Hydrogen and fundamental defects in electron-irradiated high-purity silicon

Jan Håvard Bleka

March 2008

Thesis submitted in partial fulfilment of the requirements
for the degree of Philosophiæ Doctor (Ph.D.)
at the University of Oslo

Series of dissertations submitted to the
Faculty of Mathematics and Natural Sciences, University of Oslo
Nr. 731

ISSN 1501-7710

Cover: Inger Sandved Anfinsen.
Printed in Norway: AiT e-dit AS, Oslo, 2008.

Produced in co-operation with Unipub AS.
The thesis is produced by Unipub AS merely in connection with the
thesis defence. Kindly direct all inquiries regarding the thesis to the copyright
holder or the unit which grants the doctorate.

Unipub AS is owned by
The University Foundation for Student Life (SiO)

Abstract

The dynamics of various fundamental defects in electron-irradiated high-purity silicon detectors (diodes) were investigated by deep-level transient spectroscopy (DLTS). Samples with various oxygen concentrations were used and hydrogen was intentionally introduced into some samples. The defect dynamics was investigated by first creating defect centres by irradiating the diodes with 6-MeV electrons and subsequently annealing the samples, isochronally or isothermally, at increasingly higher temperatures up to 400 °C while measuring the concentration of the various electrically active defects by DLTS. Based on the temperature- and time-dependent changes of the concentration of the defects, models explaining the observations were suggested.

In one study, the annealing of the di-vacancy–oxygen (V_2O) centre was investigated and modelled, and it was concluded that this centre anneals out through a dissociation resulting in a vacancy–oxygen (VO) centre. The binding energy between the vacancy and the VO centre was estimated to be $\sim$1.7 eV.

In another investigation, a defect centre annealing out after a few weeks at room temperature was found to have two energy levels in the band gap: one, labelled E4, 0.37 eV below the conduction-band edge (E_c) and the other, labelled E5, 0.45 eV below E_c. Comparison with annealing studies performed with Fourier-transform infrared spectroscopy (FTIR) suggested that the defect may be a di-interstitial–oxygen (I_2O) complex. E5 is known to correlate with the leakage current of silicon detectors, and it was suggested that the oxygen concentration should be minimised to reduce the formation of I_2O centres and thus reduce the leakage current.

Several annealing studies with hydrogenated samples were performed. These resulted in the identification of a hydrogen-related level at $E_c - 0.37$ eV as a vacancy–oxygen–hydrogen centre, which was labelled VOH^*. This centre was seen to form when positively charged hydrogen diffused in from the surface of the silicon diodes and reacted with the VO centre at depths with locally high hydrogen concentration. VOH^* was seen to break up when the hydrogen diffusion had resulted in a lower, more uniform hydrogen concentration. Possible identification of other hydrogen-related defect levels were also put forward; in particularly a hole trap located 0.23 eV above the valence-band edge which is suggested to be a di-vacancy–hydrogen (V_2H) centre.

Secondary-ion mass spectrometry (SIMS) was used to measure the depth profile of hydrogen in some of the samples.

Acknowledgements

This chapter could, and should, have been much longer, but I will save some space here and instead try to show my gratitude in real life to those that were not included in writing.

My gratitude goes first of all to my main supervisor Bengt Gunnar Svensson who has let me do things my own way and at my own pace, even though this has rarely been the shortest way to the goal. Bengt has, with his decades of experience, shaped our articles into what they are – at times by extensive rewriting – at times just by adding a magic word. Especially rewarding has it been to sit down and compose articles together with him – he has always listened carefully when I have disagreed and even quite often understood and taken into account my concerns. A large thank you to my second supervisor Edouard V. Monakhov who has had the undesirable task of making the initial corrections to our articles to prepare them for Bengt's scrutiny. Also, his experimental experience has been invaluable when my experimental methods and brain have failed.

My sincere gratitude goes to Mayandi Jeyanthinath – for being a reliable friend, for all the curry and for keeping a desk so overloaded with scientific papers that it has attracted some attention away from my own desk. To Ulrike Grossner, for the laughs, the mentor services, the lab collaboration and the smell of real beer. Lasse Vines, Mads Mikelsen and Lars Sundnes Løvlie, for helping me out in many an existential crisis concerning DLTS and related topics. Lars in particular, for never trusting other people's calculation and thus being valuable for consistency verifications; Mads, for making me question opinions I took for granted.

I am grateful to Simen Bræck and Knut Johan Øen who were my flatmates during my whole Ph.D. period; all those discussions in the middle of the night has taught me more than what can be put down in text. To Thomas Bryhn and Jørnar Heggsum Hubred, for providing me with much of the trash and entertainment required for the completion of my project.

Thank you, Aleksander Ulyashin, for spending many late hours patiently trying to cure my persistent naivety. Thank you, Terje Finstad, for patiently explaining me why my self-made circuits have been shooting sparks when my attitude has been that they should not. Thank you, David Wormald and it-hjelp, for the much valued help in times of broken pipes and low computing power.

I feel deeply indebted to Danie Auret and Ioana Pintilie who took very good care of me during my two-week-long research stays in Pretoria (2005) and Hamburg (2007), respectively – thank you!

To all the rest of my present and former colleagues: It is sad to abandon my messy desk and not attend the cherished MiNaLab lunches any more; experience shows that it is difficult to keep contact, but I choose to be optimistic and hope to maintain the contact with you also in the following.

Last, but unfortunately most important of all: Sorry everyone that I have so many times ”had to” be at MiNaLab instead of being out in the real world as a good friend. Let us see if this can change in the coming years.

Yes, and I am not able to leave out a warm thank you to the developers of free software who just out of good will provide the world with phenomenal applications which allow me to materialise my ideas (and documents such as the present one). These people make life with computers more enjoyable.

Contents

List of publications

I **Defect behaviour in deuterated and non-deuterated n-type Si**
J. H. Bleka, E. V. Monakhov, A. Ulyashin, F. D. Auret, A. Yu. Kuznetsov, B. S. Avset and B. G. Svensson
Sol. St. Phen. **108–109**, 553 (2005)

II **Annealing of the divacancy-oxygen and vacancy-oxygen complexes in silicon**
M. Mikelsen, J. H. Bleka, J. S. Christensen, E. V. Monakhov, B. G. Svensson, J. Härkönen and B. S. Avset
Phys. Rev. B **75**, 155202 (2007)

III **Room-temperature annealing of vacancy-type defect in high-purity *n*-type Si**
J. H. Bleka, E. V. Monakhov, B. S. Avset and B. G. Svensson
Phys. Rev. B **76**, 233204 (2007)

IV **On the identity of a crucial defect contributing to leakage current in silicon particle detectors**
J. H. Bleka, L. Murin, E. V. Monakhov, B. S. Avset and B. G. Svensson
Appl. Phys. Lett. **92**, 132102 (2008)

V **Rapid annealing of the vacancy-oxygen center and the divacancy center by diffusing hydrogen in silicon**
J. H. Bleka, I. Pintilie, E. V. Monakhov, B. S. Avset and B. G. Svensson
Phys. Rev. B **77**, 073206 (2008)

VI **Dynamics of irradiation-induced defects in high-purity silicon in the presence of hydrogen**
J. H. Bleka, E. V. Monakhov, B. S. Avset and B. G. Svensson
To be published

Relevant publications not included

- **DLTS Study of Room-Temperature Defect Annealing in High-Purity n-Type Si**
  J. H. Bleka, E. V. Monakhov, B. S. Avset and B. G. Svensson
  ECS Trans. **3**(4), 387 (2006)

- **Charge carrier removal in electron irradiated 4H and 6H SiC**
  M. Mikelsen, U. Grossner, J. H. Bleka, E. V. Monakhov, R. Yakimova, A. Henry, E. Janzén and B. G. Svensson
  To be published

- **Alignment of intrinsic defect levels in the electronic band gap of 4H and 6H SiC**
  U. Grossner, M. Mikelsen, J. H. Bleka, E. V. Monakhov and B. G. Svensson
  To be published

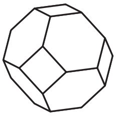
Chapter 1

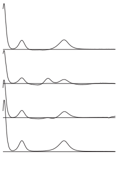
Introduction

Silicon is still the material playing the central role in the fabrication of electronic devices. It is a material with many advantageous properties and since the fabrication processes in the silicon industry have matured for several decades, silicon devices are generally cheaper to produce than devices from other materials. In addition, device fabrication has not been proven in all types of semiconductor materials and in general, the complexity increases in compound materials. Silicon is, however, not perfect in all aspects. It is not especially well suited for optical devices, as it does not have a direct band gap; several other materials have larger band gaps than silicon – something which is advantageous in many applications; and some materials, for instance SiC, or even better: diamond (effective *n*-doping is, however, a major problem with this material), can withstand more irradiation by high-energy particles than silicon before the devices become unusable (Adam *et al.* 2003 [1]). In the construction of the Large Hadron Collider (LHC) (at CERN, close to Geneva), which is currently scheduled to start operation in May 2008, the radiation hardness of the particle detectors is a great concern and a major research-and-development challenge. For instance, in the Compact Muon Solenoid (CMS), the fluence of fast hadrons will during the expected lifetime of the LHC reach $\sim 5 \times 10^{15}$ cm^{-2} at a distance of 7 cm from the beam centre. If the radiation hardness of the detectors is not significantly improved, the inner tracking detectors will have to be moved further away from the collision point for the proposed upgrade to the Super Large Hadron Collider (SLHC) around year 2015 (Gianotti *et al.* 2005 [2] (p. 318)), an upgrade which will, hopefully, give an increase in the luminosity by a factor of ten (URL [3], Gianotti *et al.* 2005 [2]).

Hydrogen is known to passivate, i.e. make electrically inactive, or shift the energy levels of many of the radiation defects in silicon (Pearton 1986 [4], Coutinho *et al.* 2003 [5], Bonde Nielsen *et al.* 1999 [6], Pellegrino *et al.* 2001 [7], Monakhov *et al.* 2004 [8]). It has, therefore, been important to investigate the defect dynamics in the presence of an increased hydrogen concentration. With sufficient knowledge, it may be possible to introduce hydrogen such that detrimental defects, i.e. those that cause an increased leakage current or that trap electrons, are transformed into defects with less harmful properties. This is the background for the investigations of hydrogenated samples in the present project. In particular, the interaction between hydrogen and vacancy-type defects has been addressed.

Hydrogen is, however, not a simple substance to investigate. It reacts easily with many defects and impurities/dopants (Pearton 1986 [4], Zundel and Weber 1989 [9]) and it can, thus, be difficult to isolate and focus on one particular reaction. The diffusivity of hydrogen, and thus also the reaction rates, depend on the charge state of the hydrogen atoms (Markevich and Suezawa 1998 [10], Weber *et al.* 2003 [11], Huang *et al.* 2004 [12]); the doping and temperature of the material will, therefore, affect the defect dynamics.





Chapter 2

Basic concepts

2.1 Semiconductors

2.1.1 Electrical properties of crystals

Let us start with the atom (figure 2.1); an atom consists of a heavy, positively charged nucleus (built from protons and neutrons) with, if the atom is neutral, as many bound electrons as there are protons in the nucleus. As the electrons have negative charge, they repel each other, while they are attracted to the positively charged nucleus. Quantum physics allows electrons to be situated only at a few discrete distances from the nucleus and also puts restrains on how many electrons can reside at a each of these distances. This causes the electrons to organise themselves in these shells with given energies, filling up from the lowest energy closest to the nucleus.

If several atoms are put closely together, the energy landscape of the electrons changes, and for electrons at certain energy levels, paths over to neighbouring atoms may open, enabling some of the electrons to move freely from one atom to the next. If atoms are placed in a regular pattern, there are several possible outcomes depending on the properties of the atoms and the pattern in which they are organised. If the structure allows many electrons to move freely in the lattice, we have a metal – if no electrons are free to move between the atoms, we have an insulator. A semiconductor is

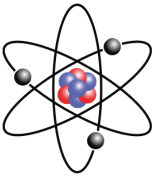


Figure 2.1: Illustration of a lithium atom (Wikipedia).

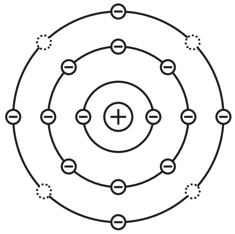
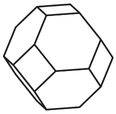


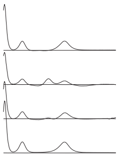
Figure 2.2: Illustration of the two filled and the one partly filled electron shells in a neutral silicon atom.

in-between these two scenarios; in a perfect semiconductor lattice, no electrons are free to move at low temperatures, but many electrons are bound so weakly to the nucleus that their thermal energy is occasionally large enough to let them escape the energy well and allow them to move freely in the lattice.

Silicon atoms have 14 protons in the nucleus and the surrounding 14 electrons are organised in shells of two, eight and four electrons with room for four more electrons in the outermost shell (figure 2.2). Silicon atoms naturally organise themselves in the same lattice structure as that of the carbon atoms in diamond. A unit cell of the diamond structure is shown in figure 2.3. When silicon atoms are put into a diamond structure, four of the states of the outermost electron shell will be shifted to a higher energy which will have a low electron occupancy, since the electrons will first populate the states with lower energies – the ensemble of all these states will form the conduction band of the crystal – while the remaining four states will get a reduction in the energy level – the these states will form the valence band. As nature tries to lower its energy, the four electrons in what used to be the outermost shell of the individual silicon atoms will normally reside in the valence band and exactly fill this. Thus, an electron in the valence band will not be free to move to a neighbouring atom, as there are usually no vacant energy states there without going to higher energies. For electrons to move freely between neighbouring silicon atoms, they must somehow acquire enough energy to lift it up across the forbidden band gap (1.12 eV for silicon) and into the conduction band where there are many available energy states.

2.1.2 Doping semiconductors





Ideal, flawless semiconductors (intrinsic semiconductors) have little value for practical applications, because these systems are rather rigid with limited response at room temperature (as nearly all electrons are in the valence band, there are no available states in the neighbouring atoms into which the electron can move, unless it gets excited up to the free states in the conduction band). To make the material more exciting, free charge carriers are introduced. Since the number of intrinsic electrons is exactly enough to fill the electron states in the valence band, in principle, any additional electrons introduced

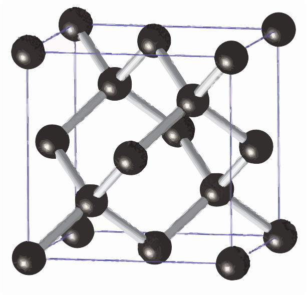


Figure 2.3: A unit cell of a lattice with diamond structure (H. K. D. H. Bhadeshia).

into the lattice will have no state in the valence band into which to fall and will, therefore, float freely around in the conduction band. A semiconducting material with an excess of free electrons is called *n*-type material. A silicon crystal can be made *n*-type by replacing some of the silicon atoms by phosphorous atoms. The phosphorous atom has one proton and one electron more than the silicon atom. In a silicon lattice, this additional electron becomes very loosely bound and the thermal energy at room temperature is enough to make it escape the host atom and jump up to the conduction band.

Another way of increasing the mobility in an intrinsic semiconductor is to remove some of the electrons in the valence band such that the band is no longer completely filled. When an electron is removed from the valence band, a hole is created in its place. This hole can move around in the valence band just as freely as electrons in the conduction band. Materials with an excess concentration of holes, compared with intrinsic material, is said to be *p*-doped. In silicon, holes can be introduced by doping with boron atoms, which have one electron less in their outer shell than the silicon atoms.

2.1.3 *pn* junctions

Something very interesting happens when *n*-type and *p*-type material are put in contact with each other. Since the *n*-type side has more electrons than the *p*-type side, the thermal motion of electrons will cause a diffusion of electrons over to the *p*-type side. Similarly, the higher hole concentration in the *p*-type material will cause diffusion of holes over to the *n*-type side. This exchange of charge carriers will cause a positive net charge on the *n*-type side of the junction, and a negative net charge on the *p*-type side – thus, an electric field is building up. This electric field will work against the net diffusion of charge carriers, and a stationary situation will be reached when the diffusion is balanced by this internal electrical field. The electric field will cause a net force on any free charge carriers, and there is, therefore, close to no free charge carriers in the volume with the electrical field – this volume is the depletion zone. The energy landscape of an electron in the proximity of the *pn* junction is illustrated in figure 2.4 –

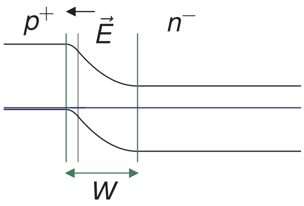


Figure 2.4: Band diagram for a *pn* junction. The Fermi level is shown as the blue line.

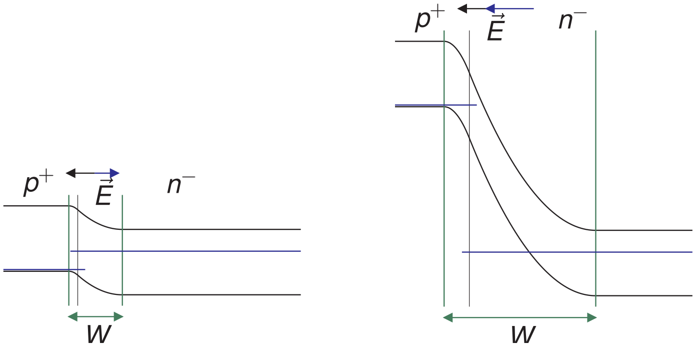
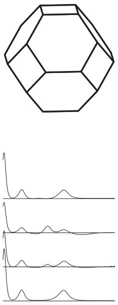


Figure 2.5: Band diagram for a *pn* junction with forward (left) and reverse (right) bias. Here, the blue line is the quasi Fermi level.

this is usually how the band diagram for a *pn* junction is shown. In this situation there is a small possibility of an electron in the conduction band having enough energy to climb the hill caused by the electric field, but this is balanced by the small possibility of an electron in the *p*-type material being excited to the conduction band and sliding down the hill to the *n*-type side.

2.1.4 Applications of *pn* junctions

A *pn* junction is a rectifying diode – only voltage applied in the forward direction of the diode will cause an electric current. If the positive pole of a battery is connected to the *p*-type side of the diode and the negative pole to the *n*-type side, the electric field in the diode will be reduced (left side of figure 2.5). In this case, electrons will again start diffusing from the *n*-type material over to the *p*-type. The larger the applied bias, the larger the flow of electrons. If the voltage is applied in the opposite direction, the energy barrier for the electrons will be very high and the diffusion of electrons even smaller than in an unbiased diode. On the *p*-type side, there are still very few electrons free to move and there is, therefore, close to no current flowing across the reverse-biased *pn* junction.



A *pn* junction also has a photovoltaic effect – it may be used as a solar cell. When photons with sufficient energy hits a semiconductor, electrons may be excited from the valence band to the conduction band such that electron–hole pairs are formed. If this happens sufficiently close to, or inside, the depletion zone, there is a substantial probability of the minority carrier being swept across the depletion zone (e.g., electrons are minority carriers in *p*-type material). As this process continues, there will be an accumulation of holes on the *p*-type side and electrons on the *n*-type side of the junction. The resulting charges will cause an electric field opposite to the internal one, and the barrier height of the *pn* junction will be reduced. This will cause a situation similar to the one in a junction with forward bias (see figure 2.5), where the quasi Fermi level is higher on the *n*-type side than on the *p*-type side. This difference in the quasi Fermi level is the voltage across a photovoltaic cell that can perform work in an electrical circuit.

The last application I will mention is the one that has been most relevant for this project. When a *pn* junction is reverse biased, it can be used to detect particles or photons that hit the diode, given that they cause a certain amount of electrons to be excited to the conduction band. For this application it is often desirable to keep the reverse bias strong and the doping low to get the depletion region as large as possible, because when an ionisation even occurs in the depletion region, the excited electrons and holes will be swept out of the depletion zone in opposite directions and cause a measurable charge pulse.

2.1.5 Radiation defects

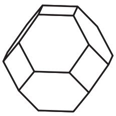
Low-energy radiation does not cause degradation of the detectors, but when the energy of the incident particles is sufficiently high, they will start displacing silicon atoms from their position in the crystal lattice. At the LHC at CERN, an enormous amount of particle detectors will be used, in layer upon layer, to track different particles created in the proton–proton collisions that are scheduled to begin in May 2008 (Wikipedia [15]). The innermost pixel detectors will be as close as 7 cm from the centre of the proton beam; here, the fluence of fast hadrons will reach $\sim 5 \times 10^{15}$ cm^{-2} during the expected lifetime of LHC (Gianotti *et al.* 2005 [2]).

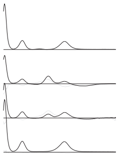
Figure 2.6 shows an example of what may happen when a silicon atom is displaced and becomes a self-interstitial. Now two defects have been created in the lattice, and the energy landscape of the nearby electrons is modified. In this example, the vacancy that is left after the silicon atom may easily move around and attach to an oxygen atom to form the vacancy–oxygen centre (VO) (Brotherton and Bradley 1982 [16], Song *et al.* 1990 [17], Keskitalo *et al.* 1997 [18]). This centre has place for an electron. The problem is that the electron that fills this state will neither have the energy of electrons in the valence band, nor that of electrons in the conduction band – its energy level will be in-between: 0.18 eV below the conduction band edge (or 0.95 eV above the valence band edge). This is undesirable in several ways: Charge-carrier traps, such as the VO centre, will reduce the concentration of free charge carriers, because electrons in the conduction band can fill these states at lower energies and thus become immobile (figure 2.7 illustrates the energy landscape in such a conduction band). A more important problem, yet, arises with energy levels close to the middle of the band gap – then valence-band electrons can rather easily be thermally excited up to the energy needed to fill the defect state, and from this state it can get excited further up to the conduction band. Hence, such defect levels in the middle of the band gap may act as electron–hole pair generators – they make it easier for electrons to be excited from the valence band to the conduction band. They may also act as recombination centres at which a conduction-band electron and valence-band holes meet and annihilate. In the case of the particle detectors, such generation centres and recombination centres will increase the noise in the circuit and thus reduce the signal-to-noise ratio.

In addition to the creation of detrimental generation centres, operation in a radiation-hard environment causes a steady increase in the concentration of acceptor-like defects (Moll 1999 [19], Lindström *et al.* 2001 [20], URL [21]). For detector applications, p^{+}–n^{-}–n^{+} structures, i.e. the bulk is weakly *n*-doped, are normally used. As the operation causes generation of acceptors, the effective concentration of free carriers (N_{eff}) in the bulk will first decrease, before the concentration of holes exceeds that of free electrons and the bulk becomes more and more *p*-type. In this situation, the depletion zone will have shifted from the p^{+}–n junction over to what is now the n^{+}–p junction on the back side (URL [21]).

As the radiation dose to which a particle detector has been exposed increases, the leakage current, and thus also the noise, increases (due to the electron–hole generation centres), the relative strength of the ionisation signal decreases (due to the increase in the acceptor concentration), the full-depletion voltage increases (also due to more acceptors), resulting in an even smaller signal if the bias is not increased. All these effects degrade the detection sensitivity of the structure. The leakage current can even, in the worst case, cause thermal runaway due to ohmic heating (Gianotti *et al.* 2005 [2]).

How much high-energy radiation a material can withstand before being rendered unusable as a detector, is referred to as radiation hardness and radiation tolerance. As it is very costly and time consuming to install a new set of detectors in the detector set-ups at CERN, considerable research has been made to engineer more radiation-hard detectors (see, e.g., the report of the CERN RD50 collaboration and references therein (URL [22])). One method is to search for materials where a certain radiation dose causes less permanent damage; epitaxial-grown (EPI) silicon is suggested to be an example of this (Kramberger *et al.* 2003 [23]). Other methods can be to introduce lattice defects that can trap radiation defects in places where they cause less damage (gettering) or to introduce defects that can combine with the harmful radiation defects to alter the energy level of the electron states and thus make them less harmful to the application. Hydrogen is one such impurity that is known to easily react with many different defects and alter their electron states. The present project was initiated to shed more light on the different reactions that hydrogen can cause in radiated silicon.





