand much more efficiently. Additionally a software FPU stack was included to calculate a reference temperature using floating point data types. The addition of this stack was responsible for the majority of the code size.

Figure 6.5: The LabVIEW Demonstrator GUI is used to switch the load and visualize the calculated temperatures as well as the actual temperature measured with a thermocouple

To control the fuse and visualize its state and the calculated wire temperatures a LabVIEW GUI was developed, as shown in figure 6.5. In addition to displaying the wire temperatures calculated using the optimized methods, the LabVIEW GUI also shows the temperature delta calculated using the floating point *double* data type and the actual wire temperature measured with a thermocouple element for comparison.

For the test setup the demonstrator was connected to a 12 V power supply and a H7 incandescent light bulb. The load would then be connected to the demonstrator using different wires. Figure 6.6 shows the test setup using a $0.12\,mm^2$ wire. The required values to calculate the wire's temperature, namely the thermal resistance R_{th}, thermal capacity C_{th} and electrical resistance R_{el} were measured or derived using curve fitting in MATLAB to a step response for a known constant current, respectively.

During operation the actual wire temperature was measured concurrently with a thermocouple element. The maximal error between the different implementation's calculations

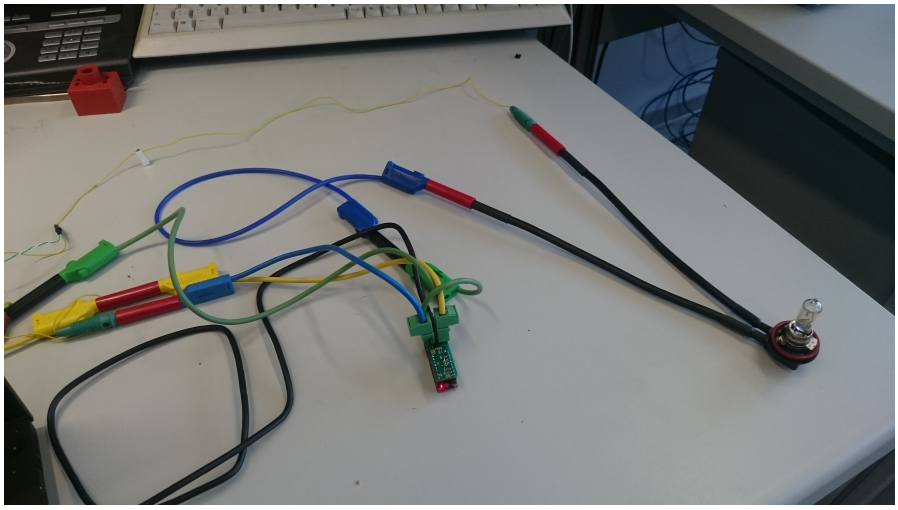


Figure 6.6: Software demonstrator connected to an H7 incandescent light bulb via $0.12\,mm^2$ wire

and the measured wire temperature is displayed at the bottom of the GUI. As explained in chapter 5 the wire temperature is calculated by the MCU and if it exceeds a specific threshold, it automatically switches off the load without any user interaction. A UART connection via USB is used for communication between the GUI and the demonstrator and acts as a power supply for the MCU as a sake of simplicity.

From the data gathered in several runs the maximum error between the actual wire temperature and calculated wire temperature was always within ±3°K, whereby these values were reached solely at the switching points of the load for a brief period. This may be due to the contacted measurement together with the fact that the accepted simple first degree RC model of a wire might lack generally insignificant higher order terms which do cause a difference upon switching actions. As seen in the hardware evaluation the temperature difference between the implementation options and the control using *double* data type was virtually nonexistent.

6.3 Hardware evaluation

For the hardware evaluation two aspects are of importance, namely resource consumption and a possible calculation error caused by the different implementation options in contrast to a floating-point calculation. While the resource consumption was given by the synthesis tool for the FPGA the calculation error was measured empirically using a test-bench.

6.3.1 Resource consumption

Developing hardware is entirely different to developing software. While the software development is done with higher level programming languages, such as C, the last step is

the compilation of the source code into machine code. This machine code represents a sequence of instructions to be executed by a CPU. The instructions which can be executed depend on the CPU being used. One of the main metrics for software is therefore the amount of memory being used by the program since it has to fit into memory.