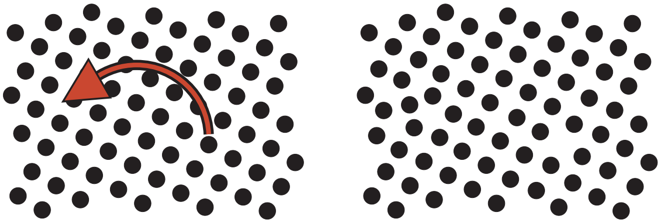


Figure 2.6: Illustration of a silicon atom being knocked out of position, leaving behind a vacancy and becoming a self-interstitial in its new position.

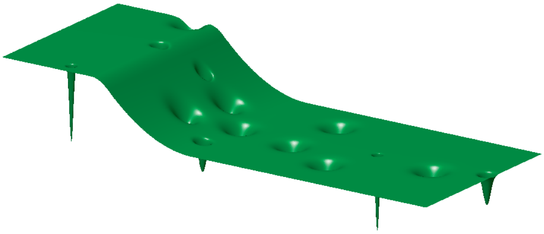


Figure 2.7: Illustration of the energy landscape experienced by an electron in the conduction band close to a p^+–n junction with the presence of lattice defects.

2.2 Characterisation techniques

2.2.1 *CV* measurements

A simple and very useful measurement technique is to measure the change in capacitance as a function of the reverse voltage applied across a diode to determine the doping concentration. When the reverse bias is increased slightly, the electric field, whose gradient is given by the concentration of fixed net charges (i.e. doping atoms), will increase and cause the depletion zone to be extended. As capacitance is defined as the change in the amount of charge as a function of the voltage, a depletion zone works as a non-linear capacitor – a change in the voltage causes a change in the amount of charge in the depletion region. Given uniform doping concentrations and abrupt interfaces, this relationship easily enables us to find the doping concentration of asymmetrical *pn* junctions (i.e. p^+–n or p–n^+) and Schottky diodes (*p*-type or *n*-type material coated with a suitable metal). The capacitance C relates to the width of the depletion region W by

$$C = \frac{\varepsilon A}{W}, \tag{2.1}$$

where ε is the permittivity of the material and A the effective area of the diode. The depletion width is

$$W = \sqrt{\frac{2\varepsilon(V_0 - V)}{q}\left(\frac{1}{N_\mathrm{a}} + \frac{1}{N_\mathrm{d}}\right)}\,, \tag{2.2}$$

where V_0 is the potential barrier when no external bias is applied, V the applied bias, q the elementary charge, N_a the net acceptor concentration (the doping concentration) in the p-type material and N_d the net donor concentration in the n-type material. If, for instance, the p-type side is doped much stronger than the n-type, N_a^{-1} can be neglected and equations 2.1 and 2.2 can be combined into

$$\frac{1}{C^2} = \frac{2}{q\varepsilon A^2 N_\mathrm{d}}(V_0 - V)\,. \tag{2.3}$$

This expression describes a straight line of the form $y = mx + b$, as the left side equals a constant multiplied with the variable V in addition to the constant where V_0 is included. Thus, to find N_d we plot $\frac{1}{C^2}$ as a function of V, which ideally gives a straight line, to find the slope m, and from

$$m = -\frac{2}{\varepsilon A^2 q N_\mathrm{d}} \tag{2.4}$$

we get

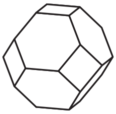
$$N_\mathrm{d} = -\frac{2}{\varepsilon A^2 q m}\,. \tag{2.5}$$

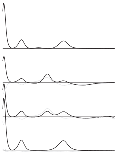
2.2.2 Deep-level transient spectroscopy (DLTS)

It is helpful to look at the Fermi level when measuring DLTS. If a system is cooled slowly to absolute zero (0 K), all the electrons will be as low as possible in the energy landscape. All the electron states will then be occupied from the lowest energies up to a level where there are no more electrons – this is the Fermi level. At absolute zero, there are no electrons above the Fermi level.

When measuring DLTS, we first fill the electron traps by dipping them into the Fermi sea; after they have been pulled out again, the electrons will, because of their thermal energy, escape the traps again as the system attempts to establish equilibrium. The deeper the trap – i.e. the higher the electron is required to jump to escape from its trap and reach the conduction band – the more time it will take for the escape to be probable.

I will describe one cycle in the measurement routine – this cycle will result in one data point in the DLTS spectrum. The sample is reverse biased most of the time during the measurement, and this is where the cycle starts (figure 2.8): A few electrons occasionally acquire enough energy to jump up to the conduction band, just to reside there for a while before falling back into a vacant trap either above or under the Fermi level.[1]





[1] The Fermi level is not defined when the system is not in equilibrium – as is the case when we apply a bias – so this is actually just the quasi-Fermi level.

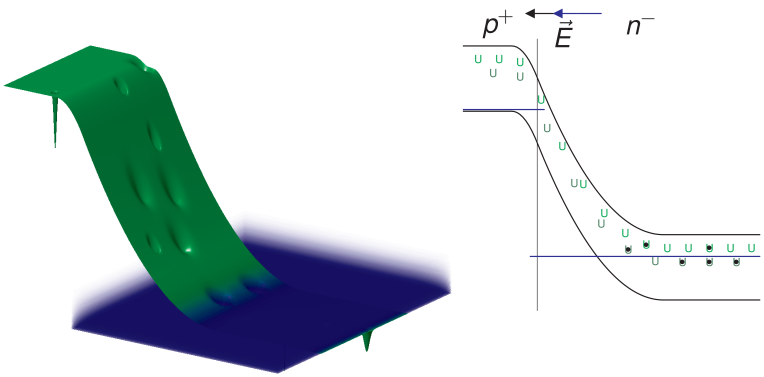


Figure 2.8: The steady state before measuring a data point to add to the DLTS spectrum. The blue haze on the left is an illustration of the Fermi sea at finite temperature, where the intensity indicates to probability of finding an electron at a certain energy.

To start the measurement cycle, the reverse bias is removed for a short moment. Figure 2.9 shows the band diagram in this unbiased state. Some of the traps that used to be empty, due to their large energy distance to the Fermi sea, become filled as they are now below the Fermi level.

When we have waited sufficiently long to allow all the submerged traps to capture an electron (typically milliseconds), we reapply the reverse bias and immediately start measuring the capacitance of the sample (figure 2.10). As always at finite temperatures, the electrons will shake and wiggle violently around in their respective traps and they will all, eventually, be able to free themselves from the potential well surrounding the trap. The electrons in the shallow traps escape at a higher rate, but the electrons escape also the deep traps, eventually. Had one of these electrons escaped its trap where the conduction band is flat, it would soon be captured by another trap (or the same one for that matter), but in the depletion region this is unlikely because the electric field will immediately sweep the electron out of the region.

As the electrons escape and flow back into the non-depleted *n*-type side, there is a change in the measured capacitance. After the reverse bias was reapplied, the capacitance has increased with time in the manner shown in the graph in the upper left corner of figure 2.11. The time required for an electron to escape a trap does not only depend on the depth of the trap, but also the temperature of the sample, as this will determine how violently the electrons wiggle. Therefore, if we assume that we have only one kind of defect, with a certain trap energy, the capacitance transient will saturate exponentially, like for instance the capacitance curve marked T_2 in figure 2.11. Had the sample been warmer, the electrons would have had more energy and they would escape the traps in a shorter time; the capacitance transient would then look more like the curve marked T_3, and vice versa if the measurement had been performed with a cooler sample.

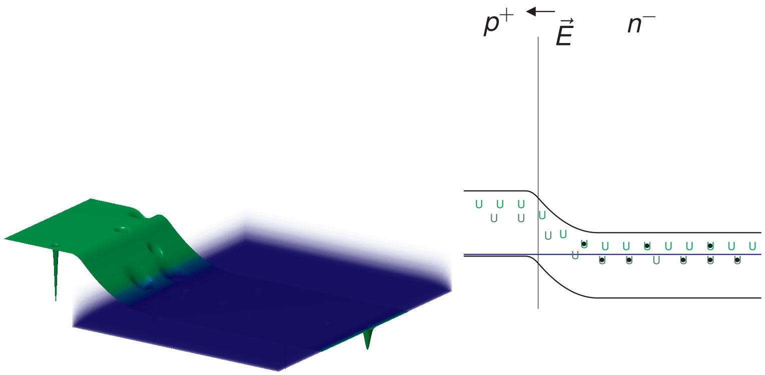


Figure 2.9: Unbiased p^+–n junction.

So the electrons escaped their traps at a certain rate and resulted in an exponential capacitance transient with corresponding time constant. To construct a DLTS spectrum, a weighting function (also known as a correlation function) is applied to the capacitance transient. A time constant is chosen for this weighting function, and the result of the mathematical operation performed gives the maximum amplitude when the time constant of the capacitance transient matches that of the weighting function. One weighting function that can be used is the lock-in type shown in figure 2.11. If the time window of the lock-in type weighting function is 320 ms, it will have the value 1 for the first 160 ms of the capacitance transient and -1 for the next 160 ms. Multiplying the weighting function with the capacitance transients gives the desired result: For slow and fast transients, such as T_1 and T_4, respectively, the output value of the operation is small, while for the transients with time constants matching the weighting function, such as T_2 and T_3, the multiplication gives a larger value.

When the whole operation is repeated at a higher and higher temperature, the weighting function will cause a peak at the temperatures where the electrons have the correct thermal energy to escape their traps with a rate that matches the weighting function. If we get two peaks in the scanned temperature region, it means that the electrons are emitted from traps with two different energy levels. When measuring the peak of the deepest trap, there will also be a contribution to the capacitance transient from the shallow trap, but this change in the capacitance is so fast that the weighting function will not be influenced by it. The amplitude of a DLTS peak is proportional to the concentration of the electron trap causing it; intuitively, many traps will emit many electrons and many electrons will cause a large transient.

If weighting functions with different time constants are applied to the capacitance transients, each resulting in a separate DLTS spectrum, it is possible to collect, from the peak position of a given defect in each of the spectra, a set of data giving the emission rate versus temperature. Subsequently, an Arrhenius plot yielding the activation enthalpy (approximately the energy difference between the conduction band and the

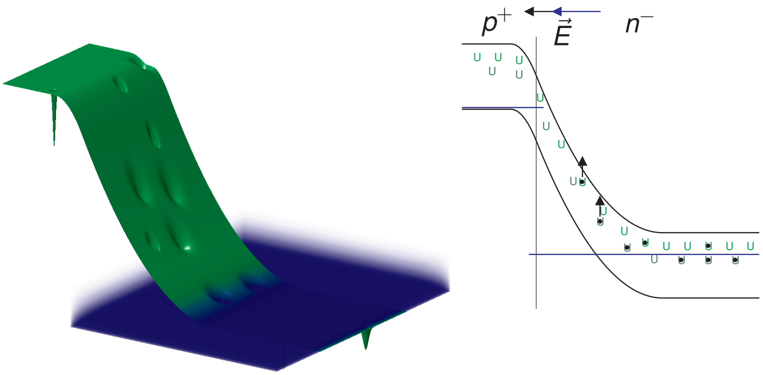


Figure 2.10: Reverse bias is again applied to the p^+–n junction and the electron traps above the quasi Fermi level emit the captured electrons and remain empty.

defect level) and apparent capture cross section of the electron trap can be made. The capture cross section also influences the rate with which electrons are emitted.

For more in-depth information on DLTS, see, for instance, Pellegrino 2001 [24], Mikelsen 2007 [25], Alfieri 2005 [26] or Blood and Orton 1992 [27].

Alternative applications of DLTS set-ups

A DLTS set-up can also be used for other investigations than ordinary DLTS measurements. Keeping the temperature constant and varying the duration of the of the filling pulse allows for the real capture cross section of a charge carrier trap to be found. Then the DLTS signal will become stronger when the pulse is sufficiently long for the traps to capture charge carriers.

Keeping the temperature and the steady state reverse bias constant while varying the pulse voltage amplitude allows for the defect concentration at different depths to be measured. This can be used to get at depth profile of a specific defect level. It is my personal opinion that the importance of this possibility can not be overestimated! For instance, profiling was essential for understanding the defect dynamics in the studies presented in paper V and paper VI.

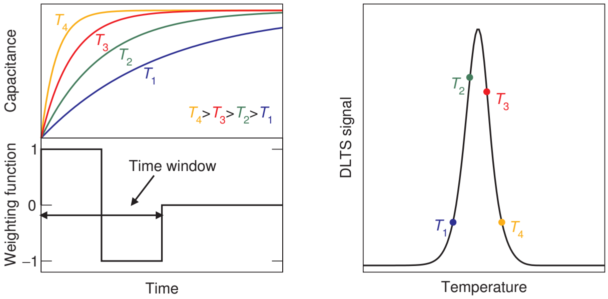


Figure 2.11: The rate with which traps emit their captured electrons depends on the temperature. The DLTS extraction will give the maximum result when the capacitance transient resonates with the chosen weighting function.

Chapter 3

Experimental details

3.1 Sample structures

The results published in this project are all obtained using SINTEF-processed detector-grade material. Float-zone (FZ) grown samples, diffusion-oxygenated FZ (DOFZ) samples and magnetic-Czochralski (MCz) grown samples have all been used for various investigations. The physical measures and the structure, which are the same for all these types of samples, are shown in figure 3.1. The thin ring of *p*-doped material around the *p*-type layer is known as the guard ring; with this attached to the ground potential, it will prevent unwanted currents from flowing along the edge surfaces of the detector and it will cause the geometry of the active volume to be defined more accurately. The guard rings have, however, not been used during these experiments since only minor improvements of the DLTS measurements are expected, given that the front contact and back contact are good.

All samples used in this work have been of high purity and have had a phosphorous concentration as low as $\sim 5 \times 10^{12}$ cm^{-3}. Mainly, the studied defects have been generated by irradiation by 6-MeV electrons and they have always been characterised by DLTS; the irradiation dose has, therefore, also been low: $\sim 5 \times 10^{12}$ cm^{-2}. The electron irradiation was performed at the Alfvén laboratory at KTH in Stockholm. Table 3.1 shows the oxygen concentration and the carbon concentration of the three various materials used.

Table 3.1: Details of the various samples investigated in the present work.

Material	**Oxygen conc.**	**Carbon conc.**
FZ	$<5 \times 10^{15}$ cm^{-3}	2–4 $\times 10^{16}$ cm^{-3}
DOFZ	2–3 $\times 10^{17}$ cm^{-3}	2–4 $\times 10^{16}$ cm^{-3}
MCz	0.5–1 $\times 10^{18}$ cm^{-3}	$\leq 1 \times 10^{16}$ cm^{-3}

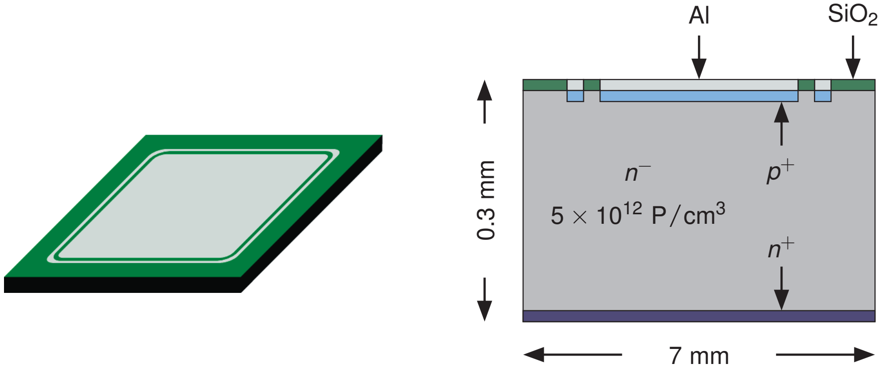


Figure 3.1: Illustration of the appearance and structure of the investigated detectors. The p^+ layers have a doping concentration of ~10^{19} cm^{-3} after ion implantation of boron and annealing. The illustration on the right is not to scale.

3.2 Hydrogenation

The first samples investigated (paper I) were hydrogenated at IMEC in Belgium where the samples were exposed to a remote deuterium plasma for up to 4 h. The motivation for using deuterium will be explained in chapter 4.

Wet chemical etching is a known technique for introducing hydrogen (see, for instance, Yoneta *et al.* 1991 [28] and Tokuda 1998 [29]). In an attempt to find a simple, in-house alternative to the plasma hydrogenation, we tried submerging samples in hydrofluoric (HF) acid diluted to 10%. This proved to be an effective method for hydrogenation.

The exact method that was used to introduce hydrogen with HF goes as follows:

- Dilute the 40% HF in water solution to a 10% solution in a plastic (of course) beaker. The recipe is 40% solution to water in the ratio 1:3 (gives 40 parts pure HF in the final solution containing $40 + 60 + 300 = 400$ parts).

- Heat a shallow water bath to ~60 °C on a hot plate. Monitor the temperature of this bath at all times.

- Let the HF beaker float around in the water bath. If it is unstable, there is too much water in the outer bath.

- Let the temperature stabilise for 10 min; by then the HF solution should be at 50 to 52 °C (once I checked this with water also in the inner beaker).

- Place the samples in the HF solution.

- The samples will often float around on the surface because of the surface tension or lie close to the beaker walls, so use a tool to move the samples around every now and then. Long protective gloves should be worn as there is surely considerable amounts of HF vapour in the air.
- Half an hour of sample stirring is, apparently, sufficient to get an amount of hydrogen that has considerable impact on the defect dynamics in the surface region.

Very little of the surrounding parameter space in this procedure has been explored. However, it does appear that submerging a sample in 10% HF at room temperature for 3 min introduces enough hydrogen to have a measurable effect on the annealing of V_2: After annealing at 275 °C there is a reduction in the V_2 concentration of $\sim$33% compared to the non-hydrogenated reference samples. Also, in the single experiment where it was tried, 10% HF at 50 °C for 5 min seemed to introduce no more hydrogen than 3 min at room temperature. However, no conclusions can be drawn from this without more experiments.

A major disadvantage of both plasma hydrogenation and HF dipping is that no usable contacts remain after the treatment, and new contacts must be created.

3.3 Connecting electrically to the samples

Achieving good electrical contacts after removing them with HF caused many months of hard work with much frustration and slow progress. Eventually, after one and a half years, I ended up with a method that was satisfactorily satisfactory. My procedure is currently as follows:

- With plastic tweezers (to avoid breaking off pieces at the edges), place the sample at a clean tissue paper.
- If the sample is unclean, soak the sample with acetone[1] and move it around on the tissue paper until it is reasonably clean on both sides.
- Clean the front side and back side of the sample with a cotton bud[2] soaked in acetone.
- Use cotton buds with acetone to clean the edges of the sample when resting it vertically on the table in the grip of the tweezers.
- If necessary, repeat the previous two steps.
- Whether the sample has a metallic back contact or not, make a silver-glue pad:

[1] Concerns have been expressed about using acetone, so some caution may be wise.

[2] Concerns have been raised also about this procedure as cotton buds tend to cause residue. Cleaning with a spray of acetone may be a cleaner method.

  - Apply a droplet of silver glue and smear it out. A cotton bud is a good tool for this also – just roll the stem of the cotton bud back and forth over the sample and let it smear out the silver glue.
  - When the glue is rather flat and almost dry, place the sample (with the pad in the making down) on a hard, flat surface, like a table, and rub the contact entirely flat. Don't apply too much force as the contact may become too thin.

- The sample should, ideally, have an evaporated aluminium contact.[3] If not, an emergency contact can be made with silver glue:
  - Make a dry, flat contact on the front side in the same way as for the back side.
  - With a cotton bud with a tiny amount of acetone, stroke the edges to take away the silver glue that is extending outside the guard ring. This step will surely leave behind visible or invisible paths of silver particles, so continue the stroking with new cotton buds for as long as necessary.
- Again, use cotton buds with a little acetone to clean the edges of the sample when resting it vertically on the table in the grip of the tweezers.

Similar care should be given when preparing the sample holder (in my case, usually the sample holder shown in figure 3.2):

- Clean the sample holder thoroughly with cotton buds and acetone. Do not use too much acetone as it will cause conductive particles to float around and be deposited in inaccessible places where there should not be electrical contact.
- To obtain a flat layer of silver glue on the sample holder, apply a droplet of silver glue and smear this out by rotating a metal rod with a flat end in the droplet until the silver glue becomes almost dry.
- The contact properties will continue to change as the silver glue becomes more and more dry. This process can be accelerated by heating the sample holder.

If there is a silver-glue layer with appropriate thickness and topography on both the sample holder and the back side of the sample, there should immediately be good contact when the sample is installed. In practice this is not always the case, so my procedure goes as follows:

- Insert the sample and lower the needle.
- Push a corner of the sample to rotate it slightly to search for a high-friction area.

[3] Aluminium seems to be the best for silicon – the Schottky barrier is low and the adhesion of the pads seems rather good. I tried one deposition of gold, but the adhesion was very poor. It is usually not worthwhile trying to maximise the contact area within the guard ring as there is always a risk of getting too close, and the contact should be sufficiently good even if it is just a moderately big spot in the middle.

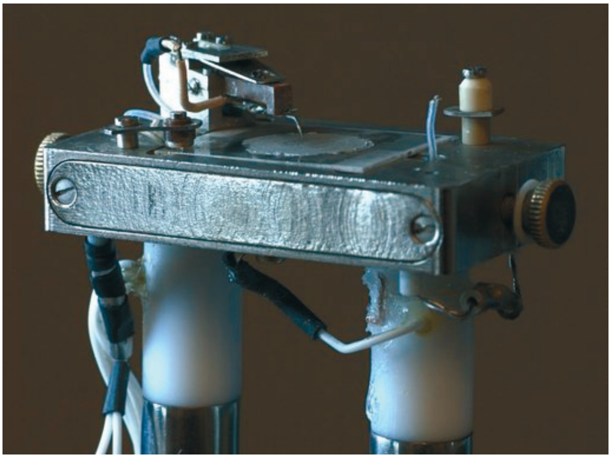


Figure 3.2: The sample holder of the Asterix set-up. Here, a thin pad of silver glue has been prepared (Photo: Klaus Magnus Johansen).

- Sequentially, put pressure on all the corners of the sample.
- Here, the needle may be lifted for a moment to release its tension.
- The capacitance should now, at room temperature, be (260 ± 20) pF for the 300-µm detectors I have been using. If it is not, even after a forward voltage of 5 V has been applied,[4] check if the silver glue pads are sufficiently thick and flat.
- Run a *CV* and *GV* (conductance versus voltage) measurement, and pay close attention to the conductance – the smaller the value, the better the contacts. A perfect sample of this type can go down to 2×10^{-6} S at a bias of -10 V, while a sample with silver-glue contacts may reach 1.2×10^{-5} S. Any conductance smaller than $\sim 4 \times 10^{-5}$ S at -10 V may give a good DLTS measurement.

An alternative route for reliable contact formation would be to clean the sample thoroughly and do a high-quality deposition of aluminium at both the front and back of the samples. The deposition should be done directly after the hydrogen introduction, while the native oxide is still thin. This route has, however, not been explored in my work.

A survey of the less successful contact methods investigated is given in table 3.2.

[4] The HP capacitance meter will anyway not give more than 1 mA.

Table 3.2: Methods to avoid (listed in chronological order). The results are ranged on a scale from A to F, where A is the currently best method, described in section 3.3.

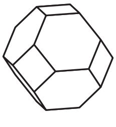
Front-side contact	Back-side contact	Result
The needle on a small, dry dot of silver glue	InGa (AKA GaIn)	C[a]
Needle directly onto the sample	InGa	E
Needle directly onto the sample with much additional pressure	InGa	C[a]
Front contact with a piece of flat copper between a silver-glue pad and the needle	InGa	B[a]
Silver-glue pad	A droplet of InGa	D[a]
A piece of indium	InGa	D[a]
A droplet of InGa with a piece of aluminium on top to protect the needle	InGa	D[a]
Silver-glue pad	A droplet of InGa distributed and scratched into the back surface	C

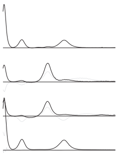
[a] I am not entirely sure about the uselessness of this method as limited attention was paid to the coverage of the back contact.

3.4 Experimental set-ups

The first annealing series were measured with the set-up labelled Tiffy (figure 3.3). This uses a Boonton 7200 capacitance meter and is cooled with a closed helium system. The helium system is sometimes controlled by a computer; the switch that allows this is described in section B.2. Some more details about the set-up are given by Mikelsen 2007 [25].