When developing hardware there is no given instruction set. Using languages such as VHDL or Verilog one has to describe the functionality one wants to achieve. This code is eventually synthesized into a set of logic gates which are realized by connecting transistors in a specific way. The metrics for measuring the footprint of a hardware solution is thus given by the gate count which defines the number of logic gates required to implement the desired functionality.

A seemingly small change in the hardware describing code can thus result in a significantly different gate count. When developing an ASIC the goal is to have the smallest gate count possible since a higher gate count means more silicon area and therefore higher costs. When synthesizing for an FPGA the size of a design is measured in the amount of lookup tables (LUTs) and flip-flops (FFs) that it requires. The LUT and FF count cannot be directly compared to the gate count, however, since a single LUT can implement several functions requiring a different amount of gates. When creating an actual ASIC solution the results may thus vary compared to this comparison.

Instance	**Total LUTs**	**Logic LUTs**	**FFs**
Denominator approximation	84	84	28
Numerator approximation	112	112	56
FXP numerator approximation	103	103	23
FXP multiplication	297	297	23

Table 6.1: Xilinx xc7a100t utilization report using Vivado v.2014.4 (win32)

Table 6.1 shows the exact amount of LUTs and FFs required for each implementation option. As stated earlier a small change in the VHDL code may lead to a big difference in the resource consumption. This holds true even for syntactical changes where the semantics of the code remain the same. As a consequence the modules were individually tweaked until no further improvement could be achieved by the author. For two options involving FXP multiplications no digital signal processor (DSP) blocks were used but the multiplications were implemented via LUTs as well. While the FXP multiplication option requires more than twice the amount of LUTs compared to any other implementation option it requires the least amount of FFs together with the FXP numerator approximation option. While showing the exact information table 6.1 is therefore not really suited for a quick comparison of the different options. Figure 6.7 shows a graphical visualization of the resource consumption whereby a larger block means more resources are required. The denominator approximation corresponds to the *den*, the numerator approximation to the *num*, the FXP numerator approximation to the *fxnum* and the FXP multiplication option to the *fxp* block in the figure. One can easily see that for the chosen FPGA target platform the denominator approximation option requires the least amount of resources.

A possibly surprising result, however, is that the FXP numerator approximation turns out to be the second most efficient implementation option. One may have assumed that the numerator approximation option would be the more economical option since the multiplication is replaced with a summation in a loop but the additional registers and summation loop overhead required in the numerator approximation require more resources than the FXP multiplication itself.

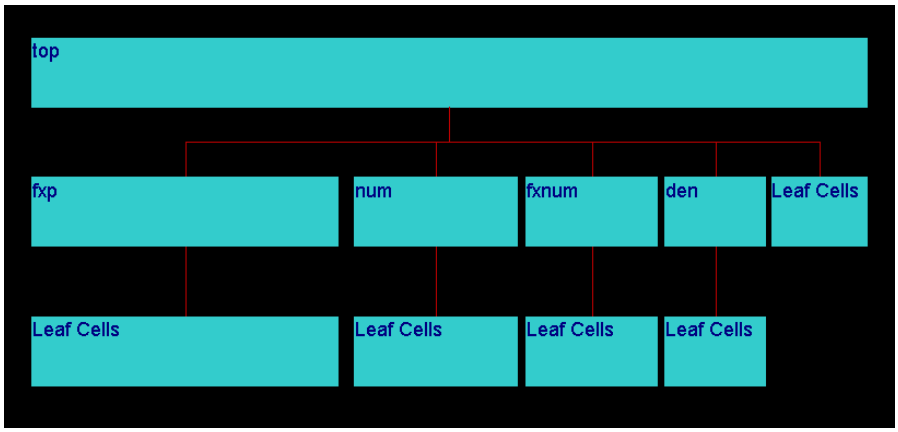


Figure 6.7: FPGA component hierarchy

6.3.2 Calculation error

In order to measure the calculation error introduced by the different options a VHDL test-bench was written whereby all the implementation modules as well as a control module evaluating the thermal model using VHDL's non-synthesizable floating point *real* data type performed their temperature calculation in parallel. The current values used as input were taken from the load profiles recorded with the DAQ box in several test drives. The test-bench therefore simulated an actual real life scenario recorded in a vehicle driving on the road.