The Asterix set-up (figure 3.2) was used for most of the measurements. This is equipped with at HP 4280A capacitance meter. The samples are cooled by simply submerging the sample holder in liquid nitrogen. See Mikelsen 2007 [25] for a few more details. Section A.1 gives details on some of the programs on the Asterix set-up. Section B.1 gives some information about the temperature-controlling nitrogen lift.





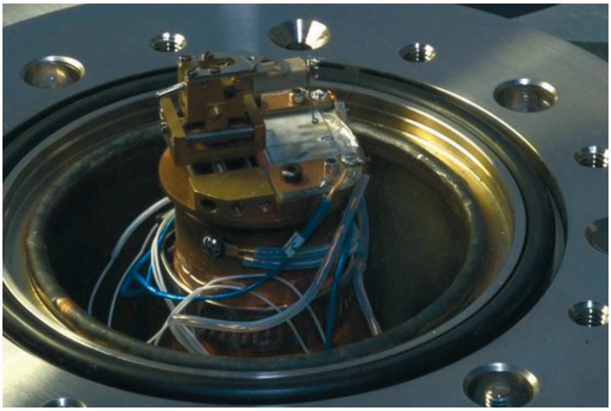
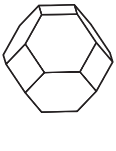
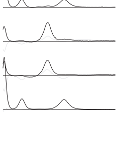


Figure 3.3: The sample holder of the Tiffy set-up (Photo: Klaus Magnus Johansen).





Chapter 4

Hydrogen-concentration measurements

Several of the samples in the present work have been exposed to deuterium rather than protium (ordinary hydrogen); this facilitates measurements of hydrogen–versus–depth profiles after the hydrogenation, because secondary-ion mass spectrometry (SIMS) is a factor of $\gtrsim$100 more sensitive to deuterium than to protium. The conditions used for the deuterium measurements were, most commonly, sputtering with 15-keV caesium-ions using a primary-beam current of $\sim$50 nA and a raster size of 150 µm and detection of negatively charged deuterium ions. The concentrations were calibrated with the use of a deuterium-implanted reference sample.

Figure 4.1 shows the SIMS profiles of the deuterium concentration measured from the front side (upper graph) and from the back side (lower graph) of four samples. These samples are plasma-deuterated DOFZ Si of the same kind as described in paper I. The samples were dipped in HF to remove the aluminium contact before they were deuterated under a pressure of 700 mTorr by remote plasma from the back side with the use of an OXFORD Plasmalab microwave system.

The deuterium profiles of sample 4-69 and sample 4-74, which are both deuterated for 4 h, display different behaviour: Before any annealing there is a substantial amount ($\sim$10^{17} cm^{-3}) of deuterium at the back side of both samples – somewhat more in 4-69 than in 4-74. After a 1-h anneal at 450 °C the deuterium concentration at the back side is slightly above the resolution limit in sample 4-74, but it is very low in all samples. At the front side, 4-74 has a deuterium concentration up to 10^{17} cm^{-3} before the anneal, while after the anneal, this has been reduced by about a factor of ten. Just a very thin layer of deuterium is measured at the front side of 4-69 before the anneal, and an even smaller amount after the anneal. This difference in the deuterium concentration at the front sides of 4-69 and 4-74 may be due to the mounting of the samples during the plasma deuteration; the results suggest non-homogeneous and unintentional exposure of the front surface of sample 4-74.

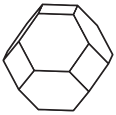
Sample 4-66 was exposed to the deuterium plasma for only half an hour. Similarly to 4-74, there is a considerable amount of deuterium at the back side of 4-66 which is undetectable after the 1-h anneal. The deuterium concentration at the back side of 4-66 is slightly less than that in 4-74, but the difference is smaller than that of 4-74 compared to 4-69, which are deuterated with the same duration. Similarly with 4-69, the deuterium concentration at the front side of 4-66 is very small and decreasing during the anneal.

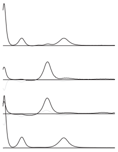
The last of the four samples in figure 4.1, sample 4-67, was deuterated for 1 h and not annealed. The front-side measurement shows that also this sample has received a substantial amount of deuterium from the front, but the profile is smoother compared to that of 4-74.

Figure 4.2 shows the deuterium profiles of two FZ samples submerged for half an hour in a 10% HF/DF solution (HF:DF:H_2O in the ratio 5:5:90, where DF is hydrofluoric acid with deuterium rather than protium) at 50 °C. Also, an aluminium contact was deposited onto the front side of the samples before the samples were annealed for 1 h in a nitrogen ambient – sample 7-28 at 450 °C and 7-39 at 250 °C. Both samples were then irradiated with 6-MeV electrons to a dose of 2×10^{12} cm^{-2}. Sample 7-39 was used for DLTS measurements and was exposed to 15-min anneals up to 150 °C before the SIMS measurements were undertaken.

Sample 7-28 is seen to contain a considerable amount ($\sim 10^{17}$ cm^{-3}) of deuterium directly below the aluminium contact down to a depth of $\sim$0.2 µm where the concentration reduces sharply. This is in sharp contrast to the profile measured at the oxide layer outside the guard ring where just a thin layer of deuterium is detected. A negligible amount of deuterium is measured at the back side of sample 7-28. The single measurement of sample 7-39 shows, like for 7-28, a presence of deuterium under the deposited aluminium contact, but with a concentration up to a factor of ten larger than in 7-28. This can readily be explained by the pre-irradiation annealing being performed at a lower temperature such that less hydrogen has diffused into the bulk or out of the sample. We have not performed SIMS measurements of non-annealed samples hydrogenated with this method.

We have, from these SIMS data, not been able to estimate the deuterium concentration in the bulk of any of the samples. No estimation can be made by extrapolating the surface concentration, either, since we do not have a model describing how the hydrogen diffuses from the boron layer into the bulk. The only thing we can say with certainty is that the bulk concentration is not above $\sim 10^{15}$ cm^{-3} in any of the measured samples. However, it is possible that more thorough investigations of non-annealed samples and samples annealed at temperatures below 450 °C could give deuterium profiles measurable also beyond the boron layer at the front side and the highly doped phosphorus layer at the back side.





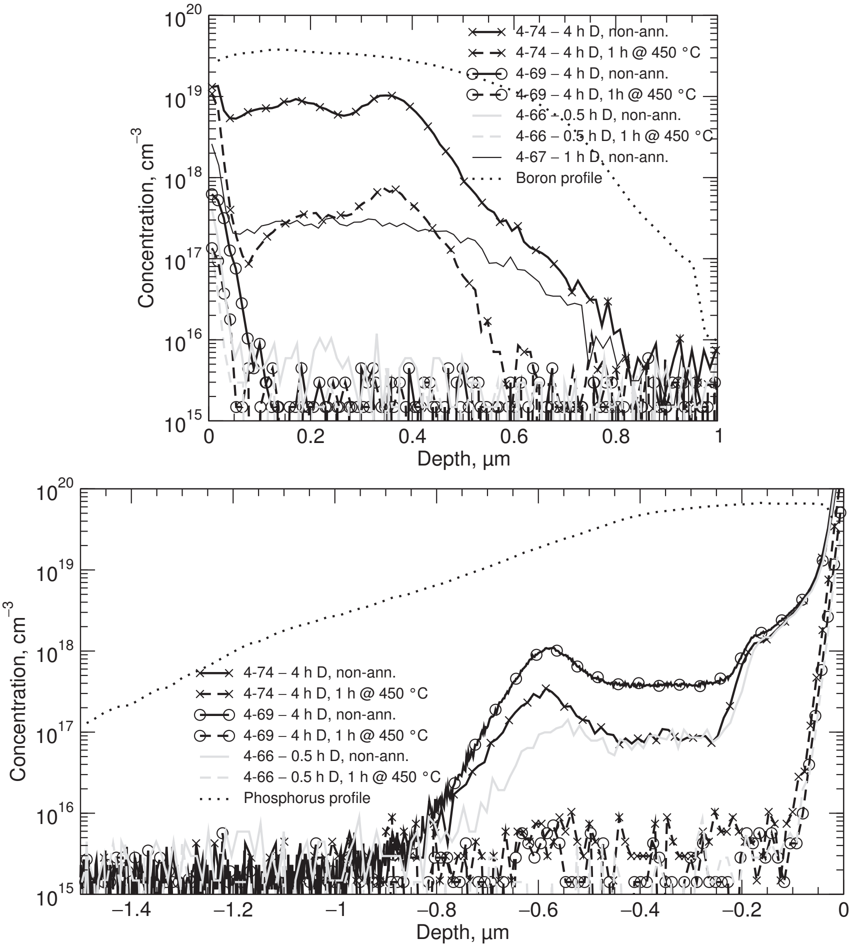


Figure 4.1: SIMS measurements of the deuterium concentration in samples exposed to deuterium plasma for 0.5, 1 and 4 h, respectively. For most of the samples, the deuterium profiles have been measured before (solid lines) and after (dashed lines) a 1-h anneal at 450 °C. The upper graph shows the deuterium concentration at the front side of the samples, while the lower graph depicts the concentration at the back side (the surface is at zero depth); the scale of the two is identical. The boron concentration and phosphorus concentration are also shown.

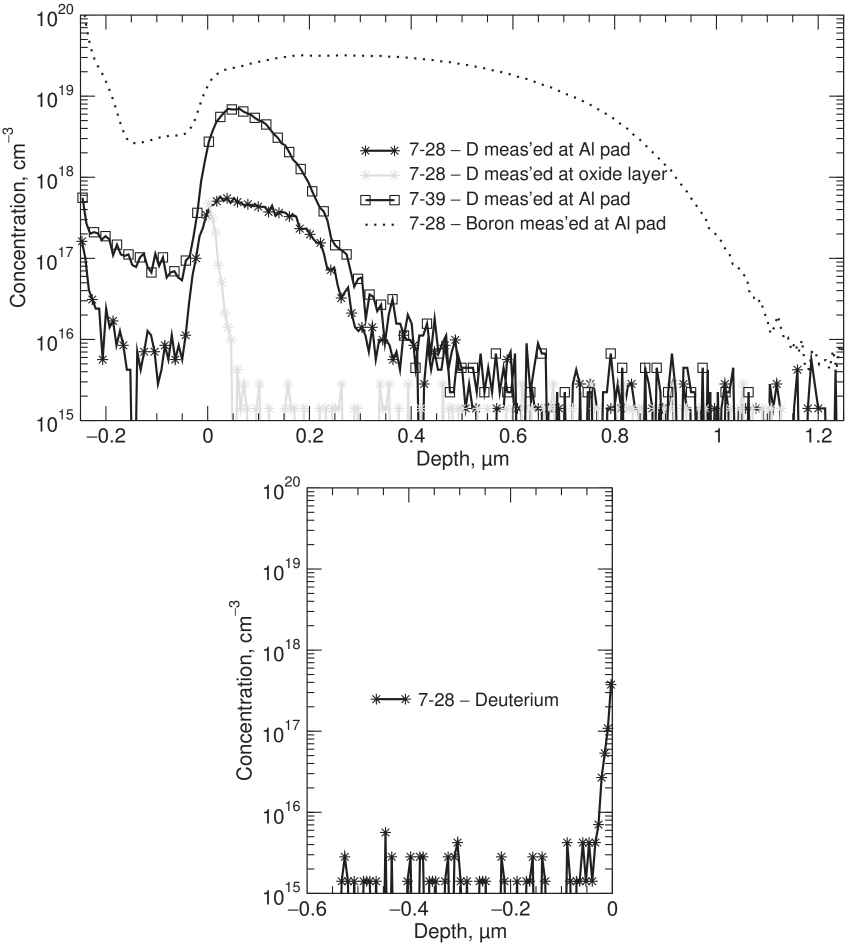


Figure 4.2: SIMS measurements of the deuterium concentration in samples submerged in a HF/DF solution for half an hour at 50 °C. Sample 7-28 was annealed for 1 h at 450 °C, while 7-39 for 1 h at 250 °C. The upper graph shows the deuterium concentration at the front side of the samples, while the lower graph displays the concentration at the back side (the surface is at zero depth). The front-side measurements conducted through the deposited 0.25-µm-thick aluminium contact have been shifted by 0.25 µm such that the silicon surface is at about zero depth. The boron concentration at the front side is also shown.

Chapter 5

Fitting and modelling

5.1 Fitting algorithms (general aspects)

The general algorithm used for fitting of the experimental data has remained rather unchanged throughout all of this work, using a method called ”simulated annealing”. This is an algorithm that searches for the global optimum by first making a parameter guess, then calculating a value describing the degree of success and accepting the new parameters with a probability depending on the error and a ”temperature” parameter (URL [30]). If the error is smaller than before, the new parameters will be accepted; if the error is larger, the new parameters will be accepted with a probability of $\exp\left(\frac{\mathrm{d}E}{T}\right)$, where d$E$ is the difference in error and T is the algorithm ”temperature”. The algorithm is started at a high ”temperature”, at which the input-parameter guesses are volatile, i.e. the guesses are made relatively far from the input-parameter values giving the currently most optimal fit. The ”temperature” is reduced with a certain rate down to a certain value, and if the ”temperature” is lowered sufficiently slowly, the global optimum will be reached; if not, the algorithm will end up in a local optimum.

The algorithm used is a slightly simplified version of the ”simulated annealing”, as I have let the volatility of the input-parameter guessing depend on the ”temperature”, but there is a zero probability of keeping the new parameters if they produce a result with a larger deviation than the currently best parameters.

5.2 Fitting of DLTS spectra

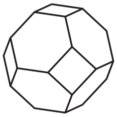
Two main approaches for extracting the activation enthalpy and apparent capture cross section of the various energy levels have been used. The first method is based on the assumption that the measured DLTS spectra follow the theoretical ones perfectly and that the capture cross section is temperature independent. The developed algorithm fits, simultaneously, a given number of DLTS peaks (calculated from three parameters: amplitude, activation enthalpy and apparent capture cross section) to the spectra of all the time windows of a set of measurements performed after different heat treatments – only the amplitudes are fitted independently for each spectrum.

The advantage of this method is that the algorithm has limited freedom to move the DLTS peaks around when trying to fit overlapping peaks. This may result in a realistic fit even of DLTS peaks that overlap strongly. An example of this is shown in figure 3 in paper I where the contribution of the di-vacancy–oxygen (V_2O) centre is separated from that of the di-vacancy (V_2) centre. The disadvantage of this routine is that it is so restricted by the massive amount of data that the the best fit normally does not reproduce each individual peak optimally. This may be caused by noise in the first windows, temperature-dependent capture cross sections, stray capacitance and series resistance in the apparatus, and limited accuracy of the temperature measurements.

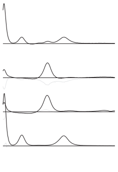
As I was particularly interested in the accurate amplitude of the DLTS peaks, this first method did not give satisfactory results. Therefore, a routine that fits each peak, or sum of peaks, in a particular spectrum, independently of any other spectra was developed. This effort evolved into the program described in section A.2. The main limitation with this program, is, apart from its user interface, that one must take great care when fitting overlapping DLTS peaks.

5.3 Modelling of reaction equations

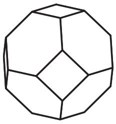
Paper VI presents a defect-reaction model accounting for the variation in defect concentration with depth which reproduces fairly well the measured annealing behaviour in hydrogenated samples for temperatures below $\sim$200 °C. Initially, an attempt was made to describe the annealing kinetics by a model only involving the different defect concentrations without taking into account the depth distribution; the corresponding program is described in section A.3. This was successful for the measurements obtained at the annealing temperatures 150 and 180 °C, but it was not possible to fit both data sets simultaneously with a consistent set of values for the reaction parameters. Moreover, the evolution of the DLTS amplitudes measured during the 195-°C annealing was difficult to simulate even by itself. Thus, realising experimentally that the defect concentrations exhibited a substantial variation as a function of depth and the unsuccessful modelling when assuming uniform defect distributions provided strong motivation for the introduction of a model addressing the measured depth profiles rather than the DLTS amplitudes. This proved successful in that it was possible to reproduced the annealing dynamics closely – from the point of view of comparison of the DLTS amplitudes calculated from the measured and simulated depth profiles. This comparison of DLTS amplitudes calculated from depth profiles was performed as a first test to see if the result looked better than when modelling depth-independent defect reservoirs. Since the result looked satisfactory, it was considered meaningful to rather optimise the fit of the simulated depth profiles to the measured depth profiles. In the simulations, the Matlab function `pdepe` is used for solving the differential equations numerically – more technical details can be found in section A.4.

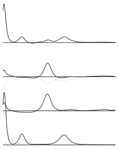


Several different defect-reaction models, including two hydrogen sources, various



shapes of the initial concentrations both of the hydrogen and of the hydrogen traps, were evaluated. It seemed not possible, however, to reproduce the experimental data without introducing a small effective hydrogen-capture radii for both the VO centre and the V_2 centre, indicating the presence of reaction barriers as further discussed in paper VI.





Chapter 6

Summary of results

6.1 Paper I

This paper presents the results from an isochronal annealing study with deuterated DOFZ samples. The reaction where VO captures a hydrogen atom and forms vacancy–oxygen–hydrogen (VOH; or VOD in this case) is prominent – especially for the sample exposed to the deuterium plasma for the longest time. This sample also shows a substantial concentration of two presumably hydrogen-related levels at $E_c - 0.17$ eV (labelled E2) and $E_c - 0.58$ eV (labelled E3), respectively. In the sample deuterated for only half as long, the E1 level, at $E_c - 0.37$ eV, reaches a higher concentration than that of the VOH centre. Also one of the non-hydrogenated reference samples shows an annealing behaviour indicating the presence of some residual hydrogen.

A curiosity worth mentioning is that the H1 level (see paper V), at $E_v + 0.23$ eV, is clearly visible in the results obtained with the Boonton set-up (figure 1); we just did not realise at that time that this was the signal from a real charge-carrier trap.

6.2 Paper II

This study shows that the V_2O centre (which was in most cases formed by a 15-min anneal at 300 °C) anneals out via dissociation which results in a VO centre. The binding energy between V and VO in the V_2O complex is estimated to be $\sim$1.7 eV. The VO centre anneals through migration and capturing by an oxygen atom. In oxygen-lean material, the VO is captured at such a low rate that there is an initial increase in the VO concentration when the V_2O centre dissociates; this increase is not seen in oxygen-rich DOFZ material. The evolution of the measured defect concentrations is closely reproduced by the proposed model. Hydrogen is included in the model such that it reproduces also the observed concentration of the VOH centre. The results suggest that VOH is formed when monatomic hydrogen reacts with VO, and also that this complex dissociates in the opposite reaction.

6.3 Paper III

What is considered to be one defect with two energy levels in the upper half of the band gap – E4 at $E_\mathrm{c} - 0.37$ eV and E5 at $E_\mathrm{c} - 0.45$ eV, which exhibit a close one-to-one proportionality with each other – is found to anneal out with first-order kinetics during a few weeks at room temperature. It is argued that the defect is fundamental, intrinsic and extended in space. It is speculated that a {110}-planar tetravacancy chain (V_4) might be a defect with these properties. V_4 is also known to have limited thermal stability.

6.4 Paper IV

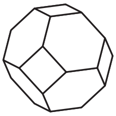
An isothermal annealing study performed with DLTS shows that the annealing kinetics of the E4/E5 centre is very similar to the annealing kinetics of the 936-cm^{-1} band measured by Fourier-transform infra-red spectroscopy (FTIR) even though the samples were irradiated with different particles to very different doses, resulting in very different defect concentrations. Since the 936-cm^{-1} band is previously reported to be a di-interstitial–oxygen (I_2O) complex, it is suggested that the E4/E5 centre and the I_2O centre are identical. The combined results from DLTS and FTIR give an activation barrier for the annealing of 1.1 eV with a pre-factor of 9×10^{12} s^{-1}, and it is argued that the defect is not annealing by migration, but rather by dissociation or changing atomic configuration. E5 is previously shown to have a strong correlation with a change in the leakage current of silicon detectors, and it is, therefore, argued that one should reduce the concentration of oxygen and carbon to reduce the leakage current in silicon devices.

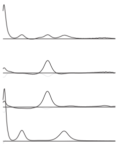
6.5 Paper V

Strong evidence is given that monatomic, positively charged hydrogen diffuses in from the surface layer at temperatures below 200 °C – probably involving dissociation of the boron–hydrogen (BH) centre – and reacts with the VO centre to form a meta-stable VOH* centre. This centre has an energy level at $E_\mathrm{c} - 0.37$ eV and dissociates at longer annealing times. The V_2 centre also reacts with the in-diffusing hydrogen; this reaction probably causes the V_2H centre to be generated. The growth of the hole trap H1 at $E_\mathrm{v} + 0.23$ eV displays a close one-to-one proportionality with the loss of the V_2 trap, and it was therefore suggested that H1 is an energy level arising from the V_2H centre.

6.6 Paper VI

This paper presents a model which reproduces closely the depth profiles from paper V measured during annealing at 195 °C where monatomic hydrogen diffuses in from the surface and causes the transformation of VO into VOH* and the annealing of V_2, before VOH* dissociates and VO is restored. Profiles measured during annealing at 225 °C for up to 11 days reveal that continued annealing after the VOH^* centre has annealed causes the VO centre to start annealing again; here, VO is not replaced by VOH^*, but rather by VOH. However, the concentration of the VOH centre is too small to account for the loss in VO; this is explained by VOH being transformed into a vacancy–oxygen–di-hydrogen (VOH_2) centre. This is further supported by the depth profile of the VOH centre having a reduced concentration both towards the bulk and towards the surface of the sample – a clear indication that VOH is passivated by something diffusing in from the surface.





Chapter 7

Concluding remarks

A main achievement of this work has been to identify the energy level at $E_\mathrm{c} - 0.37$ eV as a vacancy–oxygen–hydrogen complex, labelled VOH* in this work, which is different from the VOH centre that has energy levels at $E_\mathrm{c} - 0.32$ eV and $E_\mathrm{v} + 0.28$ eV (paper V). The observation that the H1 level, which we argue can be identified as V_2H, has a one-to-one correlation with the energy levels of V_2, may also prove valuable for future studies of hydrogen in silicon (paper V). Further evidence has been given that VOH anneals out by capturing a hydrogen atom to form VOH_2 (paper VI). There are also indications that this breaks up and restores the VO centre at ∼325 °C.

Several other hydrogen-related levels have been observed; speculations on their identity have been put forward for some of them, but a satisfying identification of all the levels is still lacking. Further investigations are clearly needed to reach a complete understanding of how hydrogen affects the defect dynamics in silicon. It would be interesting to perform more annealing studies of hydrogenated samples with different oxygen concentrations to check if some of the speculations in paper VI can be verified or disproven. For instance, whether the E2 level and E3 level are affected by the oxygen concentration and whether they *always* appear with the same concentration.

There was a hope that the present work would clarify under which conditions the hydrogen diffuses and reacts as monatomic and under which conditions it is molecular or maybe a more loosely bound dimer; it seems, however, that all the results that were obtained can be explained by monatomic hydrogen alone, and little knowledge has been gained regarding when molecular hydrogen is present. However, there may be a possibility of checking if molecular hydrogen is passivating V_2 if the assumption that the level labelled H1 is indeed the result of the reaction

$$V_2 + H \rightarrow V_2H. \tag{7.1}$$

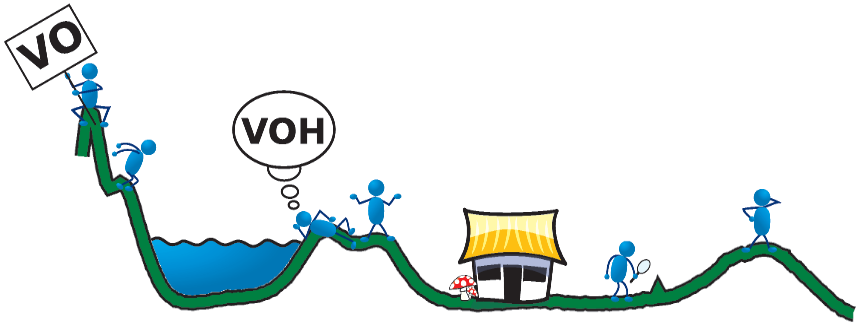
Then, if VO and V_2 are passivated by molecular hydrogen, the H1 level should not appear.

Increased understanding of how hydrogen affects the defect dynamics has been the main aim in this work. A potential application is fabrication of more radiation-hard detectors with the use of hydrogen. The $V_2(-/0)$ level can, in the presence of positively charged hydrogen, be passivated by annealing at temperatures below 200 °C (paper V), but even if this level is as deep as $E_c - 0.43$ eV, it is not considered particularly harmful to the detector operation. However, the hydrogen-related E3 level at about $E_c - 0.58$ eV (see paper VI) is, due to its position, likely to be an effective generation centre which increases the leakage current. Identification of this level might, thus, be particularly useful.

I put much effort into the investigation of the E4/E5 defect (see paper III and paper IV). This centre may be particularly important as it has been shown by Moll *et al.* 2002 [31] that the annealing of E4/E5 may be strongly correlated to a reduction in the leakage current of irradiated diode structures. After I had completed an isothermal annealing study of the E4/E5 annealing with the use of DLTS, it was discovered that the annealing behaviour of what is from FTIR expected to be a I_2O centre in samples with entirely different properties is strikingly similar to the annealing behaviour of E4/E5. This resulted in paper IV, where we argue that the E4/E5 centre may be identified as a I_2O complex.

Regarding the E4/E5 defects, it should certainly be made an attempt to reproduce the generation of what seems to be these levels by applying a large forward-current density, as performed by Fleming *et al.* 2007 [32]. If this is successful, it should be very easy to determine with FTIR whether the E4/E5 defect is identical to the I_2O centre measured by FTIR, as was suggested in paper IV. However, a current density of 12.5 A/cm^2 will surely heat our samples substantially and possibly also damage them.

DLTS spectra are never ideal; various deforming of the DLTS peaks is introduced due to effects from the apparatus and from emission parameters which may be temperature dependent. Therefore, another possible investigation would be to find if it is possible to include a temperature-dependent apparent capture cross section in the fitting routine as this changes the shape of a DLTS peak. One should always take care when increasing the freedom of a fitting algorithm, but the first fitting algorithm described in section 5.2 has so little freedom that one could possibly introduce both temperature-dependent capture cross sections and parameters for set-up artefacts (e.g. temperature offset and baseline offset) and still get trustworthy results.



Appendix A

User guides on software

A.1 Asterix how-to

A.1.1 *CV* measurements

As was mentioned in section 3.3, it is a large advantage to pay attention to the conductance of the sample. I therefore included in the *CV* program a graph showing the conductance versus the voltage. I also included output terminals for the values of the starting and ending conductance and the ending capacitance, such that I can save these parameters as a characterisation of the contact. I also made a graph showing $\frac{1}{C^2}(V)$ and the deviation of this from a straight line.

Figure A.1 shows a screenshot of `CVmeas with G.vi`.

It should also be mentioned that the HP 4280A capacitance meter performs *CV* measurements at a probe frequency of 1 MHz.

A.1.2 DLTS measurements

DLTS measurements have been performed extensively, with the natural consequence that the DLTS program (figure A.2) has undergone several modifications. The first and most important change was, for obvious reasons, to make the program save the data continuously rather than after the `Stop` button has been pushed.

The most noticeable (and by far the most time consuming) change to the program was implementation of the automatic temperature control. The automatic control is activated by `Autolift(TM)`. The `Heat`/`Cool` switch is of utmost importance to achieve the desired result. The heating/cooling rate is set in `#scans setpoint`. This is a rather rough estimate of how many transient recordings the capacitance meter will manage during a 1-K interval if the longest time window is $(640\text{ ms})^{-1}$ (if `# Windows` is 6). If longer time windows are used, the temperature rate will still be similar, but `#scans setpoint` and `Estimated number of scans` will no longer refer to the actual number of transient recordings during a 1-K interval. The most common use of the temperature control is to just **Disable** the **lift when** the temperature is outside the given range; this is to avoid damage to the stepper motor (see section B.1 for more). The temperature will usually go down to liquid-nitrogen temperature (currently ∼78.1 K)[1] when the lower disabling limit is set to the default 80.1 K. The higher disabling limit should not be set above 0 °C unless the temperature rate is very low or the heater power is more than ∼30%, as melting of the ice dramatically slows down the temperature rate. **Faster lift** is just an option that causes the Lift to be triggered twice rather than once if the temperature rate is far too low.

A more exotic feature is the one that will **Turn** the **lift** (or rather the sign of the temperature rate). This function will, when the lower temperature limit is reached, flip **Cool** to **Heat** and continue measuring with increasing temperature. Make sure to disable **Turning disabled** if using this feature during more than one measurement. Possibly even more exotic, yet, is the possibility to make the program calculate the temperature rate as a given function of the current temperature. This was implemented in a period when the Asterix set-up was very popular and it seemed like a good idea to scan quickly at the temperatures where there are no DLTS peaks. The default example,

```
r=s+s*0.7*exp(-(t-80)^2/(2*10^2))+→
→s*0.7*exp(-(t-114)^2/(2*20^2))+→
→s*0.7*exp(-(t-196)^2/(2*20^2))-→
→(1+sign(t-210))*0.5*(t-210)/15,
```

where **s** is the value in **#scans setpoint** and **t** is the current temperature, describes a uniform number of scans per temperature interval (given by the first **s**) with the addition of Gaussian peaks, with the height of 70% of **#scan setpoint**, at the positions of VO, V$_2$(=/−) and V$_2$(−/0). This will cause the temperature rate to slow down in these areas. If the temperature is above 210 K (**(1+sign(t−210))*0.5** $= 1$) and increasing, the heating rate will be set higher and higher (**r** lower and lower). A negative **r** does not cause errors, but is not different from **r** being zero.

Save each transient is a feature I always enable; it causes the generation of an additional **.trans** file which contains each single transient recorded, preceded by the temperature at that point. I have a special liking for large amounts of data, so many users can live happily without this feature.

Autostop is very useful when performing overnight measurements with all the transients being stored – the difference is tens of Mb. The feature also reduces the possibility of the stepper motor being stuck for eternity with a task it can not perform (such as lowering the lift through the table).

I have added a boolean variable **Update graphs** which, when disabled, allows the user to zoom in on the spectra without being disturbed by the auto-scaling. It has also become possible to show the DLTS spectra extracted by a range of time windows by changing the variables **Window** – being the first window to display – and **Display # windows**.

[1] According to the Asterix set-up.

A peculiarity worth noting is that when **High Resolution** is enabled (this is a feature of the HP 4280A capacitance meter), the **C-range** setting has no effect – it is automatically the most sensitive setting (10 pF).

The currently newest program is **DLTS_Asterix_current_version.vi**. This is a copy of **DLTS_Asterix_20070215r.vi**. A change log can be found in **DLTS_Asterix_current_version.txt**.

A.1.3 Depth-profile measurements

The first changes I made to the profiling program was to include a routine that would control the Lift and such keep the temperature constant. This has some drawbacks, because when the profiling is finished, the temperature will no longer be kept constant. This was solved by keeping the temperature-control routine in a separate program that always runs in the background: **stand-alone_PID_control_current_version.vi** (section A.1.4). Thus, the only change I have made to the current program (**Profiling_semitrap_with_dS_vs_dV_with_stopnow.vi**) is to include a graph showing $\frac{\mathrm{d}S}{\mathrm{d}V}$; this gives a rough picture of the profile and is helpful when determining whether the number of averages is large enough to get a smooth profile. There is also a button **Stop now** which finishes the profiling and quits even if the temperature is not within range. The last voltage sweep is then not included in the average. Figure A.3 shows a screenshot of the program.

The depth-profiling program would benefit from the possibility of changing the voltage in a quadratic manner such that the concentration is plotted with equidistant depth intervals. This requires, however, a substantial reprogramming, because instead of instructing the capacitance meter which voltage to use for each transient, it is instructed to change the voltage stepwise in an interval.

A.1.4 PID control that keeps the temperature constant

This program (**stand-alone_PID_control_current_version.vi**) moves the Lift (section B.1) up and down in order to keep the temperature constant. In the beginning I made this program an integrated part of the depth-profiling program (see section A.1.3), but I later realised that several programs can be run simultaneously, so I took it out as a separate program to be run together with the depth-profiling program or the capture-cross-section program.

The program is rather tedious and complicated to use, so I will just leave it to evolution to decide if the program is worthy of the operating instructions being transferred from one generation of Ph.D. students to the next. I have made an attempt to write a user guide in the program itself.

The biggest disadvantages of the program are, apart from the non-intuitive user interface, that it usually takes 10 to 30 min, or more, before the program is able to keep the temperature constant, and the program takes up non-negligible run time (hence, the measurements are performed slower). On the positive side, the program can, on a very good day, keep the temperature constant with a deviation of no more than 0.05 K until the nitrogen boils away.

A.1.5 Temperature logger

I find it very useful to have the temperature history whenever I try to stabilise the temperature. **`temperature_logger.vi`** is a simple program that provides this.

A.2 DLTS-transient analyser

This program (screenshot in figure A.4) has evolved through many generations; one later generation was even started again from scratch, but this has not prevented several quick and dirty implementations of solutions and features. On the positive side, the program does have some useful features. The program is made for running through a data set of any size and I believe, therefore, that given a sufficiently large number of DLTS measurements, it is worth the trouble of setting all the parameters to make the program run.

A.2.1 Feature overview

- Load standard DLTS files (data files containing transients averaged within a fixed temperature interval, and the reverse capacitance)
- Load transient files (these are optionally generated by the Asterix set-up and contain each single transient recorded and the temperature at which it was measured)
- Generate DLTS spectra with any user-defined weighting function, although currently only stepwise constant functions (lock in, GS4 and such)
- Average the temperature and amplitude of points in the DLTS spectra closer to each other than a given temperature interval
- Save generated DLTS spectra as a standard DLTS file (similar to the output files from the Asterix set-up)
- Fit any part of the DLTS spectrum by any assumed number of overlapping peaks to extract the amplitude, activation enthalpy and apparent capture cross section
- Average the Arrhenius plots of several weighting functions, windows and files to enlarge the amount of data before calculating the average activation enthalpy and apparent capture cross section
- Make a graph showing the evolution, for instance, from annealing step to annealing step, of the amplitude of a set of DLTS peaks

- In case of a DLTS peak with changing amplitude being overlapped by one having constant amplitude, subtract each transient of one data file from the closest (in temperature) available transient in another data file before generating the DLTS spectra and performing the analysis
- Make a graph showing, for each weighting function, activation enthalpy versus apparent capture cross section to see how the results make a line
- Save **`.png`** files of all the generated spectra and each best fit of the chosen DLTS peaks to be able to check what has happened during an overnight analysis
- FFT smoothing of the spectra (not working)

A.2.2 Description of each of the features

The present section will discuss some of the most important settings in the input file (which, at present, should bear the name **`inputdata_temp.m`**). Appendix C shows an example of an input file for this transient-analysis program.

Load standard DLTS files

Loading the transients correctly from a DLTS file is maybe what is the most difficult with this program and usually the reason for all the mysterious error messages.

All the file names of the DLTS files (containing temperature interval, reverse capacitance, spectra and transients) should be listed in the variable **`name`**, where

`name=char('file_1.dat','file_2.dat',`[...]**`,'file_n.dat')`**.

The path of the files goes into **`path`**. The program must be told which format the files in **`name`** have. This goes into the variable **`format_of_files`**. This is a 2-by-*n* matrix, where *n* is the number of data files. The first row has value 0 if the file is a transient file (containing only the temperature and transient), 1 if it is a last-generation standard DLTS file from the Asterix set-up (contains temperature, number of scans at that temperature, reverse capacitance, pulse capacitance, lock-in spectrum, GS4 spectrum and transient), 2 if it is an older standard DLTS file from Asterix (like 1, but without the GS4 columns). The second column should contain the number of windows stored in the data file (six with default settings).

If one of the two numbers in **`format_of_files`** does not describe correctly the corresponding input file, there will be problems. The program may go on without error messages, but the spectra will usually look special; they may, for instance, increase almost monotonically with temperature.

As said, the strength of this program is that it can handle large sets of data. Microsoft Windows may be time consuming – certainly not less so when it comes to performing repetitive tasks on a large number of files. To load 1000 DLTS files, it is necessary to write 1000 file names in the variable **`name`**. On a POSIX system this can easily be achieved by, for instance,

```
ls *.dat|tr '\n' ','|sed 's/,/'\'','\''/g'|→
→sed 's/\(.*\),'\''$/name=char('\''\1);/g'<enter>
```

which produces the required

```
char('1000_files_of_transients_on_the_disk_to_load.dat',→
→'999_files_of_transients_on_the_disk_to_load.dat',[...]);
```

which can just be pasted into the input file. Information from **Crname** is not used, since the reverse capacitance is contained in the standard DLTS files, and information from **outname** is not necessary unless the resulting spectra are to be written to disk (described below); despite this, both variables must be declared (just do a **variable=[];**).

Variables in the input file relevant for this feature:

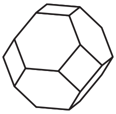
```
path
name
format_of_files
```

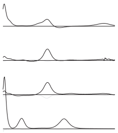
Load transient files

What I refer to as transient files are additional data files from the DLTS program (the option is presently only available at the Asterix set-up) which contain each single transient recorded, preceded by the present temperature. My motivation for implementing this was my liking for large amounts of data and avoiding the negative consequences of averaging: with a varying temperature rate it is not correct to use the temperature in the centre of the interval, with averaging it is difficult to discover incorrect transients and un-averaging is difficult.

As with standard DLTS files, the variable **name** lists the file names, but now **.trans** files instead of **.dat** files. The **.trans** files do not contain the reverse capacitance; each of the transient files must, therefore, be coupled with the standard data file containing these values. The **.dat** files corresponding to the **.trans** files are listed in **Crname**, but not necessarily in an ordered one-to-one pattern as **belongs_to_name_index** maps the **.dat** files to the **.trans** files.[2] Like for the **.trans** files, **format_of_Cr_files** contains the format and number of windows of the **.dat** files. The format should not be 0, since a transient file does not contain reverse capacitance (see previous section on loading standard files).

As I tend to use long file names, analyse many measurements and, besides, do anything to avoid repetitive tasks, I have written a bash script for generating the three input variables **name**, **Crname** and **outname** from a set of files. This is called **choosefiles**. My system is to have a folder with all measurements, e.g. **Sample_A**. In this folder I have a subfolder **Best** with a symbolic link to the standard DLTS files of all the best measurements in the parent folder. Then, with **Best** as the working directory, I run **choosefiles .. *.dat**. This will give three Matlab lines loading all the





[2] Right now this does not seem to make much sense to me, whereas the opposite *could* have a mission.

.dat files (if ***.dat** is used) into **Crname** and the corresponding **.trans** files into **name**. Also, a corresponding list of names for output files will be placed in **outname**.

I have mostly used the transient-analysis program with transients from the transient files rather than the standard averaged data files – I therefore expect the former method to have less bugs than the latter.

Variables in the input file relevant for this feature:

```
path
name
format_of_files
Crname
format_of_Cr_files
belongs_to_name_index
```

Generate DLTS spectra with any user-defined weighting function

DLTS spectra will be calculated with the weighting functions in the cell structure **weights** and only those. For instance, **weights** may contain six time windows of a lock-in type weighting function. Then

```
weights{1}{1}=[-1 1]
weights{1}{2}=[-1 -1 1 1]
weights{1}{3}=[-1 -1 -1 -1 1 1 1 1]
⋮
```

all the way to **weights{1}{6}** which contains 64 elements. The six first time windows of the lock-in function can be loaded into the first cell of **weights** by simply giving the command

```
weights{1}=binary_pattern2dlts_weight([-1 1],64).
```

The term binary is somewhat misleading as the first five time windows of the GS4 function can be loaded by, for instance,

```
weights{2}=binary_pattern2dlts_weight([1 -25 48 -24],64).
```

I did not have sufficient need of using anything else than stepwise constant weighting functions and have not programmed anything more sophisticated than this; if sine functions and such are desired, the routine **binary_pattern2dlts_weight** would have to be replaced by something loading **weights** with sine values in arrays of different lengths. I do not foresee any reason why this should not be trivial.[3]

transientlength is automatically taken as the number of data points in the transients of the measurement with the fewest windows. Sometimes it is desirable to

[3] Then again, I rarely foresee any obstacles before being halfway into the programming task.

override this if only the spectrum in one window is fitted; for instance, VO is not so easily fitted in the $(2.56\ \mathrm{s})^{-1}$ window (window 8 for lock in), so **`transientlength=2^6`** may be a good idea in order to discard windows 7 and 8 (lock in).

Variables in the input file relevant for this feature:

```
weights
plot_spectra=1
divide_by_Cr=1
transientlength
```

Average points in the DLTS spectra closer than a given temperature

All DLTS points are sorted, and this algorithm starts with the data point with the lowest temperature, makes an average of all the points closer than **`average_interval`** (K), and then repeats this process for the first point with a temperature outside this interval (only points with higher temperatures are included in the average). If no averaging is wanted, use **`average_interval=0`**.

Variables in the input file relevant for this feature:

```
average_interval
```

Save generated DLTS spectra as a standard DLTS file

If this option is enabled, the generated spectra will be saved in a data file that is filled up with numbers (the values for the pulse capacitance and conductivity are currently ignored and just set to zero) to resemble a last-generation data file from the Asterix set-up. The program will give a warning unless **`divide_by_Cr=1`**, because in the standard DLTS files, the spectrum is not divided by the reverse capacitance. I suspect that the current algorithm will fail if **`weights`** contains more than one weighting function, i.e. if **`length(weights)>1`**.

Variables in the input file relevant for this feature:

```
write_outfiles=1
outname
divide_by_Cr=0
```

Fit any part of the DLTS spectrum

Here lies the core function of this program. With the fitting procedure enabled (**`do_arrhenius=1`**), the program will fit the calculated spectra with an ideally-looking synthetic DLTS peak or a combination of several of these. When an optimal fit is found, the temperature position of the peak and its amplitude is kept. If the weighting function (in **`weights`**) has more than one window, an Arrhenius plot is made, from which the enthalpy and apparent capture cross section are extracted.

The algorithm will search for peaks defined in the cell structure **dlts_peak_guesses**. Each cell will cause a best-fit search in one particular area in the spectra of all windows of all weighting functions. For instance,

dlts_peak_guesses{1}=[5.1e-14 0.1817],

where the first value is the apparent capture cross section and the second the activation enthalpy, will result in an attempt to fit one synthetic DLTS peak somewhere close to the VO peak (these values are chosen empirically and for the $(640\ \text{ms})^{-1}$ window in this case). If there are overlapping peaks in the spectra, it may be required to fit simultaneously with two or more synthetic DLTS peaks.

dlts_peak_guesses{2}=[1.1418e-14 0.37252;3.1e-15 0.429]

will lead to a simultaneous fitting of the peaks of VOH* and $V_2(-/0)$. Fitting two overlapping peaks is difficult – fitting three and more is hazardous as it opens for a lot of different interpretations.

The variable **range_to_include_beside_outermost_peaks** states how large part of the spectrum to include when fitting the set of peaks given in a single cell of **dlts_peak_guesses**. The variable name is meant to mean how many Kelvin of the spectra to include on either side of the temperature positions calculated from the values in **dlts_peak_guesses**. If one of these cells tells to fit two peaks at 165 and 196 K, respectively, **range_to_include_beside_outermost_peaks=10** will cause all the data in the range 155 to 206 K to be included when calculating the error of the two synthetic peaks. It is difficult to give advice here, because including a large temperature range gives more data and more accuracy, but the exact maximum of the peak will often be in the wrong position with a wrong amplitude, while fitting a narrow range of data gives the correct peak amplitude, but more noise and uncertainty (especially for low concentrations). **range_to_include_beside_outermost_peaks** sets the range for all the fitting sets in **dlts_peak_guesses** while **T_range_offset** can be used to alter this individually. Each cell of **T_range_offset** corresponds to a cell of **dlts_peak_guesses**. If **T_range_offset{1}=[-3 5]**, the range of data will be increased by 3 K on the low-temperature side of the first fitting set and increased by 5 K on the high-temperature side. The range of measurement data included for each of the fitting sets is marked, in the graphs showing the spectra, by vertical lines connected by a horizontal line.[4]

When it is defined in which temperature range to search for which energy levels, it remains to define *how* to search. For each synthetic DLTS peak, there are three parameters: amplitude, apparent capture cross section and activation enthalpy. I realised that it is more effective to optimise the peak position and peak width rather than the capture cross section and enthalpy. Therefore, in the fitting routine, the parameters that are being optimised are the peak amplitude, the peak width and the peak position. Despite

[4] Or, to be more specific: A horizontal line at the average of the amplitudes of the DLTS points lying closest to the peak positions calculated from the defect properties in **dlts_peak_guesses**.

of this, the old variable names remain, and the variables **amplitude_volatility**, **sigma_volatility** and **energy_volatility** state how much to change the amplitude, width and position, respectively, in each fitting attempt. These values are standard deviations of the distribution functions used to pick random values for the fitting attempt. Which distribution function to use for each of the three parameters is given in **guessmethod_init**. Gaussian distribution (0), gamma distribution (1) and uniform distribution (2) are currently supported. I have been using the gamma distribution for the amplitude and peak width and the Gaussian distribution for the peak position. Not all parameters will be changed in each fitting attempt; the probability of any parameter being changed is **chanceofchange** (currently 60%).

The three volatility parameters are multiplied by a fitting temperature. This temperature starts high and is reduced as the fitting proceeds. In this manner, the synthetic DLTS peaks move violently around at first, while smaller and smaller changes are tested towards the end of the fitting of that particular set of peaks. The initial fitting temperature is given in **T_init** and the fitting temperature at which to stop is given by **T_end**. After **iterations_in_fitting_function** attempts to find better values for the three (unless there are overlapping peaks) fitting parameters have been made, the fitting temperature is multiplied by **T_change** (which is less than 1). These four variables together define how many fitting attempts to make for each point of interest in each window.

It is nothing short of terribly difficult to find good values for the volatility and fitting temperature – i.e. to find an effective (in terms of CPU time) way to reach the optimal fit. Increasing the value of **plot_frequency** (e.g. to 4) may simplify the task as the fitting window will then be updated more frequently. It is then possible to see how far away the guesses are from the currently best fit, and if any of the three fitting parameters are varied too little compared to the others.

When a fitting starts, only the amplitude of the peak(s) will be optimised such that the fit will be roughly correct (if the values in **dlts_peak_guesses** are good) before all the parameters are let loose. The internal parameters **number_of_cool_steps_in_amplitude_optimisation** and **iterations_in_amplitude_optimisation** determine how much time to spend for this initial process.

Sometimes the fitting algorithm will print a lot of asterisks (*****) – this is a warning that one of the fitting parameters (usually the peak width) has been chosen very close to the illegal limit in the gamma distribution (usually zero) and that the guess value will be forced away from this limit. This frequently happens if the fitting routine tries to move the base line up (or down if **allow_negative_amplitude=1**) by having one very broad peak. If a peak is chosen sufficiently broad, it will cause a division by zero.

It is also possible to fit minority-carrier traps; this requires **allow_negative_amplitude=1**.