The evaluation of several current recordings showed that none of the different options introduced any significant error. As shown exemplary in table 6.2 the maximal absolute deviation to the control module was always below 1°K. In figure 6.8 one can see a run of the test-bench with the temperature and deviation to the control module plotted for every implementation module. The current signal used as input can be seen at the top of the figure with the control module's temperature output below it.

Component	Max. deviation [°K]	Time [s]
FXP	-0.15	234.58
FXNum	0.59	194.38
DEN	-0.17	236.01
NUM	0.33	300.00

Table 6.2: VHDL simulation maximum temperature deviation

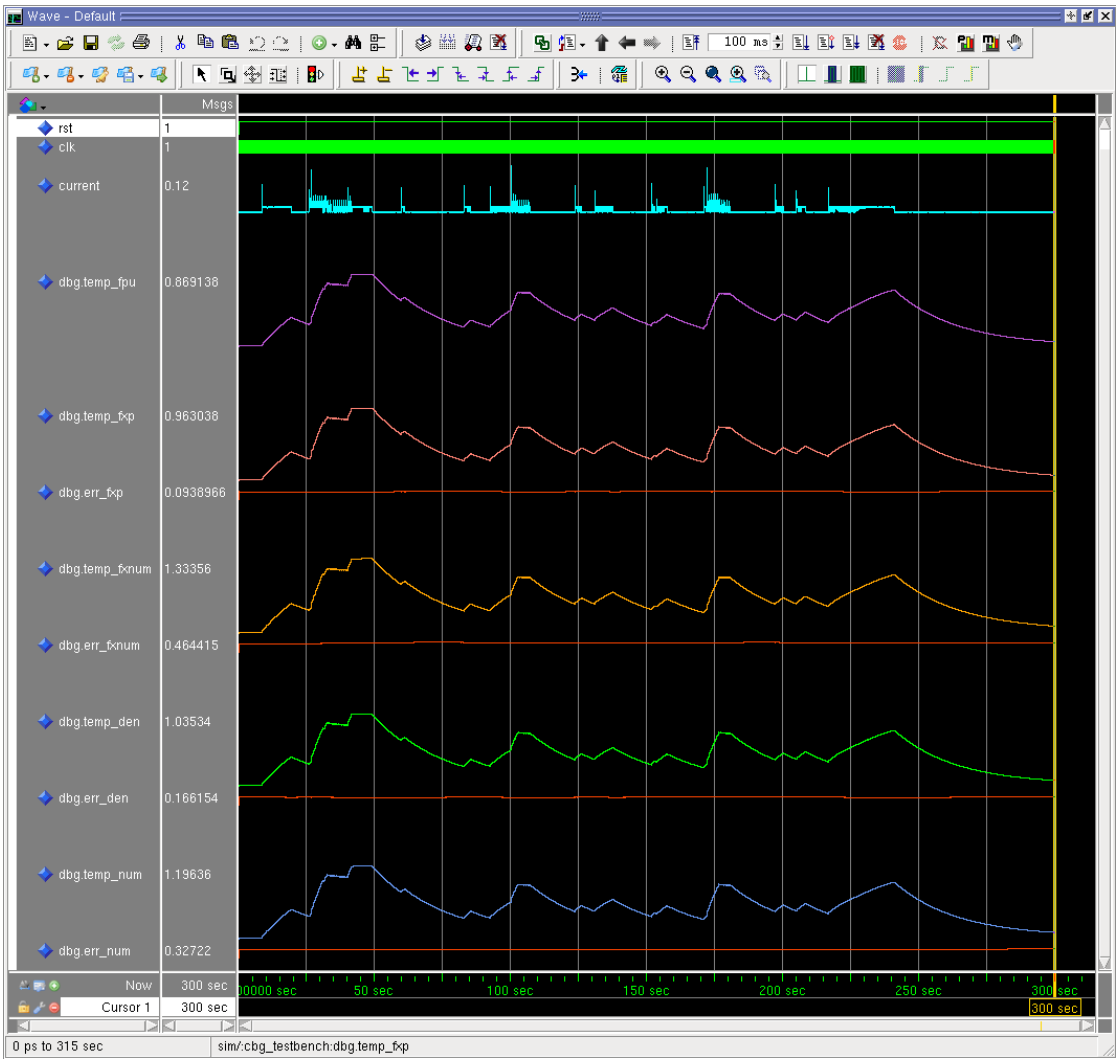


Figure 6.8: VHDL simulation of the different implementation options with current values of recorded load profiles