Variables in the input file relevant for this feature:

```
do_arrhenius=1
dlts_peak_guesses
```

```
T_init
T_end
range_to_include_beside_outermost_peaks
T_range_offset
weights
window_offset
iterations_in_fitting_function
errorlimit
T_change
plot_frequency
amplitude_volatility
sigma_volatility
energy_volatility
guessmethod_init
allow_negative_amplitude
```

Average the Arrhenius plots of several weighting functions windows and files

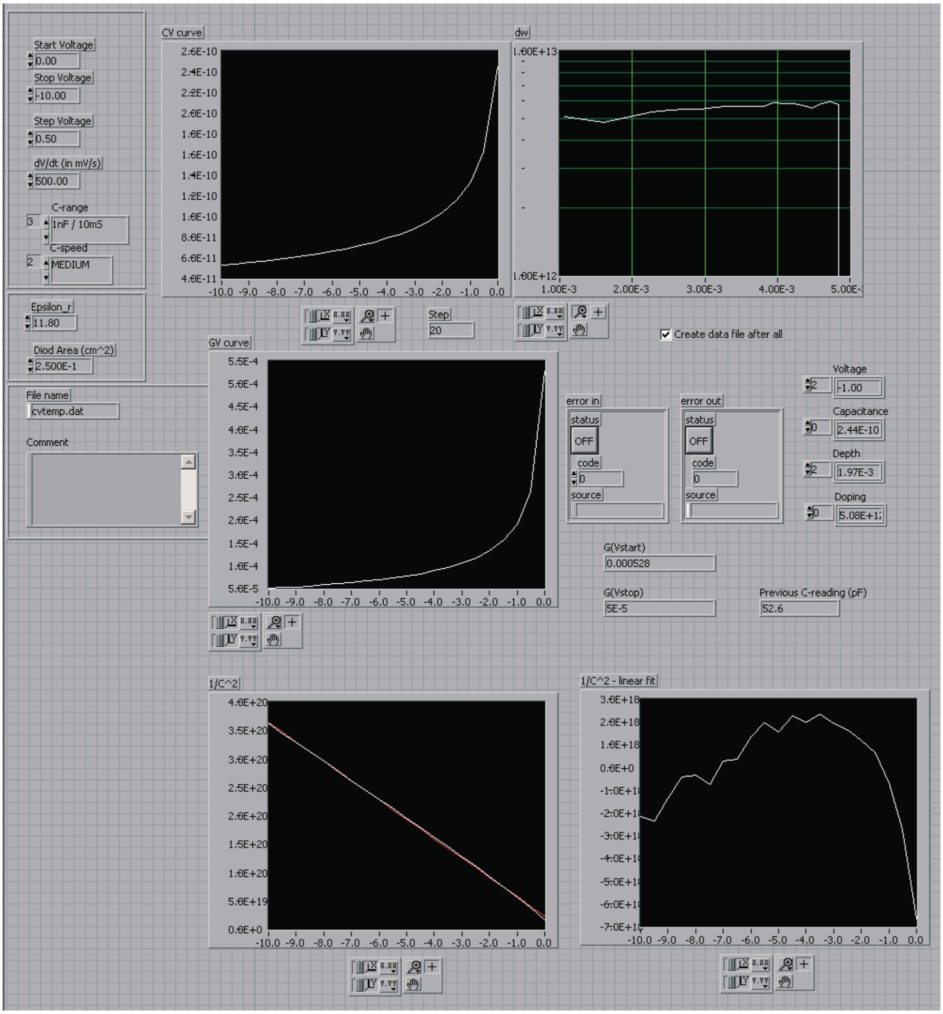
This is done automatically in the fitting routine if there are several weighting functions windows and/or several measurement files. There must be more than one window for the weighting function(s), and preferably as many as possible to maximise the number of points in the Arrhenius plot.

Sometimes peaks are misfitted; if the activation enthalpy extracted from the Arrhenius plot of a single weighting function is more than a factor **`errorlimit`** away from the activation enthalpy given in **`dlts_peak_guesses`**, the result will not be included when averaging the apparent capture cross section and activation enthalpy for all the different weighting functions. If the result is ignored, the linear fit in the Arrhenius plot is drawn in black.

$T_{\mathrm{p,LI,W6}}$ in the Arrhenius plot, is the temperature position in lock-in window 6 of the DLTS peak arising from a defect with the given properties.

Variables in the input file relevant for this feature:

```
do_arrhenius=1
errorlimit
weights
dlts_peak_guesses
allow_negative_amplitude
T_init
T_end
range_to_include_beside_outermost_peaks
T_range_offset
iterations_in_fitting_function
```



Start Voltage 0.00

Stop Voltage -10.00

Step Voltage 0.50

dV/dt (in mV/s) 500.00

C-range 3 1nF / 10mS

C-speed 2 MEDIUM

Epsilon_r 11.80

Diod Area (cm^2) 2.500E-1

File name cvtemp.dat

Comment

Step 20

☑ Create data file after all

error in status OFF code 0 source

error out status OFF code 0 source

Voltage 2 -1.00

Capacitance 0 2.44E-10

Depth 2 1.97E-3

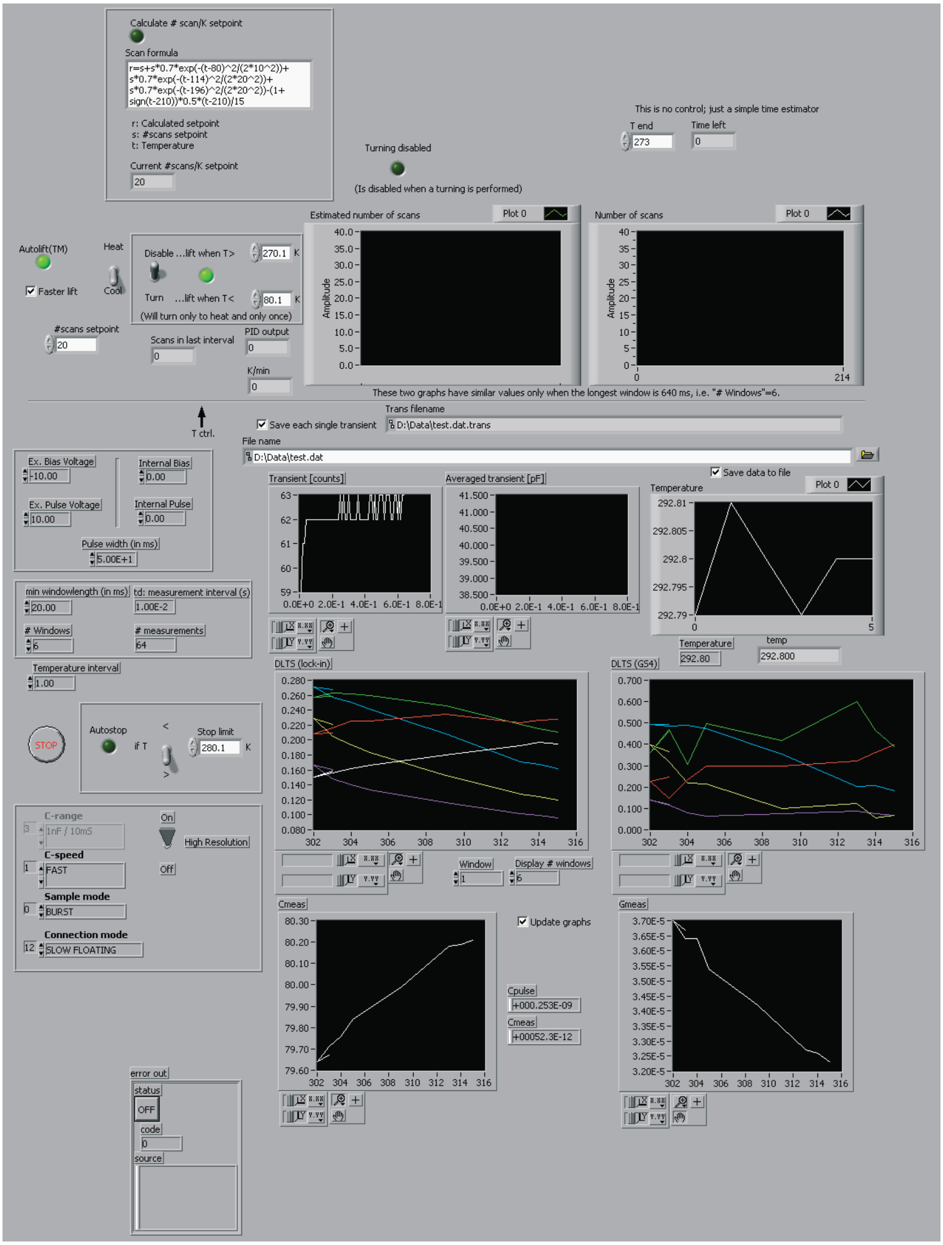
Doping 0 5.08E+1:

G(Vstart) 0.000528

G(Vstop) 5E-5

Previous C-reading (pF) 52.6

Figure A.1: Screenshot of the *CV* program `CVmeas with G.vi`.



Calculate # scan/K setpoint

Scan formula

```
r=s+s*0.7*exp(-(t-80)^2/(2*10^2))+
s*0.7*exp(-(t-114)^2/(2*20^2))+
s*0.7*exp(-(t-196)^2/(2*20^2))-(1+
sign(t-210))*0.5*(t-210)/15
```

r: Calculated setpoint
s: #scans setpoint
t: Temperature

Current #scans/K setpoint: 20

This is no control; just a simple time estimator

T end: 273
Time left: 0

Turning disabled

(Is disabled when a turning is performed)

Autolift(TM)

☑ Faster lift

Heat / Cool

Disable ...lift when T> 270.1 K

Turn ...lift when T< 80.1 K

(Will turn only to heat and only once)

#scans setpoint: 20

Scans in last interval: 0

PID output: 0

K/min: 0

These two graphs have similar values only when the longest window is 640 ms, i.e. "# Windows"=6.

T ctrl.

☑ Save each single transient

Trans filename: D:\Data\test.dat.trans

File name: D:\Data\test.dat

☑ Save data to file

Ex. Bias Voltage: -10.00
Internal Bias: 0.00
Ex. Pulse Voltage: 10.00
Internal Pulse: 0.00
Pulse width (in ms): 5.00E+1

min windowlength (in ms): 20.00
td: measurement interval (s): 1.00E-2
Windows: 6
# measurements: 64

Temperature interval: 1.00

STOP

Autostop if T < / > Stop limit 280.1 K

C-range: 3 1nF / 10mS
C-speed: 1 FAST
Sample mode: 0 BURST
Connection mode: 12 SLOW FLOATING

On / Off High Resolution

Temperature: 292.80
temp: 292.800

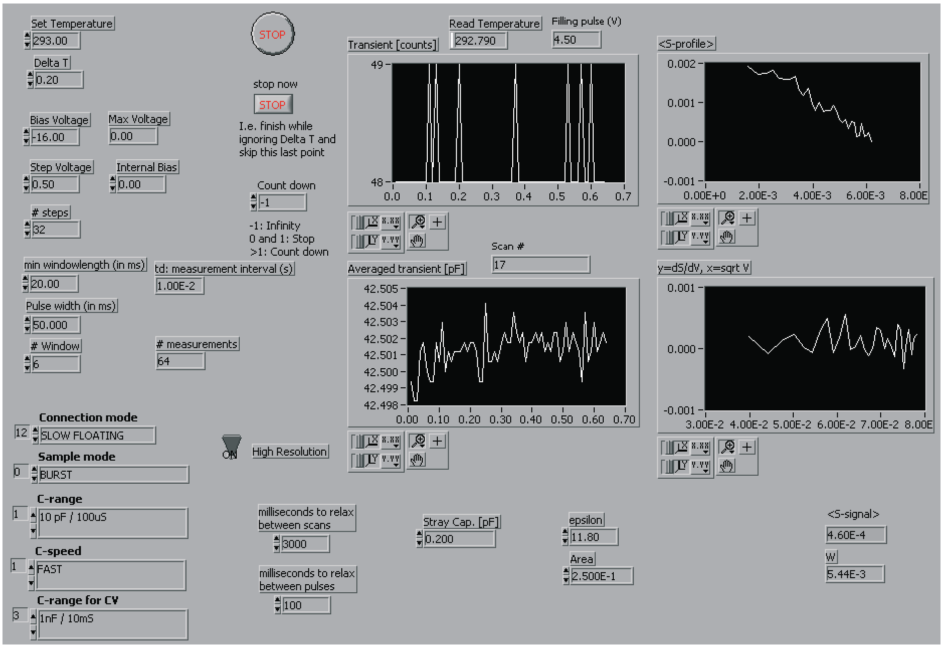
Window: 1
Display # windows: 6

☑ Update graphs

Cpulse: +000.253E-09
Cmeas: +00052.3E-12

error out
status: OFF
code: 0
source:

Figure A.2: Screenshot of the DLTS program on Asterix: `DLTS_Asterix_current_version.vi`.


Set Temperature 293.00

Delta T 0.20

Bias Voltage -16.00

Max Voltage 0.00

Step Voltage 0.50

Internal Bias 0.00

steps 32

min windowlength (in ms) 20.00

td: measurement interval (s) 1.00E-2

Pulse width (in ms) 50.000

Window 6

measurements 64

STOP

stop now STOP

I.e. finish while ignoring Delta T and skip this last point

Count down -1

-1: Infinity
0 and 1: Stop
>1: Count down

Read Temperature 292.790

Filling pulse (V) 4.50

Scan # 17

Connection mode 12 SLOW FLOATING

Sample mode 0 BURST

C-range 1 10 pF / 100uS

C-speed 1 FAST

C-range for CV 3 1nF / 10mS

ON High Resolution

milliseconds to relax between scans 3000

milliseconds to relax between pulses 100

Stray Cap. [pF] 0.200

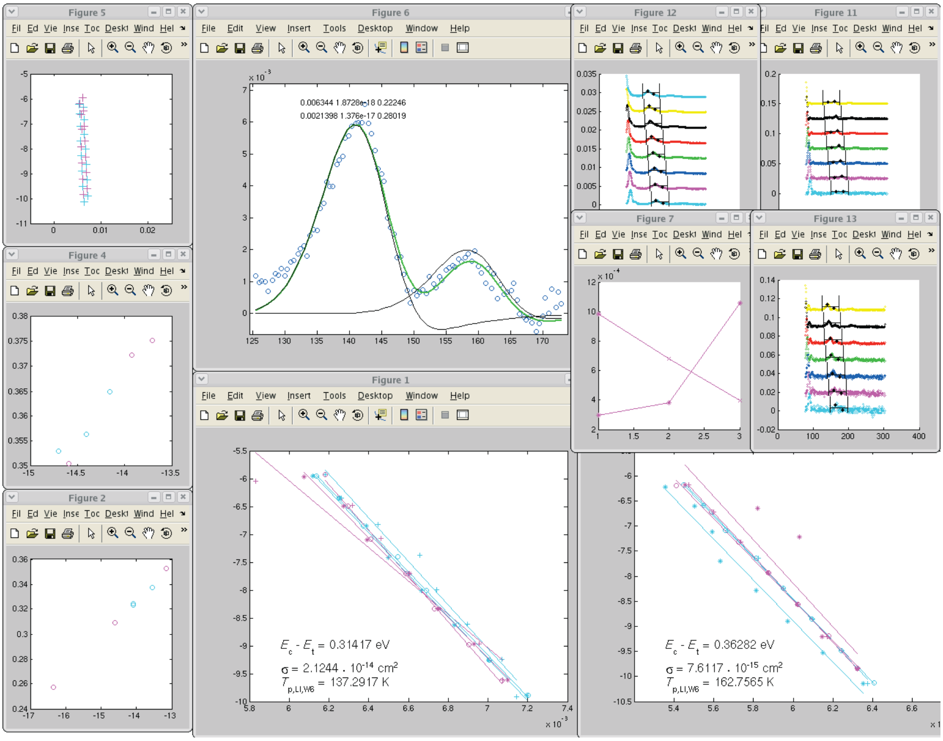
epsilon 11.80

Area 2.500E-1

<S-signal> 4.60E-4

W 5.44E-3

Figure A.3: Screenshot of the depth-profiling program `Profiling_semitrap_with_dS_vs_dV_with_stopnow.vi`.



0.006344 1.8728e-18 0.22246
0.0021398 1.376e-17 0.28019

$E_c - E_t$ = 0.31417 eV
σ = 2.1244 . 10^{-14} cm^2
$T_{p,LI,W6}$ = 137.2917 K

$E_c - E_t$ = 0.36282 eV
σ = 7.6117 . 10^{-15} cm^2
$T_{p,LI,W6}$ = 162.7565 K

Figure A.4: Screenshot after the DLTS-transient analyser is done fitting the VOH peak and the E1 peak in the spectra recorded after three different annealing times. Lock in and GS4 are used as weighting functions.

Generate a peak-amplitude evolution plot

As I have investigated defect dynamics throughout the whole of this project, it has been important to extract all the peak amplitudes from the spectra after each annealing step. This program usually does a good job in this respect if **`do_peakplot=1`**.

When extracting peak amplitudes, I tend to use a rather small value for **`range_to_include_beside_outermost_peaks`** (e.g. 4, or even 2), although this is always a difficult decision. The advantage of a small range is a much faster fitting and the peak value is usually visually correct; the disadvantage is that this peak value may be wrong due to the peak being misshaped or overlapped by other peaks.

When I extract peak amplitudes, I usually do that and nothing else; hence, I need only one window of one weighting function. Usually I want to have the amplitude of the VO peak, and therefore I use the $(640\text{ ms})^{-1}$ window (window 6 for lock in), unless the signal-to-noise ratio is low – in which case I also use the $(2.56\text{ s})^{-1}$ window (window 8). **`peakplotwindow`** and **`peakplotweight`** gives which window to use from which weighting function, respectively, for extracting the peak amplitudes of the peaks sought after (those listed in **`dlts_peak_guesses`**). The most effective way to extract the peak amplitudes from the spectrum calculated with the sixth lock-in window, is to load **`weights{1}{1}`** with the 64-element long lock-in pattern and give explicitly that **`transientlength=2^6`**. The window index enters, however, the calculation of the DLTS signal and it is therefore necessary to set **`window_offset=5`**. A more general syntax can be found in the input file:

```
weights{length(weights)+1}=binary_pattern2dlts_weight→
→([repmat(-1,1,transientlength/2) →
→repmat(1,1,transientlength/2)],transientlength);
window_offset=-1;
elements_in_first_window=2;
```

If **`window_offset=-1`** it will automatically be set to one less than the longest window possible with the given transient length and the given number of elements in the first window of the weighting function. In other words, with these lines only **`transientlength`** needs attention.

A useful function in this business is **`complete_peak_plot`**. I call it like this:

```
[all_peakamplitudes,amplitudetable]=→
→complete_peak_plot(all_best_fits,e_n,weights,→
→abscissa,peaks_to_ignore,window_offset),
```

where **`all_best_fits`** is the structure resulting from a DLTS-peak fitting run, **`e_n`** is the emission rates at the peak maxima (these are calculated by **`generate_e_n_max`** and should already be in the memory after the fitting), **`abscissa`** is either the annealing times or the annealing temperatures of each annealing step, and peak numbers listed in **`peaks_to_ignore`** will not be included in the resulting **`amplitudetable`**. A figure with the peak-amplitude evolution should have emerged by now, and **`amplitudetable{1}{1}`** will give the same data organised in a table.

Variables in the input file relevant for this feature:

```
do_peakplot=1
peakplotweight
peakplotwindow
do_arrhenius=1
dlts_peak_guesses
T_init
T_end
range_to_include_beside_outermost_peaks
T_range_offset
weights
window_offset
iterations_in_fitting_function
T_change
allow_negative_amplitude
```

Subtract each transient of one data file from those in another

This feature is used to follow the evolution of small DLTS peaks overlapped by other constant peaks. **subtractname**, **format_of_subtract_files**, **Crname_subtract**, **format_of_Cr_files_subtract** and **Cr_belongs_to_subtract_name_index** should contain file names, file formats, **.dat** file names corresponding to each of the transient files (unless standard DLTS files are used), the format of these, and the array mapping the **.dat** files to the **.trans** files, everything analogous to the variables for loading files for normal calculation of DLTS spectra (see above).

In addition to these variables is **subtract_belongs_to_name_index** which maps the files listed in **subtractname** to the files in **name**. With **do_subtract=1**, each transient of each file in **name** will be subtracted by the closest (in temperature) transient from the corresponding file in **subtractname** before the program continues exactly as usually.

Variables in the input file relevant for this feature:

```
do_subtract=1
subtractname
format_of_subtract_files
Crname_subtract
format_of_Cr_files_subtract
Cr_belongs_to_subtract_name_index
subtract_belongs_to_name_index
```

Make a graph showing activation enthalpy versus apparent capture cross section

When an Arrhenius plot is being made, a graph will be plotted to show how the activation enthalpy and apparent capture cross section (calculated for each file with all weighting functions) relate. The point is that they will spread out on a diagonal line, because the temperature position of a DLTS peak is rather well defined and the temperature position calculated from a set of activation enthalpies and apparent capture cross sections is rather well defined, but there is significant uncertainty in the extracted activation enthalpy and apparent capture cross section. Therefore I like to take the average of the activation enthalpies I calculate and look at this graph to find the apparent capture cross section that will give the DLTS peaks in the correct positions.

Save `.png` files

Most of the graphs that are being plotted can be saved automatically to a **`.png`** file.[5] If **`plot_spectra=1`** and **`save_spectra=1`**, files such as **`Spectrum_wg03_f01.png`** will contain all the windows of the calculated spectra for each weighting function and each file. If **`do_arrhenius=1`**, files such as **`Totalarrheniusplot_pk02.png`** will contain all the Arrhenius plots calculated for one particular peak and **`Defectplot_pk02.png`** will contain the corresponding plot of the activation enthalpies versus the apparent capture cross sections.

If **`save_fitting_plots=1`** and **`plot_frequency`** is larger than zero, the fitting window will be saved to files like **`Fitting_pk02_win07_wg03_f04.png`** when the best fit is reached. The three (or multiples of three if fitting overlapping peaks) fitting parameters are also displayed in the graph.

Variables in the input file relevant for this feature:

```
save_spectra=1
save_fitting_plots=1
plot_spectra=1
do_arrhenius=1
plot_frequency
```

FFT smoothing of the spectra

FFT smoothing was supposed to be a sophisticated method for reducing the noise in DLTS spectra, but was never fully implemented. I wanted to do FFT, make a frequency cut-off to remove fast fluctuations and then do an inverse FFT. Later I learnt that polynomial fitting (**`polyfit`** in Matlab) is the way to go about these matters,[6] but this is also not implemented.

[5] If a bitmap format for line drawings is needed, then **`.png`** is the one to use; it does not require a patent license like **`.gif`**, it supports palettes of 24-bit RGB colours (compared to the 8-bit palette of **`.gif`**), it is lossless (unlike **`.jpg`** which causes horrendous compression artefacts in single-coloured surfaces such as those found in graphs) and with simple graphs, **`.png`** files are smaller than normal **`.jpg`** files.

[6] Lasse Vines, private communication.

Variables in the input file relevant for this feature:

do_fft=1

Functions not implemented

Some functions never got developed further than the introduction of an unused, distractive variable in the code. In the input file, some of these variables are:

> **successlimit**: Some annealing-fitting algorithms lower the temperature after a given number of successes; for this purpose such a variable would be necessary
>
> **plot_raw_also**: When the graph shows a smoothened spectrum, it would be nice to compare with the raw one (lying as a shade in the background)

Other comments

The spectra calculated with all the different weighting functions are located in **spectrum**. The averaged spectra are in **spectrum_avg**. The corresponding temperatures are in **T** and **T_avg**, respectively. The final fitting parameters are stored in the cell structure **all_best_fits**. **fitted_peak_positions** contains all the peak positions found during the fitting. The data points of the resulting Arrhenius plots are stored in the cell structure **arrhenius_points**. **e_n** contains the emission rates at the peak maxima – one value for each window of each weighting function. **generate_e_n_max** generates these values with a rather generic algorithm.

A.3 Defect-reaction modeller

Already at an early stage I was sufficiently aware of my own limitations when it comes to correctly punching long differential equations to see the necessity of making a program that would do this task automatically. This program solves just this problem – from just simple reaction equations as input, it will construct the whole set of differential equations necessary to describe the model. Further, the differential equations will be solved numerically by the Matlab function **ode15s**. Some versions of the program will also run an infinite number of simulations, simultaneously for any number of annealing temperatures, while changing the desired parameters to search for a model that fits better the measured defect concentrations.

Appendix D shows an example of an input file for the defect-reaction modeller. This file, named **props_and_reactions.m**, contains most variables that requires attention. In this program the various defects are not referred to by name, but rather by an index. The index of a defect is given by its position in the cell structure **defect_properties**; its labelling should be put in corresponding cell in the cell structure **legendstr**. The various properties of the defect essential to the model should also be placed in **defect_properties**. The properties, organised in a five-element vector, are initial defect concentration (-1 will be replaced by the initial concentration from the measurement files), pre-factor and activation energy for dissociation of this defect, pre-factor and activation energy for migration of this defect. Values not essential to the model are ignored and can be set to **NaN**.

Once the various defects are given an index, the reaction equations can be entered. These reside in the two cell structures **dissociation** and **bonding**. For instance, if the defects have the following indices: VO: 1, $\mathrm{O_i}$: 2, $\mathrm{VO_2}$: 3, H: 4, VOH: 5, the reaction

$$\mathrm{VOH} \rightarrow \mathrm{VO} + \mathrm{H} \tag{A.1}$$

can be set as the first dissociative reaction by giving the command

```
dissociation{1}=[5 1 4].
```

In other words: Defect 5 dissociates into the defects 1 and 4. The order of the two latter has no importance. The dissociation rate will be calculated from **defect_properties{5}(2)** (pre-factor) and **defect_properties{5}(3)** (dissociation energy of defect 5). The reaction

$$\mathrm{VO} + \mathrm{O_i} \rightarrow \mathrm{VO_2} \tag{A.2}$$

can be set as the second migration reaction by giving the command

```
bonding{2}=[1 2 3].
```

In other words: Defect 1 migrates to defect 2 – these react and form defect 3. Here the order is important, as the migration rate will be calculated from the properties of the first defect.

The effective capture radius of the bonding reactions is given by **R_bonding**. The syntax of this changed under way as I opened for the possibility of having an energy barrier for a bonding reaction. In the latest versions, each cell of **R_bonding** should correspond to the cell in **bonding** with the same index. The cells of **R_bonding** should have a three-element vector with the capture radius for the corresponding bonding reaction, the pre-factor of the barrier for this reaction to happen and the energy barrier for this reaction, respectively. A bonding reaction without a reaction barrier would typically require

```
R_bonding{2}=[5e-8 1 0] % cm 1 eV,
```

where the capture radius is 0.5 nm.

This model-input structure is not entirely general; for instance, some reprogramming is necessary to give the possibility for a defect to dissociate in two different reactions. Also, the bonding reactions assumes that one of the species is stationary.

To automatically optimise the parameters to model as close as possible measured concentrations, it is necessary to load the measurement data. The variables for this is in the beginning of **props_and_reactions.m**. **filenames** and **path** point to the files to load. These files should have the annealing time in the first column followed by columns with the concentration of various defects. The two dimensional matrix **columns_to_import** tells which columns to read – the columns of one file for each row with the corresponding defects arranged in columns. The vector **data_map** tells the program which defects are being loaded by stating the defect index of each of the defects in **columns_to_import**. The width of the two variables should, thus, be equal. **annealing_temperatures_of_files_C** (**C** for Celsius) gives the annealing temperature of each of the measured annealing series and will also give the temperatures at which to run the simulation. The measurement data from each file will be multiplied by the corresponding value in the vector **conversion_factors** (this allows for a mixture of files containing both DLTS amplitudes and concentrations to be loaded) and subtracted by the concentration in **background** (which can be either a single number or a vector with elements corresponding to each data file).

What remains now is to tell which parameters to optimise. These variables have, unfortunately, become somewhat spread around in the program. Some are found in **model_optimiser_multi_temperature.m**. **do_pin_start_concentration** should be zero or one and states whether or not to optimise the initial defect concentrations. If **unpin_measured_start_concentration=1**, even the measured initial concentrations (which are loaded by setting the initial concentration in **defect_properties** equal to -1) will be optimised. **do_keep_concentrations_equal_for_different_samples** states whether or not to guess the initial defect concentrations independently for each sample/annealing temperature (if the processing of the samples is not identical, it is probably correct to set this to zero). This file also contains **plotfrequency** which can be increased from 1 to 3 to print information about the parameter guessing to simplify trouble shooting. Taking these various choices into account, **initialise_guess_rules.m** will generate the matrix **guess_and_guessmethod_init** with the following columns: measurement set (the sample for which this parameter is used; 0 if the parameter is common to all samples/annealing temperatures), defect identifier (defect index), defect property (1 for initial concentration, 2 and 3 for dissociation properties, 4 and 5 for diffusion properties), initial guess value, probability distribution (0 for Gaussian, 1 for gamma, 2 for linear), volatility with which to guess, optional parameter for the distribution. The parameters from **R_bonding** are included similarly, but the second column refers to the cell index of **R_bonding** and the third to the column index of the property plus 100.[7] The capture radius is not optimised by default (i.e. not automatically included in **guess_and_guessmethod_init**) if it is 0.5 nm (in **R_bonding**). The reaction

[7] The addition of 100 (**R_bonding_propertycounter_offset** in **model_optimiser_multi_temperature.m**) is a non-optimal trick to make it possible to distinguish the bonding-reaction parameters from the normal defect properties.

barrier will not be optimised by default if the pre-factor is 1 and the energy barrier is 0.

If the algorithm in **initialise_guess_rules.m** does not prepare for the optimisation of the desired parameters, and only those, **guess_and_guessmethod_init** can be customised in **customise_guess_rules.m**.

The solver of ordinary differential equations, **ode15s**, requires a function that contains the set of differential equations. **display_diff_eqns.m** generates a file **dN_function.m** which contains all this. Thus, after the command

```
display_diff_eqns(defect_properties,dissociation,→
→bonding,R_bonding,'generate_file'),
```

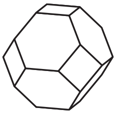
the simulation can be run for each annealing temperature by simply issuing the command

```
[time,concentrations]=→
→ode15s(@(t,N)dN_function(t,N,new_defect_properties,→
→new_R_bonding,model_temperature,k_B),→
→measurement_times,new_initial_concentrations),
```

where **new** means that the current guessing values replace the initial values. In the first generations of this program, I tried solving the differential equations my own way, but things became a lot faster and reliable after I was told about the **ode** functions.[8]

Over time, the program for automatic parameter optimisation has become more developed than the program doing one single simulation run, but this can still be used by calling the outermost function **manual_run.m**. The program that searches for the parameters giving the best fit to the measured data is started by running **loop_multi_temperature.m**. As for the transient-analysis program (section A.2) this program has a fitting temperature which determines how volatile to choose the guessing parameters. The initial fitting temperature is loaded from the file **T_init.dat**. This kind of input from files allows the values to be changed during run time – even remotely. After simulating as many times as given in **iterations_in_fitting_function.dat**, the fitting temperature is multiplied by 0.9 (**T_change**).[9] When the value in **T_end.dat** is reached, the currently best simulation is stored as a **.png** picture and a **.mat** file such as, for instance,

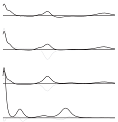
```
model_optimiser_result_→
→042_06-Dec-1976 16:52:32_1.626e-02.mat,
```



where the latter number is the minimum error returned by **calc_deviation.m**. This function gives a certain contribution from linear deviation and another for the logarithmic deviation. This because if only the linear deviation mattered, only the evolution of the large concentrations would be fitted, while minimisation of the logarithmic error would

[8] Lars Løvlie, private communication.

[9] In **model_optimiser_multi_temperature.m**.



take into account concentrations too small to be measured. The current algorithm is the result of some trial and error and is unlikely to be optimal.

Figure A.5 shows the optimal result of running the optimisation program for a few days. Here, measurements from annealing at only one temperature, namely 150 °C, was fitted.

A.4 Depth-dependent defect-reaction modeller

Unlike the defect-reaction modeller (section A.3), which attempts to reproduce DLTS amplitudes (or indeed any kind of concentration measurements), the depth-dependent defect-reaction modeller searches for parameters that make the model reproduce depth profiles of the various defect concentrations.

As the **`ode`** functions of Matlab can not handle both time and space simultaneously, this program uses the Matlab function **`pdepe`** to perform the defect simulation. I expect it to be possible to write a general algorithm that takes reaction equations as input and generates a function that can simulate the described model – like the algorithm described in section A.3 – but I have not done this for the depth-dependent program. The program is, therefore, rather rigid and some programming is required whenever the reaction equations of the model is changed.

`pdepe`, which runs the whole simulation, is called in **`simulate_diffusion.m`**. **`simulate_diffusion.m`** also contains the structure of the model. The parameters with which to simulate are fed to **`simulate_diffusion.m`** by **`guess_parameters2simulation_parameters.m`**, which collects and bundles the current model parameters in a pattern understandable to **`simulate_diffusion.m`**.

`auto_fit_hdiffusion_single_run.m` searches for model parameters that reproduces better (in terms of the error calculated, from a mixture of the linear and logarithmic deviation, by **`calc_profile_deviation.m`**) the measured depth profiles. The depth resolution, initial values of the parameters to optimise and the volatility with which to make guesses for these are loaded from **`start_values.m`**. Similarly to the defect-reaction modeller described in section A.3, the initial value of the fitting temperature, **`T_fit`**, is loaded from **`T_init.dat`** and is used to calculate a fitting volatility which **`hdiffusion_annealfit.m`** will use to make parameter guesses. This function takes the currently best parameters, perturb them according to the volatility, and runs the simulation by calling **`simulate_diffusion`**. The deviation from the measured data is calculated; if the deviation is smaller, the new parameters are kept for the next run. This is repeated **`iterations_in_fitting_function.dat`** times, after which **`T_fit`** is reduced one step. When **`T_end.dat`** is reached, the best parameters are returned.

The outermost function to run is **`loop.m`**. This will cause an infinite calling of parameter optimisations and saving the result of each run to a **`.png`** and **`.mat`** file. The latest generation of **`loop.m`** also stores the currently best set of parameters in a file so that it is transferred to the next run. This allows for several computers to use the currently best result to try, in parallel, to further improve the parameters.

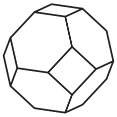
In summary, the model is defined in **`simulate_diffusion.m`** which must have a compatible **`guess_parameters2simulation_parameters.m`**. The initial parameter values to be optimised are set in **`start_values.m`**. The most automated optimisation routine is run by **`loop.m`**. Figure A.6 shows the final result of such a process.

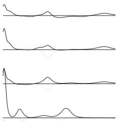
A.5 DLTS-movie maker

Comparing hundreds of DLTS spectra can be challenging at times. Therefore, I made a program that interpolates the DLTS spectra between the annealing temperatures. The name of the files,[10] their format, their path and various labelling information is defined in **`fileinfo.m`**. The remaining settings are in the main program: **`dlts_visualisation.m`**. The code (and commenting) is not particularly mature, but it is also not a very complicated program. The interpolation is done by

```
interpolated_spectra_of_single_sample=→
→interp1([1:number_of_annealing_steps)],→
→spectra_of_single_sample,→
→[1:1/annealing_resolution:number_of_annealing_steps],→
→'pchip').
```

Figure A.7 shows a snapshot from a movie created from some 100 DLTS spectra that shows, rather effectively, how the seven samples behave differently throughout the annealing series. The small graphs that can be seen on the left of the even pages throughout this thesis are also created by this program.





[10] This is, again, a time-consuming task made easy by Linux. An example script can be found in the program.

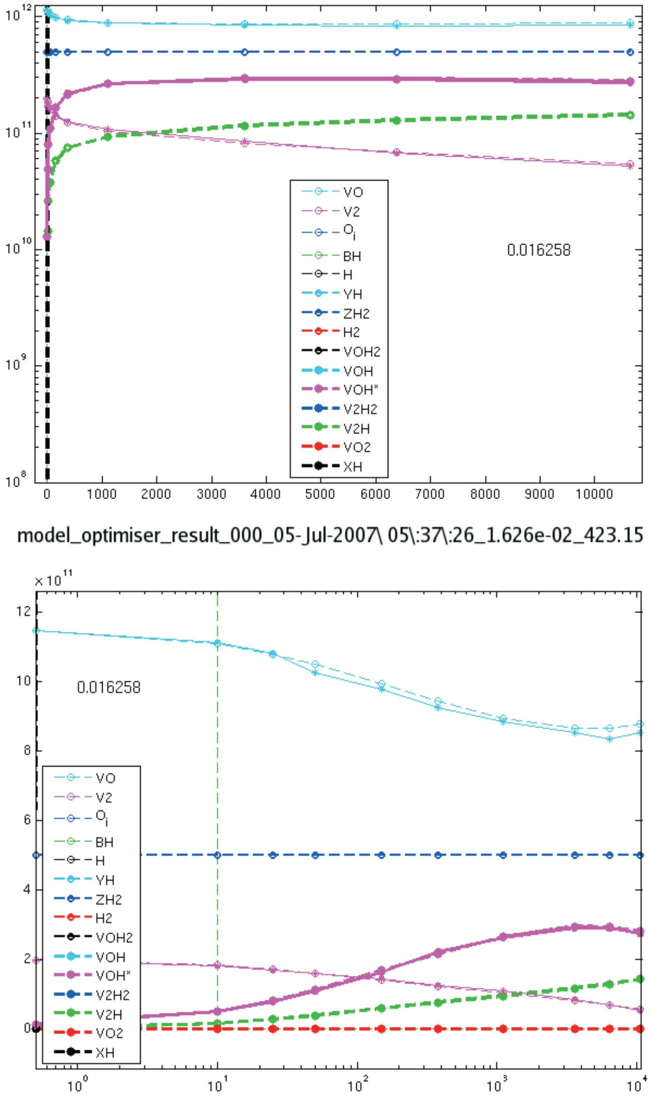


Figure A.5: Example of a parameter optimisation that causes the simulation (circles and dashed lines) to reproduce rather well the measured DLTS amplitudes (stars and solid lines). This sample was annealed isothermally at 150 °C. This particular model has one source of hydrogen and the hydrogen can be captured by VO, V_2 and a hydrogen trap. Also, VOH* dissociates. The *x*-axis shows the annealing time in minutes; the *y*-axis shows trap concentration. Both graphs show the same data, but with different axes.

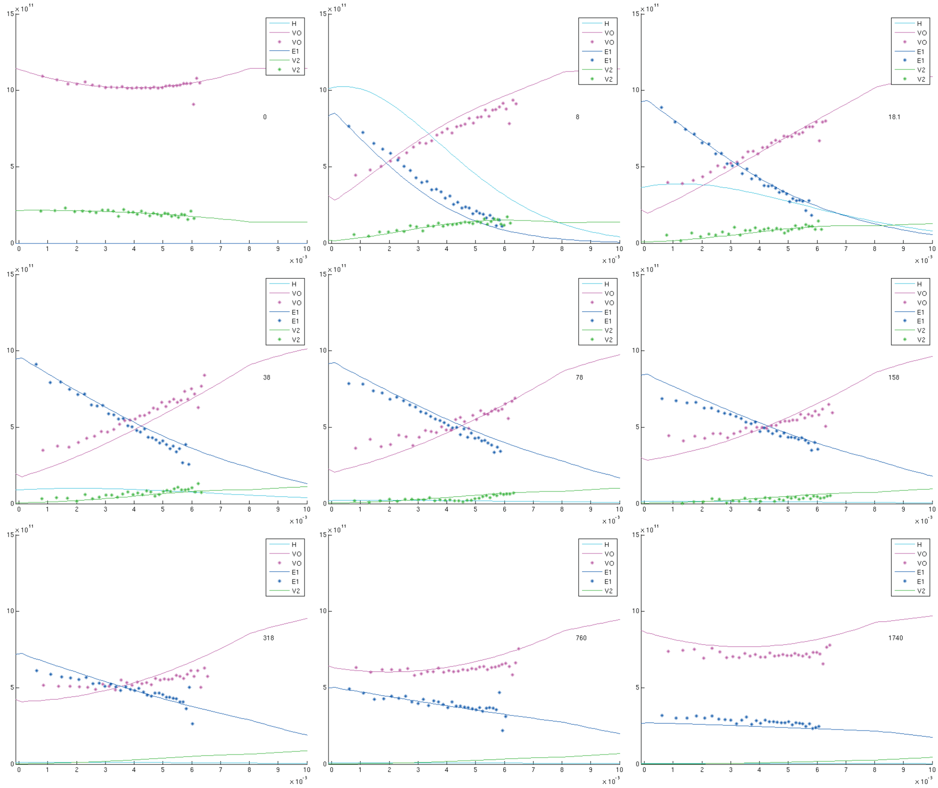


Figure A.6: The result of a continuous run of the depth-dependent defect-reaction modeller. The measured profiles are shown as stars and the simulation is shown as solid lines. This sample was annealed isothermally at 195 °C; this is actually the complete result of which a few annealing times is shown in paper VI.

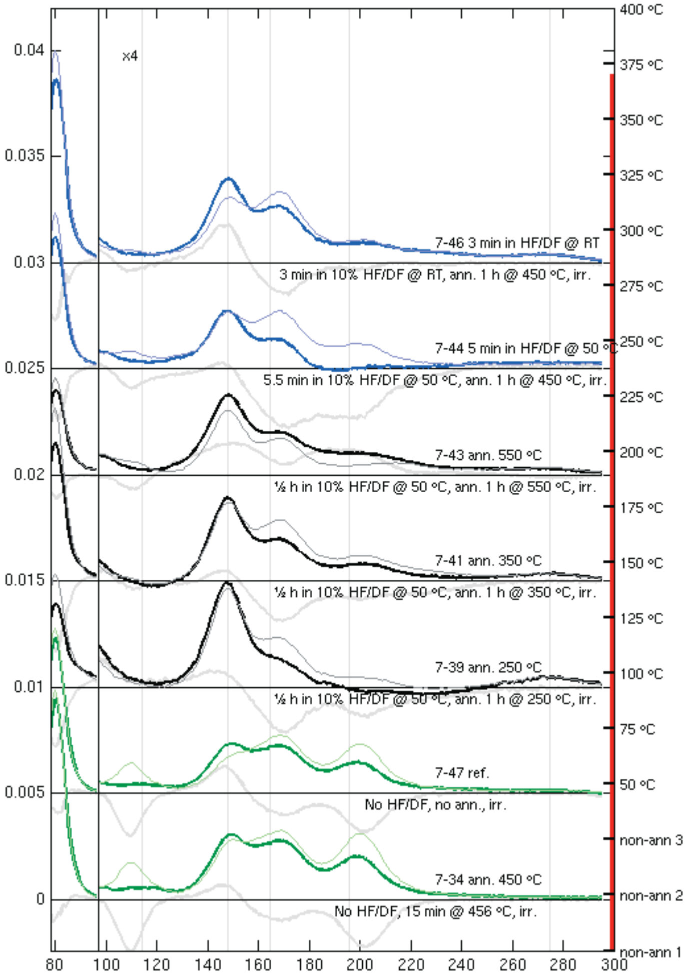
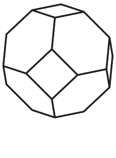
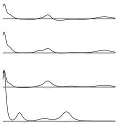


Figure A.7: Snapshot from a movie showing, simultaneously, how the DLTS spectra of seven samples changes throughout an isochronal annealing series. The annealing temperature is indicated by the red column on the right. The annealing was performed in steps of 25 °C – the nine frames in-between are interpolated. The thin, coloured line shows where the DLTS spectrum was five frames ago – the thick, grey line shows the difference of this to the current spectrum (scaled up).





Appendix B

User guides on hardware

B.1 Lift for automatically controlling the nitrogen level on the Asterix set-up

The purpose of the Lift is to have a way of controlling the temperature of the sample by the computer programs. Viewed as a black box, the Lift takes two input signals and a power supply of ∼16 V. A pulse going from 0 to 5 V on one of the two input signals will cause 100 clockwise or counter-clockwise steps on the stepper motor. The voltage for driving the system is not critical, but 16 V seems to give a good balance between lifting force and noise level (the motor runs less smoothly at higher voltages).

At present, the input signals of the Lift is connected to two pins in the parallel port of the computer and the letters ”A” and ”B” will cause the bit pattern 01 and 10, respectively, causing the Lift to lift or lower (or vice versa, depending on how the triggers from the parallel port is connected to the microcontroller) the level of nitrogen.

The stepper motor is very robust, but the users of the DLTS program are, nevertheless encouraged to make sure the lift is started sufficiently high such that it will not reach the lower limit before the stop-lift temperature (`Disable lift when T>`) is reached. Make sure the stop-lift temperature is set at a temperature matching the temperature rate and heater power and also that automatic Lift stopping is enabled. After the measurement is started, please wait half a minute to check that the Lift is able to lift and that it lifts in the correct direction.

Trivia: In the first version of the Lift, the stepper motor was controlled by simply closing four relays in the HP 34970A Data Acquisition Switch Unit in sequential order. The system worked satisfactory, but was noisy, slow and not sufficiently aesthetic to remain as it was.

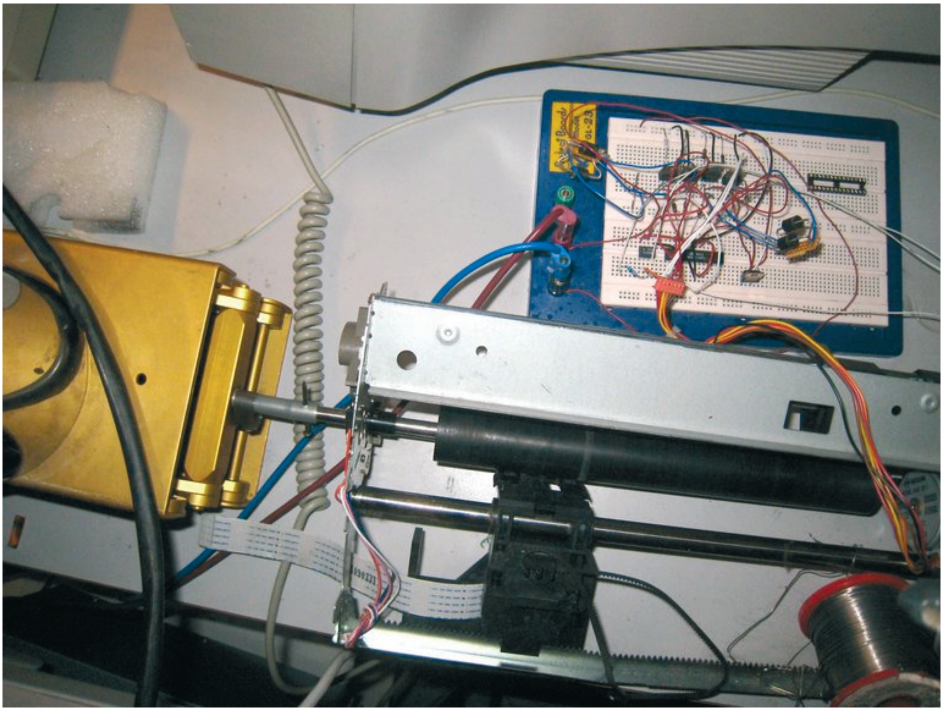


Figure B.1: The Lift.

B.1.1 Trouble shooting

Sometimes it may be necessary to reconnect lose wires according to figures B.2 and B.3. Sometimes the stepper motor will make noise, but not actually be able to apply any force. In this case there is usually a bad contact on one of the six wires (four red and two white wires) connected to the stepper-motor contact. Take out and reinsert all of them one at a time. The same symptoms will occur if the red wires are not connected in the correct sequence. For a stepper motor to run, it depends on getting pulses on its terminals in the correct order. Thus, if there is a signal to each of the terminals at different times, but the motor head does not rotate even if it is free to, the four input signals must be rearranged (or there may be a problem inside the stepper motor).