CHAPTER 7

Conclusion

The future of the automotive industry is going to break with conventional fusing techniques due to major changes in the electric requirements and wiring topology coming inevitably as cars develop more and more from vehicles into autonomously moving entertainment systems. The combination of high power MOSFETs and digital circuitry can be used to create resettable electronic fuses, which can be installed in areas where physical access doesn't have to be provided to the end user. Regarding the complexity of calculation, the results gathered in this document show that an efficient implementation of an electronic fuse is possible both as a hardware or software solution. Key factors are a simple thermal model of a wire and simplifications and optimizations upon evaluation of the thermal model.

Simulations with current profiles recorded during multiple test drives showed that all the implementation options are suitable for calculation. The demonstrator which was built proved to be fully functional as well and acts as a proof of concept.

However, regarding a final product, operational safety has to be taken into consideration. What happens, for example, if there is a short break in the power supply, forcing the electronic fuse to cease operation? If the power loss is not detected and thus results in the temperature delta getting lost, starting off at a delta of zero after the fuse takes up operation again may lead to potentially severe wire damage. These risks have to be addressed and taken care of in order to create a safe fuse.

List of Figures

List of Tables

Acronyms

ADC analog-to-digital converter. 22, 35, 37, 38, 48

ASIC application specific integrated circuit. 2, 3, 21, 24, 51

CPU central processing unit. 18, 26, 28, 35, 51

CSSM current sense and switching module. 8

CSV comma-separated values. 46

DAQ data acquisition. 45–47, 52, 58

DSP digital signal processor. 51

FF flip-flop. 51

FPGA field-programmable gate array. 2, 3, 22, 24, 35, 40, 41, 50, 51

FPU floating-point unit. 2, 6, 18, 19, 26, 27, 49

FXP fixed-point. 19, 20, 26, 28, 30, 31, 39, 51, 52, 57

GUI graphical user interface. 39, 46, 47, 50, 58

IDE integrated development environment. 48

IIR infinite impulse response. 28, 30, 31

LUT lookup table. 51

MCU microcontroller unit. 2, 3, 6, 8, 19, 22, 24, 26, 28, 30, 31, 35, 39, 41, 48, 50

MOSFET metal–oxide–semiconductor field-effect transistor. 7, 8, 22, 36, 55

PC personal computer. 26

PCB printed circuit board. 36

PVC polyvinyl chloride. 11

RMS root mean square. 31

TDMS technical data management streaming. 46

UART universal asynchronous receiver/transmitter. 35, 39, 48, 50

VHDL Very High Speed Integrated Circuit Hardware Description Language. 3, 40–42, 51, 52

Bibliography

[Ana03] Analog Devices, Inc. RMS Calculation for Energy Meter Applications Using the ADE7756, 2003. Application note AN-578, available at http://www.analog.com/media/en/technical-documentation/application-notes/AN578_a.pdf.

[Ana09] Analog Devices, Inc. Oversampling Interpolating DACs, 2009. Tutorial MT-017, available at http://www.analog.com/media/cn/training-seminars/tutorials/MT-017.pdf.

[ARM] ARM Ltd. Floating Point - ARM. Website. Available at https://www.arm.com/products/processors/technologies/vector-floating-point.php; accessed on 27th July, 2015.

[Bad14] Markus Bader. Auslegung und Entwurf einer Halbleitersicherung für ein Niederspannungsbordnetz. Master's thesis, Technische Universität München, October 31 2014.

[BS07] Vincent Bidault and François Schoeffler. Short circuit requirement: context, tests and wires sizing improvement. In *Automotive wiring components forum 2007*. ACOME, 2007.