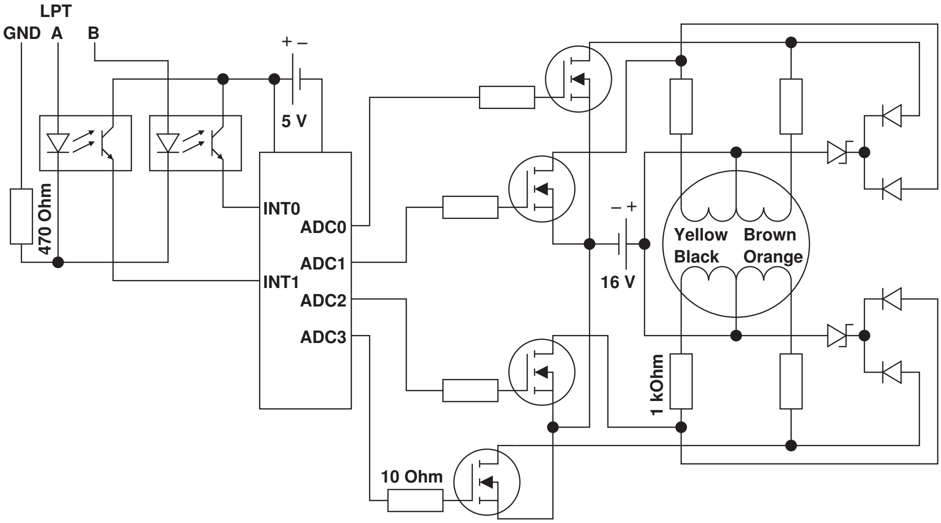


Figure B.2: A diagram of the electrical circuit of the stepper driver. The big rectangle is the microcontroller (ATmega8-16PI), into which (on `INT0` and `INT1` on pin 4 and pin 5, respectively) the trigger signals from the parallel port goes. The computer is separated from the microcontroller by sending the two trigger signals through an optocoupler (HCPL-2531). On the out signals (`ADC0` to `ADC3` on pin 23 to pin 26) there are 10-Ω resistors in front of the MOSFETs (BUZ11A). 1-kΩ resistors are connected in front of the coils of the stepper motor. Behind the motor, there are zener diodes (BZT03C24) and high-current diodes (HER604) to shield the driver circuit from voltage spikes. The voltage for the microcontroller and optocoupler (5 V) is supplied by a voltage regulator (MC7805).

B.1.2 Relevant LabView programs

- **`c:\measurements_A\dlts\DLTS_library_based_on_Obelix.llb`**: **`Lift control with PID cluster own PID c.vi`**[1] (for keeping the heating rate constant)

- **`c:\measurements_A\Misc\stand-alone_PID_control.vi`** (for keeping the temperature constant)

- **`c:\measurements_A\Misc\fastforward_lift_lpt.vi`**[2] (for continuously lowering or lifting)

[1] The file name states that the input and output parameters are transferred through cluster-type variables, that the standard PID algorithm of LabView has been replaced and that this is version `c`.

[2] It is very slow, as a matter of fact.

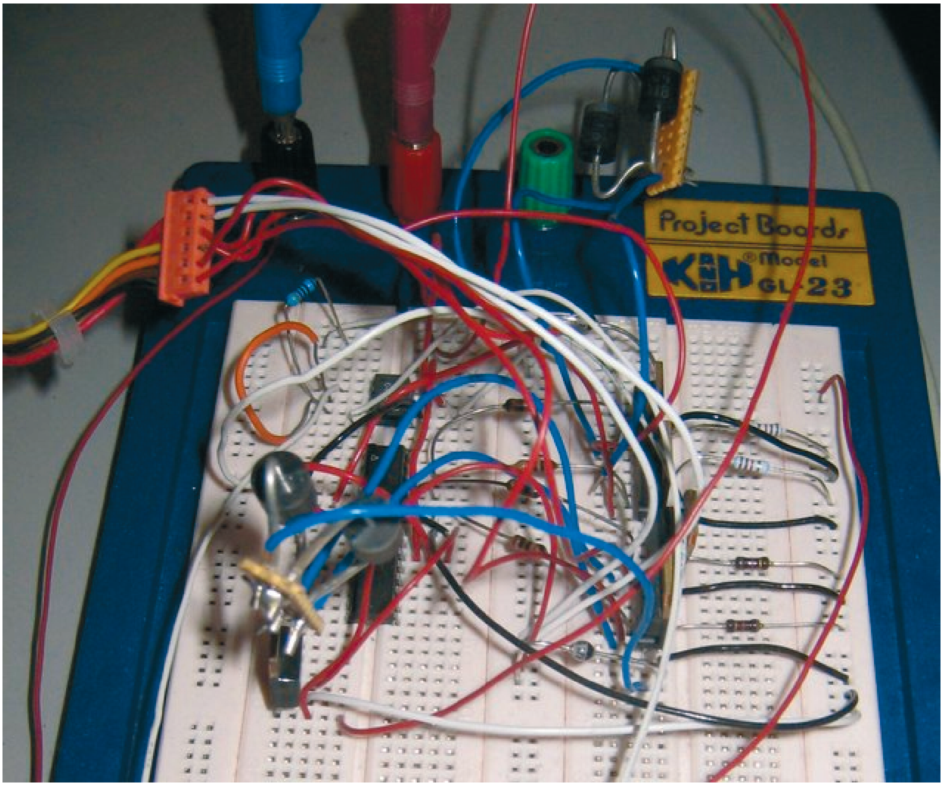


Figure B.3: An example of how the driver circuit may look when it is working. Ironically, this picture shows that a wire from one of the diodes has become disconnected at some point in time, but the Lift was working nevertheless. If there is no sound from the Lift when it is triggered by the computer, check for loose wires and symmetry breaks.

B.1.3 Spare parts

- The stepper motor and gear system is taken from a Star LC-10. It is ever more difficult to find these, but any stepper motor should work.
- Microcontroller: ATmega8-16PI
- Optocoupler: HCPL-2531
- Resistors: 10 Ω and 1 kΩ
- MOSFET: BUZ11A
- Zener diode: BZT03C24

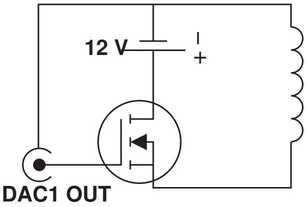


Figure B.4: Diagram of the electrical circuit of the computer-controllable mains switch. A coaxial cable connected to the BNC-2110 interface (on the `DAC1 OUT`) should be connected between the gate and the negative pole of the voltage source. The voltage on the gate should be 5 V to connect the relay and 0 V to disconnect. The voltage source for the relay should match the relay; this is currently 12 V.

- High-current diode: HER604
- Voltage regulator: MC7805

B.2 Controllable mains switch on Tiffy

This device was constructed to be able to use the computer to switch on and off the cooling pump on the closed helium system, but it can be used for switching on and off the voltage to any appliance that runs on the mains. After having failed a couple of times, the device is now in its third generation and has a design more pragmatic than ever. The input voltage is 12 V (for the coil in the relay) and a signal of 5 V is what is needed on the input signal to make the MOSFET switch the relay. For trouble shooting, compare the circuit with figures B.4 and B.5.

B.2.1 Relevant LabView programs

- `c:\measurements_tiffy\TSC\relaycontrol.vi`

B.2.2 Spare parts

- Any MOSFET that can run the relay without drawing much current from the input signal – for instance BUZ11A.
- Any 12-V relay that can withstand the desired current and can be driven by the MOSFET. An Omron G2R-1-SN(S) is currently installed, but it seems this may not be commercially available any longer.

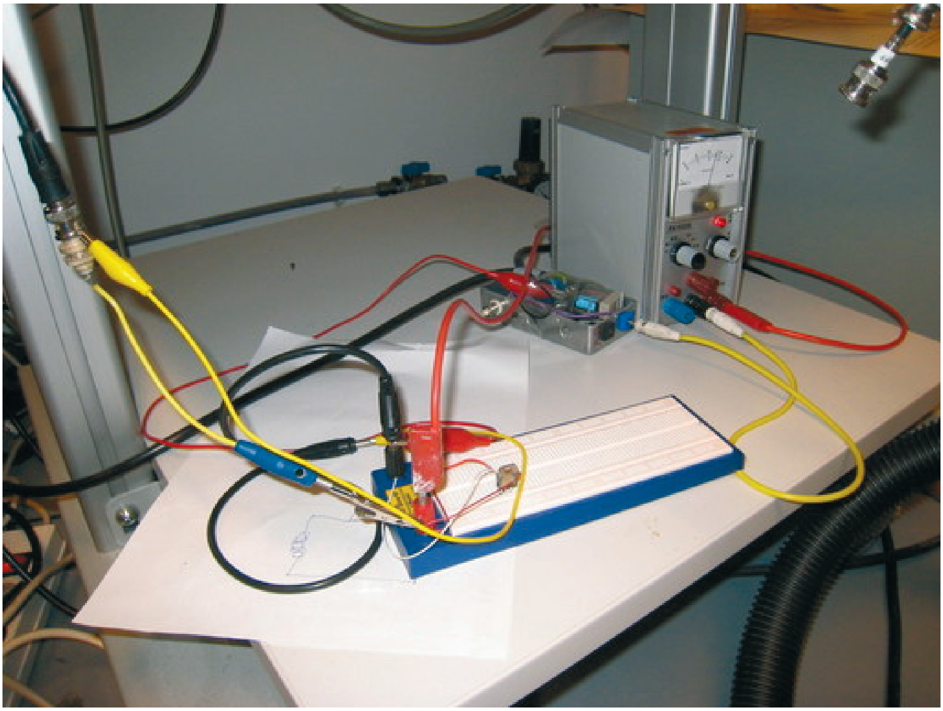


Figure B.5: The mains switch in operation.

B.3 Power amplifier for the heating elements on the Obelix set-up

This circuit is a solution to the problem that Obelix is equipped with two 50-W heating elements connected in series, while the current temperature controller, a Neocera LTC-11, can supply no more than 50 W. The circuit uses the analogue output signal of the temperature controller as input. This signal varies from 0 to 12 V as the internal PID controller in the Neocera gives an output of 0 to 100%, accordingly. The concern with this circuit is the temperature of the bipolar transistor. The BUT12A is expected to work if the junction temperature is below 150 °C, but this is easily exceeded if one is not cautious. The generation of heat in the transistor is

$$P_{\text{trans}} = (V_{\text{ext}} - R_{\text{heater}} I) I \, , \tag{B.1}$$

where V_{ext} is the voltage applied with the power supply, R_{heater} the total resistance of the heaters and I the current through the circuit (heaters). This is at maximum when I is 50% of the maximum current. This is the current when the PID on the the Neocera

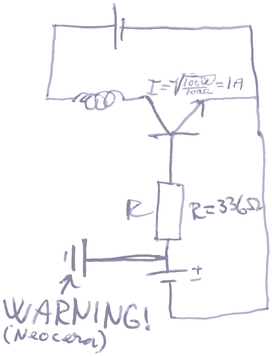


Figure B.6: A napkin diagram of the power amplifier for the heaters on the Obelix set-up. The coil is actually the two 50-W heating elements connected in series. The uppermost voltage source is the high-voltage power source (up to ∼90 V). The bipolar transistor is a BUT12A. In front of the transistor base there is a resistor which should be ∼340 Ω. The lowermost power source is the analogue output from the Neocera. The higher potential from the Neocera (see figure B.7) is connected to ground (marked here by a `WARNING!`); for this reason the high-voltage power source should be isolated from the ground. To me, at least, it seems not optimal to have two ground connections at different potentials in the circuit. The heating elements have, in total, a resistance of 100 Ω; the maximum current – when the Neocera wants 100% heating and gives 11.7 V on the analogue output – is, as the formula says, 1 A.

gives 50%.

The set-up has been operated with an voltage supply of 90 V to reach a sample temperature of 750 K, but then extreme caution must be taken with the transistor temperature. If the temperature of the junction becomes sufficiently large, it will start conducting regardless of what the base current is. Hence, if this situation arises, the set-up will heat 100% until the mains run dry. To avoid this, make sure the air cooling is switched on, monitor the temperature of the transistor, make sure the transistor has good thermal contact with the radiator fins and be very concerned if the power stays at ∼50% for several seconds.

If the transistor overheats and becomes uncontrollable, just switch off the voltage supply and let it cool off. The transistor will normally recover again after being cooled.

With this design there is a large voltage drop across the transistor, and if reliability and sensible use of energy is in focus, the circuit should be designed differently. A system with a duty cycle (such that the heaters are either on or off periodically) would satisfy these requirements. Then, if the output is 20% the heaters will be switched on for one second each five seconds. This is surely much more elegant, but I am slightly concerned about whether the constant changing of the current in the heater coils might cause noise in sensitive measurements – either via the magnetic field or the mains. Also, it would cost more time and money to construct such a circuit. I was planning to

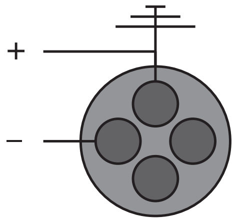


Figure B.7: The analogue output socket. The lowest potential (marked with a $-$ here) is connected to the ground of the power source in figure B.6, and the highest potential (marked with a $+$ here) is connected to the base resistor in figure B.6. The wire that is to be connected to $+$ is marked with ”up” and the one for the $-$ with ”left”.

improve the reliability of the circuit by replacing the bipolar transistor by two, or more, field-effect transistors in parallel to share the generation of heat between them, but this remains as a task for future generations of Ph.D. students.

B.3.1 Software

There are no LabView programs for this amplifier circuit – all that is needed is to make the Neocera give analogue output. This is achieved by choosing, on the **`Analog`** line in the **`Output configuration`** menu, **`SENSOR`**: **`#1`**, **`MODE`**: **`PID`**, **`MAX PWR`**: **`Ramp`**. It is also worth mentioning that Ioana Pintilie and I have, with great effort, arrived at a set of PID parameters that are currently being used for the Obelix set-up. They are not perfect (optimal parameters anyway depend on the maximum heating power, i.e. the voltage source), but may get the job done:

$$\begin{aligned} P &= 10 && \text{(B.2)} \\ I &= 100 && \text{(B.3)} \\ D &= 50 && \text{(B.4)} \end{aligned}$$

These parameters should be put on the **`Analog`** line. Also, the value of **`P0`** is zero.

B.3.2 Spare parts

As pointed out previously, the circuit should be redesigned rather than fixed, but if someone insists on still using it:

- The transistor handling the most power and the highest junction temperature – my choice was BUT12A

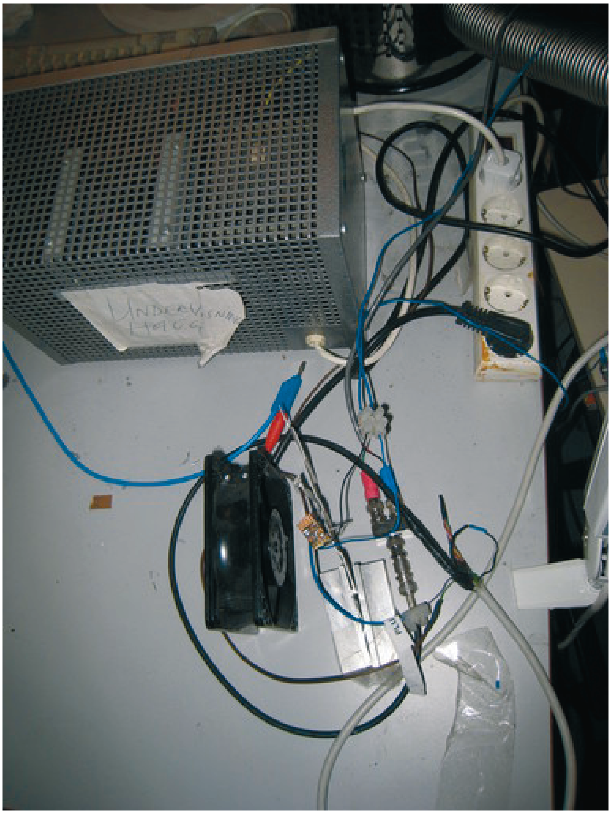
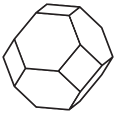
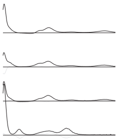


Figure B.8: The power amplifier. The cooling fan runs on 240 V AC.





Appendix C

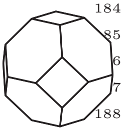
Example input file for the DLTS-transient analyser

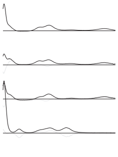
inputdata_temp.m

```
   % 2006 by Jan H. Bleka

   function[name,format_of_files,Crname,belongs_to_name_index >>
>> ,format_of_Cr_files,path,dlts_peak_guesses,T_range_offset, >>
>> allow_negative_amplitude,guessmethod_init,T_init,T_end,T_c >>
>> hange,iterations_at_T,iterations_in_fitting_function,succe >>
>> sslimit,range_to_include_beside_outermost_peaks,errorlimit >>
>> ,divide_by_Cr,do_arrhenius,do_peakplot,peakplotweight,peak >>
>> plotwindow,do_fft,average_interval,plot_frequency,save_fit >>
>> ting_plots,plot_spectra,save_spectra,plot_raw_also,write_o >>
>> utfiles,verbose,transientlength,weights,window_offset,do_s >>
>> ubtract,subtractname,subtract_belongs_to_name_index,format >>
>> _of_subtract_files,Crname_subtract,Cr_belongs_to_subtract_ >>
>> name_index,format_of_Cr_files_subtract,outname]=inputdata( >>
>> )

   % Should contain every input variable ever necessary to ch >>
>> ange


   % Run e.g. choosefiles .. *MCz-32*.dat in /j/eksperimenter >>
>> /raw/A/MCz-32/best/
   % /j/eksperimenter/Transient\\ converter/choosefiles .. *M >>
>> Cz-32*_3V_*6W*.dat
   % To extract the list of annealing times from the file nam >>
>> es:
   % ls *MCz-32*_3V_*6W*.dat|sed ’s/non-ann/0m/g;s/.*_\\([0-9 >>
>> o]\\+\\)m_.*/\\1/g’|tr ’\\n’ ’ ’|tr ’o’ ’.’
   % Returns: 0 8 18.1 38 78 158 318 760 1740


```

```

     name=char('A070304a_MCz-32_non-ann_f_ev_Al_b_dry_Ag_paste >>
>> _fwdnrev_-10V_3V_6W_cool_.dat.trans','A070404c_MCz-32_8m_1 >>
>> 95C_f_ev_Al_b_dry_Ag_paste_fwdnrev_-10V_3V_6W_cool_.dat.tr >>
>> ans','A070406a_MCz-32_18o1m_195C_f_ev_Al_b_old_Ag_paste_fw >>
>> dnrev_-10V_3V_6W_cool_.dat.trans','A070407a_MCz-32_38m_195 >>
>> C_f_ev_Al_b_Ag_paste_hot-air_dried_10s_fwdnrev_-10V_3V_6W_ >>
>> cool_.dat.trans','A070408a_MCz-32_78m_195C_f_ev_Al_b_old_A >>
>> g_paste_fwdnrev_-10V_3V_6W_cool_.dat.trans','A070408g_MCz- >>
>> 32_158m_195C_f_ev_Al_b_Ag_paste_hot-air_dried_5s_fwdnrev_- >>
>> 10V_3V_6W_cool_.dat.trans','A070409f_MCz-32_318m_195C_f_ev >>
>> _Al_b_Ag_paste_hot-air_dried_15s_fwdnrev_-10V_3V_6W_cool_. >>
>> dat.trans','A070410f_MCz-32_760m_195C_f_ev_Al_b_dry_Ag_pas >>
>> te_fwdnrev_-10V_3V_6W_cool_.dat.trans','A070411c_MCz-32_17 >>
>> 40m_195C_f_ev_Al_b_added_dry_Ag_paste_hot-air_dried_10s_fw >>
>> dnrev_-10V_3V_6W_cool_.dat.trans');
    Crname=char('A070304a_MCz-32_non-ann_f_ev_Al_b_dry_Ag_past >>
>> e_fwdnrev_-10V_3V_6W_cool_4.dat','A070404c_MCz-32_8m_195C_ >>
>> f_ev_Al_b_dry_Ag_paste_fwdnrev_-10V_3V_6W_cool_4.dat','A07 >>
>> 0406a_MCz-32_18o1m_195C_f_ev_Al_b_old_Ag_paste_fwdnrev_-10 >>
>> V_3V_6W_cool_4.dat','A070407a_MCz-32_38m_195C_f_ev_Al_b_Ag >>
>> _paste_hot-air_dried_10s_fwdnrev_-10V_3V_6W_cool_4.dat','A >>
>> 070408a_MCz-32_78m_195C_f_ev_Al_b_old_Ag_paste_fwdnrev_-10 >>
>> V_3V_6W_cool_4.dat','A070408g_MCz-32_158m_195C_f_ev_Al_b_A >>
>> g_paste_hot-air_dried_5s_fwdnrev_-10V_3V_6W_cool_4.dat','A >>
>> 070409f_MCz-32_318m_195C_f_ev_Al_b_Ag_paste_hot-air_dried_ >>
>> 15s_fwdnrev_-10V_3V_6W_cool_4.dat','A070410f_MCz-32_760m_1 >>
>> 95C_f_ev_Al_b_dry_Ag_paste_fwdnrev_-10V_3V_6W_cool_4.dat', >>
>> 'A070411c_MCz-32_1740m_195C_f_ev_Al_b_added_dry_Ag_paste_h >>
>> ot-air_dried_10s_fwdnrev_-10V_3V_6W_cool_3.dat');
    outname=char('A070304a_MCz-32_non-ann_f_ev_Al_b_dry_Ag_pas >>
>> te_fwdnrev_-10V_3V_6W_cool_4_alltrans.dat','A070404c_MCz-3 >>
>> 2_8m_195C_f_ev_Al_b_dry_Ag_paste_fwdnrev_-10V_3V_6W_cool_4 >>
>> _alltrans.dat','A070406a_MCz-32_18o1m_195C_f_ev_Al_b_old_A >>
>> g_paste_fwdnrev_-10V_3V_6W_cool_4_alltrans.dat','A070407a_ >>
>> MCz-32_38m_195C_f_ev_Al_b_Ag_paste_hot-air_dried_10s_fwdnr >>
>> ev_-10V_3V_6W_cool_4_alltrans.dat','A070408a_MCz-32_78m_19 >>
>> 5C_f_ev_Al_b_old_Ag_paste_fwdnrev_-10V_3V_6W_cool_4_alltra >>
>> ns.dat','A070408g_MCz-32_158m_195C_f_ev_Al_b_Ag_paste_hot- >>
>> air_dried_5s_fwdnrev_-10V_3V_6W_cool_4_alltrans.dat','A070 >>
>> 409f_MCz-32_318m_195C_f_ev_Al_b_Ag_paste_hot-air_dried_15s >>
>> _fwdnrev_-10V_3V_6W_cool_4_alltrans.dat','A070410f_MCz-32_ >>
>> 760m_195C_f_ev_Al_b_dry_Ag_paste_fwdnrev_-10V_3V_6W_cool_4 >>
>> _alltrans.dat','A070411c_MCz-32_1740m_195C_f_ev_Al_b_added >>
>> _dry_Ag_paste_hot-air_dried_10s_fwdnrev_-10V_3V_6W_cool_3_ >>
>> alltrans.dat');

    name_oldformat=char('A070304a_MCz-32_non-ann_f_ev_Al_b_dry >>
>> _Ag_paste_fwdnrev_-10V_3V_6W_cool_4.dat','A070404c_MCz-32_ >>
>> 8m_195C_f_ev_Al_b_dry_Ag_paste_fwdnrev_-10V_3V_6W_cool_4.d >>
>> at','A070406a_MCz-32_18o1m_195C_f_ev_Al_b_old_Ag_paste_fwd >>
>> nrev_-10V_3V_6W_cool_4.dat','A070407a_MCz-32_38m_195C_f_ev >>
>> _Al_b_Ag_paste_hot-air_dried_10s_fwdnrev_-10V_3V_6W_cool_4 >>
>> .dat','A070408a_MCz-32_78m_195C_f_ev_Al_b_old_Ag_paste_fwd >>
>> nrev_-10V_3V_6W_cool_4.dat','A070408g_MCz-32_158m_195C_f_e >>
>> v_Al_b_Ag_paste_hot-air_dried_5s_fwdnrev_-10V_3V_6W_cool_4 >>
>> .dat','A070409f_MCz-32_318m_195C_f_ev_Al_b_Ag_paste_hot-ai >>
>> r_dried_15s_fwdnrev_-10V_3V_6W_cool_4.dat','A070410f_MCz-3 >>
>> 2_760m_195C_f_ev_Al_b_dry_Ag_paste_fwdnrev_-10V_3V_6W_cool >>
>> _4.dat','A070411c_MCz-32_1740m_195C_f_ev_Al_b_added_dry_Ag >>
>> _paste_hot-air_dried_10s_fwdnrev_-10V_3V_6W_cool_3.dat');





   format_of_files=[repmat(0,1,size(name,1))]; % 1, or more,  >>
>> are old fashion DLTS files with averaged transients and C_ >>
>> r. 0 are files with each single transient and requires a c >>
>> orresponding traditional file to get C_r.

   %Add information on number of windows:
   format_of_files=[format_of_files;repmat(6,1,length(format_ >>
>> of_files))];
   %format_of_files(2,x)=8;

   belongs_to_name_index=[1:size(name,1)]; % Files of format  >>
>> 0 must have a corresponding C_r file.
   format_of_Cr_files=[repmat(1,1,size(Crname,1))]; % Format  >>
>> can not be 0.


   % Now a function that subtracts the transients of one file >>
>>  from another. If the format is 0, the transients of the f >>
>> iles in name are subtracted by the transient closest in te >>
>> mperature from the corresponding file from subtractname


   subtractname=[]; % Spectra to subtract from spectra in nam >>
>> e


   subtract_belongs_to_name_index=[1:size(subtractname,1)]; % >>
>>  Which file in name do we subtract this from?
   format_of_subtract_files=[repmat(0,1,size(subtractname,1)) >>
>> ];

   % Window information:
   format_of_subtract_files=[format_of_subtract_files;repmat( >>
>> 8,1,length(format_of_files))];
   %format_of_subtract_files(2,x)=8;


   Crname_subtract=[];

   Cr_belongs_to_subtract_name_index=[1:size(Crname_subtract) >>
```

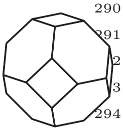
```
>> ]; % Files of format 0 must have a corresponding C_r file.
   format_of_Cr_files_subtract=[repmat(1,1,size(Crname_subtra >>
>> ct,1))];



   path=char('~/j/eksperimenter/raw/A/MCz-32/');


   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% end of file-related  >>
>> variables


   % Fitting parameters:
   VO_properties=[5.1e-14 0.1817];
   p89K_properties=[1e-15 0.174]; % Just a rough approximatio >>
>> n assuming 1e-15.
   V2double_properties=[1.2e-14 0.252];
   VOH_properties=[4e-15 0.32];
   VOH_properties=[1.1963e-17 0.24561]; % For GS4 W7 25012007
   %p165K_properties=[1e-18 0.24];%[8e-15 0.37];
   p165K_properties=[1.1418e-14 0.37252]; % For GS4 W7 250120 >>
>> 07
   p165K_unstable_properties=[1.3e-15 0.341];
   %V2single_properties=[3.1e-15 0.429];
   V2single_properties=[3.5326e-16 0.39244]; % For GS4 W7 250 >>
>> 12007
   V2single_overlap_properties=[1.5e-14 0.474];
   p214K_properties=[5.5633e-17 0.38994]; % For GS4 W7 250120 >>
>> 07
   H1_properties=[2.7388e-17 0.16214] % Doing some fitting in >>
>>  GS4W5 01052007

   dlts_peak_guesses={};
   dlts_peak_guesses{length(dlts_peak_guesses)+1}=[H1_propert >>
>> ies];
   %dlts_peak_guesses{length(dlts_peak_guesses)+1}=[VO_proper >>
>> ties];
   %dlts_peak_guesses{length(dlts_peak_guesses)+1}=[V2double_ >>
>> properties];
   %dlts_peak_guesses{length(dlts_peak_guesses)+1}=[VOH_prope >>
>> rties];
   %dlts_peak_guesses{length(dlts_peak_guesses)+1}=[p165K_pro >>
>> perties];
   %dlts_peak_guesses{length(dlts_peak_guesses)+1}=[p165K_pro >>
>> perties;V2single_properties];
   %dlts_peak_guesses{length(dlts_peak_guesses)+1}=[V2single_ >>
>> properties];
   %dlts_peak_guesses{length(dlts_peak_guesses)+1}=[p214K_pro >>
>> perties];
   %dlts_peak_guesses{length(dlts_peak_guesses)+1}=[p165K_uns >>
>> table_properties];
   %dlts_peak_guesses{length(dlts_peak_guesses)+1}=[p165K_uns >>
```

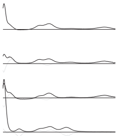




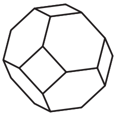
```
>> table_properties;V2single_properties];
   %dlts_peak_guesses{length(dlts_peak_guesses)+1}=[p165K_uns >>
>> table_properties;V2single_overlap_properties];
   %dlts_peak_guesses{length(dlts_peak_guesses)+1}=[V2single_ >>
>> properties];
   %dlts_peak_guesses{length(dlts_peak_guesses)+1}=[V2single_ >>
>> overlap_properties];

   T_range_offset={}; % NaN for automatic. The range of data  >>
>> to use when fitting a certain set of peaks
   %T_range_offset{length(T_range_offset)+1}=[-5 0];
   T_range_offset{length(T_range_offset)+1}=[NaN];
   T_range_offset{length(T_range_offset)+1}=[NaN];
   T_range_offset{length(T_range_offset)+1}=[NaN];
   T_range_offset{length(T_range_offset)+1}=[NaN];
   T_range_offset{length(T_range_offset)+1}=[NaN];
   %T_range_offset{length(T_range_offset)+1}=[-5 0];
   %T_range_offset{length(T_range_offset)+1}=[-10 30];
   %T_range_offset{length(T_range_offset)+1}=[NaN];


   amplitude_volatility=1e-2;
   sigma_volatility=5e-2;
   energy_volatility=1e-3;
   allow_negative_amplitude=1; % Needed to fit contribution f >>
>> rom minority carriers. If 1, guessmethod should probably b >>
>> e Gaussian, not gamma (I.e. guessmethod(1,1)=0 instead of  >>
>> 1
   guessmethod_init=[1 amplitude_volatility 0;1 sigma_volatil >>
>> ity 0;0 energy_volatility 123]; % distribution (Gaussian=0 >>
>> ,gamma=1,linear=2), volatility/distribution width (for Gau >>
>> ssian/gamma) / lower limit (for linear), NaN (for Gaussian >>
>> ) / min/max limit (for gamma) / max limit (for linear)
   T_init=30 ;
   T_end=3;
   T_change=0.9; % 0.9 gives a reduction of 10%
   iterations_at_T=1; % 1. Not implemented
   iterations_in_fitting_function=200; % 300. Number of itera >>
>> tions in annealfit. So the number of guesses at each T is  >>
>> actually iterations_at_T * iterations_in_fitting_function
   successlimit=iterations_at_T/20; % Not implemented. Reduce >>
>>  temperature after successlimit successes even if iteratio >>
>> ns_at_T is not reached
   range_to_include_beside_outermost_peaks=2; % Kelvin. 7. Ho >>
>> w much of the adjecent data to fit
   errorlimit=60/100; % Ignore result if it is more than erro >>
>> rlimit*100% outside the original guess in dlts_peak_guesse >>
>> s

   do_subtract=0; % Do subtract the transients of files in na >>
>> me by those in subractname
   divide_by_Cr=1; % Wether to divide the DLTS signal by the  >>
>> reverse capacitance
```

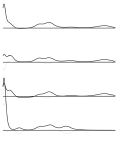
```
   do_arrhenius=1; % Fit data to find peaks and create Arrhen >>
>> ius plot
    do_peakplot=1; % Plot amplitudes vs time/annealing temper >>
>> ature
     peakplotweight=1; % Weight to use when making a peak-amp >>
>> litude plot
     peakplotwindow=1; % Window to use when making a peak-amp >>
>> litude plot
   do_fft=0; % Not functioning. Whether or not to do FFT afte >>
>> r the averaging
   average_interval=0.1; % Kelvin. If several points are with >>
>> in this interval they will be joined in an average. Use 0  >>
>> if averaging is not desired.
   plot_frequency=1; % 1. 0-6. How much to plot during fittin >>
>> g. Above 1 is slower.
   save_fitting_plots=1; % Creates png files
   plot_spectra=1; % Show the generated spectra
    save_spectra=1; % Makes png files
    plot_raw_also=0; % Not implemented. Include raw data for  >>
>> comparison
   write_outfiles=0; % Save generated spectra in given outfil >>
>> es
   verbose=1; % Tell which files are being opened

   % Calculate the maximum common transient length:
   if size(format_of_files,1)>1
   % We only construct weights to cover the file with the lea >>
>> st data points in the transients:
    transientlength=2^min(format_of_files(2,:));
   else
   % If # of windows is not defined for the files, assume 6 w >>
>> indows = 64 data points:
    transientlength=2^6;
   end

   % VO is not in window 8, so six windows may be wanted:
   %transientlength=2^6;

   weights={};

   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   window_offset=0; % is used for the 2^window and must be ch >>
>> anged to 5 if only the weight of window 6 is used (Don't c >>
>> hange here. See further down.). Set window_offset=-1 to ge >>
>> t the longest window. Keep window_offset=0 if there is mor >>
>> e than one weight.

     % R1, Lock in:
   %weights{length(weights)+1}=binary_pattern2dlts_weight(-[1 >>
>>  -1],transientlength);

     % Lock in: Only longest window
   %weights{length(weights)+1}=binary_pattern2dlts_weight(-[r >>
```





```
>> epmat(1,1,transientlength/2) repmat(-1,1,transientlength/2 >>
>> )],transientlength);window_offset=-1;elements_in_first_win >>
>> dow=2;

     % R2:
   %weights{length(weights)+1}=binary_pattern2dlts_weight(-[0 >>
>>  1 0 -1],transientlength);

     % R3:
   %weights{length(weights)+1}=binary_pattern2dlts_weight(-[1 >>
>>  0 0 -1],transientlength);

     % R4:
   %weights{length(weights)+1}=binary_pattern2dlts_weight(-[1 >>
>>  -1 -1 1],transientlength);

     % Double rectangular:
   %weights{length(weights)+1}=binary_pattern2dlts_weight([0  >>
>> -1 -1 1 1 1 1 0 0 -1 -1 -1 -1 1 1 0],transientlength);

     % Double rectangular: Only longest window:
   %weights{length(weights)+1}=binary_pattern2dlts_weight([re >>
>> pmat(0,1,transientlength/16) repmat(-1,1,transientlength*2 >>
>> /16) repmat(1,1,transientlength*4/16) repmat(0,1,transient >>
>> length*2/16) repmat(-1,1,transientlength*4/16) repmat(1,1, >>
>> transientlength*2/16) repmat(0,1,transientlength/16)],tran >>
>> sientlength);window_offset=-1;elements_in_first_window=16;

     % GS4:
   %weights{length(weights)+1}=binary_pattern2dlts_weight(-[- >>
>> 1 25 -48 24],transientlength);

    % GS4: Only longest window:
   weights{length(weights)+1}=binary_pattern2dlts_weight(-[re >>
>> pmat(-1,1,transientlength/4) repmat(25,1,transientlength/4 >>
>> ) repmat(-48,1,transientlength/4) repmat(24,1,transientlen >>
>> gth/4)],transientlength);window_offset=-1;elements_in_firs >>
>> t_window=4;

   %%%%%%%%%%%%%%%%%%%%%

   if window_offset==-1
    window_offset=log(transientlength/elements_in_first_windo >>
>> w)/log(2);
   end
```





Appendix D

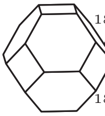
Example input file for the defect-reaction modeller

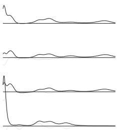
props_and_reactions.m

```
   % 2006 by Jan H. Bleka

   function[defect_properties,legendstr,dissociation,bonding, >>
>> R_bonding,k_B,elements_not_to_plot,filenames,path,annealin >>
>> g_temperatures_of_files,columns_to_import,data_map,convers >>
>> ion_factors,background,data_correction_factors]=props_and_ >>
>> reactions()

   % Don't plot these elements:
   elements_not_to_plot=[5 8 10 19]; % B Y Z X

   filenames=char('A061024a_MCz-29_VO_V2mm_165K_0-10640m.peak >>
>> ','A070412d_MCz-32_A070412a_and_b_peaks_fixed_with_A070412 >>
>> c_GS4_W5_and_W7_conc.peak');
   path=char('~/j/eksperimenter/raw/A/Selected_peak_files/');
   annealing_temperatures_of_files_C=[150 195]; % Celsius
   annealing_temperatures_of_files=annealing_temperatures_of_ >>
>> files_C+273.15; % K
   columns_to_import=[2 3 4;2 3 4]; % Sensitive to order. 1 ( >>
>> time) is ignored as it is always imported.
   data_map=[1 2 14]; % A map of which element each of the da >>
>> ta columns corresponds to. Column 1 is the first non-one n >>
>> umber in columns_to_import.

   conversion_factors=[5.0072e13 1]; % The factors to get fro >>
>> m concentration from the .peak files. 5.0072e13 for LI W6  >>
>> at a sample with N_d=5e12 cm^-3.
   background=1e10; % 24042007. This is usually the value i f >>
>> ind.

   files_to_exclude=[2]; % []. A simple way of not including  >>
```

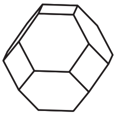
```
>> certain temperatures
   filenames=remove_rows(filenames,files_to_exclude);
   annealing_temperatures_of_files=remove_elements(annealing_ >>
>> temperatures_of_files,files_to_exclude);
   columns_to_import=remove_rows(columns_to_import,files_to_e >>
>> xclude);
   conversion_factors=remove_elements(conversion_factors,file >>
>> s_to_exclude);

   data_correction_factors=repmat(1,1,size(filenames,1)); % C >>
>> hange from 1 if individual data set should be scaled. For  >>
>> instance to account for different irradiation doses.

   % Physical constant:
   k_B=8.617386e-5; % eV/K

   %%%%%%%%%%%%%%%%%%%%%%%
   % DEFECT PROPERTIES
   defect_properties={}; % E.g. [VO_init(cm^-3) Dissociation: >>
>> C_0_VO(1/s) C_E_VO(eV) Diffusion:D_0_VO(cm^2/s) D_E_VO(eV) >>
>> ]
   legendstr={};

   % 1: VO
   legendstr{length(legendstr)+1}='VO';
   defect_properties{length(defect_properties)+1}=[-1 NaN NaN >>
>>  NaN NaN];

   % 2: V2
   legendstr{length(legendstr)+1}='V2';
   defect_properties{length(defect_properties)+1}=[-1 NaN NaN >>
>>  NaN NaN];

   % 3: O_i
   legendstr{length(legendstr)+1}='O_i';
   defect_properties{length(defect_properties)+1}=[2.5e17 NaN >>
>>  NaN NaN NaN];

   % 4: BH
   legendstr{length(legendstr)+1}='BH';
   defect_properties{length(defect_properties)+1}=[6.727e+16  >>
>> 1.3364e+11 2.6802e-07 NaN NaN]; % fit

   % 5: B
   legendstr{length(legendstr)+1}='B';
   defect_properties{length(defect_properties)+1}=[0 NaN NaN  >>
>> NaN NaN];

   % 6: H
   legendstr{length(legendstr)+1}='H';
   defect_properties{length(defect_properties)+1}=[1.8396e+07 >>
>>  NaN NaN 0.00056472 0.63003]; % fit

```

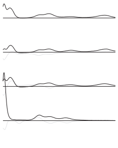
```
    % 7: YH
    legendstr{length(legendstr)+1}='YH';
    defect_properties{length(defect_properties)+1}=[5e13 NaN N >>
>> aN NaN NaN];

    % 8: Y
    legendstr{length(legendstr)+1}='Y';
    defect_properties{length(defect_properties)+1}=[0 NaN NaN  >>
>> NaN NaN];

    % 9: ZH2
    legendstr{length(legendstr)+1}='ZH2';
    defect_properties{length(defect_properties)+1}=[5e11 NaN N >>
>> aN NaN NaN];

    % 10: Z
    legendstr{length(legendstr)+1}='Z';
    defect_properties{length(defect_properties)+1}=[0 NaN NaN  >>
>> NaN NaN];

    % 11: H2
    legendstr{length(legendstr)+1}='H2';
    defect_properties{length(defect_properties)+1}=[0 NaN NaN  >>
>> 2.6e-4 0.78]; % From Markevich 1998

    % 12: VOH2
    legendstr{length(legendstr)+1}='VOH2';
    defect_properties{length(defect_properties)+1}=[0 NaN NaN  >>
>> NaN NaN];

    % 13: VOH
    legendstr{length(legendstr)+1}='VOH';
    defect_properties{length(defect_properties)+1}=[0 NaN NaN  >>
>> NaN NaN];

    % 14: VOH*
    legendstr{length(legendstr)+1}='VOH*';
    defect_properties{length(defect_properties)+1}=[-1 2.6252e >>
>> +12 1.4028 NaN NaN]; % fit

    % 15: V2H2
    legendstr{length(legendstr)+1}='V2H2';
    defect_properties{length(defect_properties)+1}=[0 NaN NaN  >>
>> NaN NaN];

    % 16: V2H
    legendstr{length(legendstr)+1}='V2H';
    defect_properties{length(defect_properties)+1}=[0 NaN NaN  >>
>> NaN NaN];

    % 17: VO2
    legendstr{length(legendstr)+1}='VO2';
    defect_properties{length(defect_properties)+1}=[0 NaN NaN  >>
>> NaN NaN];

   % 18: XH
   legendstr{length(legendstr)+1}='XH';
   defect_properties{length(defect_properties)+1}=[0 NaN NaN  >>
>> NaN NaN];

   % 19: X
   legendstr{length(legendstr)+1}='X';
   defect_properties{length(defect_properties)+1}=[6.7033e+16 >>
>>  NaN NaN NaN NaN]; % fit


   % REACTIONS
   dissociation={}; % [initial_element final_element1 final_e >>
>> lement2 final_element3...]. The order of the final element >>
>> s have no importance.
   % E.g.: V2O -> VO + V (V2O:defect_properties{1}, VO:2, V:3 >>
>> ): [1 2 3]
   % E.g.: V2 -> 2V (V2:1, V:2): [1 2 2] But this function is >>
>>  not tested!


   dissociation{length(dissociation)+1}=[4 5 6]; % H source 1
   dissociation{length(dissociation)+1}=[14 1 6]; % VOH* -> V >>
>> O + H

   % dissociation{length(dissociation)+1}=[7 8 6]; % H source >>
>>  2
   % dissociation{length(dissociation)+1}=[9 10 11]; % H2 sou >>
>> rce
   % dissociation{length(dissociation)+1}=[12 1 11]; % VOH2 - >>
>> > VO + H2
   % dissociation{length(dissociation)+1}=[13 1 6]; % VOH ->  >>
>> VO + H


   bonding={}; % [initial_element1 initial_element2 initial_e >>
>> lement3... final_element]. The first element is assumed to >>
>>  be the migrating one.
   % E.g.: VO + V -> V2O (VO:defect_properties{1}, V:2, V2O:3 >>
>> ): [1 2 3]
   % E.g.: 2V -> V2 (V:1, V2:2): [1 1 2] But this function is >>
>>  not tested!

   bonding{length(bonding)+1}=[6 1 14]; % H + VO -> VOH*
   bonding{length(bonding)+1}=[6 2 16]; % H + V2 -> V2H
   bonding{length(bonding)+1}=[6 19 18]; % H + X -> HX

   R_bonding={}; % Bonding radii and formation barriers. [rad >>
>> ius(cm) Formation barrier:C_0(1) E_b(eV)]. Pre-factor has  >>
>> no unit. 1 0 means no barrier.
   R_default=5e-8; % cm
```





```

   R_bonding{length(R_bonding)+1}=[R_default 0.017446 0.04300 >>
>> 3]; % fit
   R_bonding{length(R_bonding)+1}=[1.8093*R_default 0.023659  >>
>> 0.040163]; % fit
   R_bonding{length(R_bonding)+1}=[R_default 0.82918 0.079831 >>
>> ]; % fit
```





Bibliography

[1] W. Adam, B. Bellini, E. Berdermann, P. Bergonzo, W. De Boer, F. Bogani, E. Borchi, A. Brambilla, M. Bruzzi, C. Colledani, *et al.*, Nucl. Instrum. Methods Phys. Res. A **511**, 124 (2003)

[2] F. Gianotti, M. L. Mangano, T. Virdee, S. Abdullin, G. Azuelos, A. Ball, D. Barberis, A. Belyaev, P. Bloch, M. Bosman, *et al.*, European Physical Journal C **39**, 293 (2005)

[3] URL `http://care-hhh.web.cern.ch/CARE-HHH/LUMI-06`

[4] S. J. Pearton, Mater. Res. Soc. Symp. Proc. **59**, 457 (1986)

[5] J. Coutinho, O. Andersen, L. Dobaczewski, K. Bonde Nielsen, A. R. Peaker, R. Jones, S. Öberg, and P. R. Briddon, Phys. Rev. B **68**, 184106 (2003)

[6] K. Bonde Nielsen, L. Dobaczewski, K. Goscinski, R. Bendesen, Ole Andersen, and B. Bech Nielsen, Physica B **273–274**, 167 (1999)

[7] P. Pellegrino, P. Lévêque, J. Lalita, A. Hallén, C. Jagadish, and B. G. Svensson, Phys. Rev. B **64**, 195211 (2001)

[8] E. V. Monakhov, A. Ulyashin, G. Alfieri, A. Yu. Kuznetsov, B. S. Avset, and B. G. Svensson, Phys. Rev. B **69**, 153202 (2004)

[9] T. Zundel and J. Weber, Phys. Rev. B **39**, 13549 (1989)

[10] V. P. Markevich and M. Suezawa, J. Appl. Phys. **83**, 2988 (1998)

[11] J. Weber, S. Knack, O. V. Feklisova, N. A. Yarykin, and E. B. Yakimov, Microelectron. Eng. **66**, 320 (2003)

[12] Y. L. Huang, Y. Ma, R. Job, and A. G. Ulyashin, J. Appl. Phys. **96**, 7080 (2004)

[13] C. Herring and N. M. Johnson, J. Appl. Phys. **87**, 4635 (2000)

[14] V. P. Markevich and M. Suezawa, J. Appl. Phys. **87**, 4637 (2000)

[15] URL `http://en.wikipedia.org`

[16] S. D. Brotherton and P. Bradley, J. Appl. Phys. **53**, 5720 (1982)

[17] L. W. Song, X. D. Zhan, B. W. Benson, and G. D. Watkins, Phys. Rev. B **42**, 5765 (1990)

[18] N. Keskitalo, A. Hallén, J. Lalita, and B. G. Svensson, *Defects and Diffusion in Silicon Processing* (Materials Research Society, Pittsburg, 1997), vol. 469 of *Mater. Res. Soc. Symp. Proc.*, p. 233

[19] M. Moll, Ph.D. thesis, University of Hamburg (1999), DESY-THESIS-1999-040

[20] G. Lindström, M. Ahmed, S. Albergo, P. Allport, D. Anderson, L. Andricek, M. M. Angarano, V. Augelli, N. Bacchetta, P. Bartalini, *et al.*, Nucl. Instrum. Methods Phys. Res. A **465**, 60 (2001)

[21] URL `http://rd48.web.cern.ch`

[22] URL `http://rd50.web.cern.ch`

[23] G. Kramberger, D. Contarato, E. Fretwurst, F. Hönniger, G. Lindström, I. Pintille, R. Röder, A. Schramm, and J. Stahl, Nucl. Instrum. Methods Phys. Res. A **515**, 665 (2003)

[24] P. Pellegrino, Ph.D. thesis, KTH, Stockholm (2001)

[25] M. Mikelsen, Ph.D. thesis, University of Oslo (2007)

[26] G. Alfieri, Ph.D. thesis, University of Oslo (2005)

[27] P. Blood and J. W. Orton, *The Electrical Characterization of Semiconductors: Majority Carriers and Electron States* (Academic Press, London, 1992)

[28] M. Yoneta, Y. Kamiura, and F. Hashimoto, J. Appl. Phys. **70**, 1295 (1991)

[29] Y. Tokuda, Jpn. J. Appl. Phys. **37**, 1815 (1998)

[30] URL `http://en.wikipedia.org/wiki/Simulated_annealing`

[31] M. Moll, E. Fretwurst, M. Kuhnke, and G. Lindström, Nucl. Instrum. Methods Phys. Res. B **186**, 100 (2002)

[32] R. M. Fleming, C. H. Seager, D. V. Lang, E. Bielejec, and J. M. Campbell, Appl. Phys. Lett. **90**, 172105 (2007)