[Dip09] Gebhart Dippold. Development of a system model and of algorithms for a digital gate driver for smart power switches. Master's thesis, FH Kärnten, Villach, 2009.

[Dru14] Oliver Druhm. Bordnetz der zukunft und dessen absicherung. *Elektronik automotive*, 2014(1):30–32, 2014.

[Dru16] Oliver Druhm. Ein erster schritt in die zukunft des bordnetzes, March 8 2016.

[DW85] Mark AZ Dippé and Erling Henry Wold. Antialiasing through stochastic sampling. *ACM Siggraph Computer Graphics*, 19(3):69–78, 1985.

[Ed 04] Ed Fournier (Biokinetics and Associates Ltd.). Measurement of under hood temperatures. Online report, September 7 2004. Available at http://www.mvfri.org/Contracts/biokinetics-UH%20Temp.html.

[EMW+14] S. Ecker, J. Marquardt, M. Wortberg, M. Stemplinger, and Y. Müller. Device for detecting fault in an electric line, October 30 2014. US Patent App. 14/263,073.

[Est98] Hannes Estl. Ersatz von Sicherungen durch intelligente Leistungsschalter. Master's thesis, Technische Universität Graz, May 1998.

[FGO+14] Manfred Fuisting, Rüdiger Gillhaus, Joachim Olk, Thomas Gauter, Holger Völkel, and Jan Falkenstein. Current switch and sense module for the intelligent power distribution in future e-/e-architectures. *GMM-Fachbericht-AmE 2014–Automotive meets Electronics*, 2014.

[Has10] K Joseph Hass. Synthesizing optimal fixed-point arithmetic for embedded signal processing. In *Circuits and Systems (MWSCAS), 2010 53rd IEEE International Midwest Symposium on*, pages 61–64. IEEE, 2010.

[HKS98] C.N. Huber, M.E. Kirita, and S.D. Swift. RMS converter using digital filtering, September 2 1998. EP Patent App. EP19,970,306,146.

[HT10] Y. Higuchi and S. Takahashi. Wire protection method and wire protection device, October 7 2010. US Patent App. 12/734,684.

[Ilg04] Audrius Ilgevicius. *Analytical and numerical analysis and simulation of heat transfer in electrical conductors and fuses.* PhD thesis, University of the Bundeswehr Munich, 2004.

[JB15] Martin Jaiser and Manfred Brandl. Demosystem für elektronische Sicherungen: Mehr Intelligenz im Bordnetz. *Elektronik automotive boardnetz*, 2015(9):46–49, 2015.

[Kes05] Walt Kester. *Data conversion handbook.* Elsevier Newnes, Amsterdam Boston, 2005.

[KY10] H.A. Kojori and Y. Ye. Method and apparatus for generalized ac and dc arc fault detection and protection, November 16 2010. US Patent 7,834,637.

[MJKF13] Tobias Carsten Müller, Christian Junge, Florian Kühnlenz, and Thomas Form. Dezentrale elektronische absicherung im 12v-bordnetz. *GMM-Fachbericht-AmE 2013*, 2013.

[MNU13] A. Maruyama, Y. Nakamura, and K. Ueta. Protection apparatus for load circuit, October 29 2013. US Patent 8,570,700.

[otDPM05] Great Britain. Office of the Deputy Prime Minister. *Fire Statistics, United Kingdom, 2003.* Stationery Office, 2005.

[SD16] Georg Scheidhammer and Oliver Druhm. Lösungen zur dezentralen bordnetzversorgung. *ATZ - Automobiltechnische Zeitschrift*, 118(4):48–53, 2016.

[SFWLR14] Anette Sedlmaier-Fuchs, Michael Wortberg, Stefan Lobmeyer, and Karl Ring. New approach to energy distribution backbone architecture as enabler for innovative vehicle electrical systems. *ATZelektronik worldwide*, 9(5):30–35, 2014.

[WKM15] M. Wortberg, A. Kerber, and J. Marquardt. Interruption of a current, October 15 2015. US Patent App. 14/687,156.

[Wor12] M. Wortberg. Device and method for securing an electric line with a controllable switching element, May 31 2012. US Patent App. 13/380,575.