Eindhoven University of Technology

Eindhoven University of Technology

BACHELOR

Average distance between indium antimonide nucleation sites on an indium phosphide surface

van Wieringen, Simone

Award date:
2019

Eindhoven University of Technology
Department of Applied Physics
Advanced Nanomaterials & Devices

Average Distance between Indium Antimonide Nucleation Sites on an Indium Phosphide Surface

Bachelor Final Project

S. van Wieringen
s.v.wieringen.1@student.tue.nl

Supervisors:
E.P.A.M. Bakkers
J. Jung
R.L.M. op het Veld

Eindhoven, 1st November 2019

Abstract

In a quantum computer, defect free nanowires are required. Therefore, the properties of the growing process should be known. In this work, the average distance between the nucleation sites of in-plane selective area network was measured. This was done by creating in-plane nanowires. Only the nucleation sites were formed, which were made of indium antimonide. The measurements were performed on an indium phosphide surface. With a scanning electron microscope the samples were analysed. This was done for different V/III ratios and different crystal orientation. Moreover, the difference was compared between samples which were annealed and not annealed. When the V/III ratio is increased, the average distance between the nucleation sites increased. The average distance between the nucleation sites was larger for the samples which were annealed than for the samples which were not annealed. Moreover, there was no significant difference between the orientation directions of the lines. Lastly, the difference in surface plane was analysed. For the (111)A plane, the number of nucleation sites was more constant when changing the V/III ratio than in the (111)B plane.

Contents

Chapter 1

Introduction

In 1937 Ettore Majorana first wrote about a fermion which is its own anti-particle [1]. Right now, researchers use nanowires to detect these particles. A nanowire is a structure which has a small diameter and a large length. In this region quantum effects play a large role. A nanowire has a one-dimensional structure, which has different properties than a bulk material. In 1997 Alexei Kitaev first wrote about a topological quantum computer [2]. He proposed to use this Majorana fermion in a quantum computer. However, only in high quality nanowires without defects it is able to detect a Majorana particle [3]. Therefore defect-free nanowires are required.

A new way to make nanowires was recently proposed. With this method, it is possible to create the nanowire in a desired pattern, so the flexibility of the structure is increased. Furthermore, the structure will be grown within a mask, keeping the structure one dimensional with a controllable and constant cross-section. Lastly, it is easy to scale up this method, making it possible to produce plenty of nanowires at the same time [4].
In this work, the properties of the growth process of these nanowires are analysed. It is measured how many nucleation sites per unit length there were for different growth conditions. This is done on a substrate of indium phosphide. A substrate of indium phosphide in the (111)A plane has not been analysed before.

In chapter 2 the crystal structure is explained. In chapter 3 the process of growning nanowires is discussed. In addition, the experimental setup and procedure is explained. In chapter 4 the results of the average number of nucleation sites per unit length is presented. Moreover, the results of the (111)A and (111)B plane are compared and discussed in chapter 4 and 5 respectively.

Chapter 2

Theory

2.1 Crystal Lattice

Indium phosphide and indium antimonide both have a zincblende structure. This is shown in figure 2.1a. The lattice can be cut in two different directions, which is visualised in figure 2.1b. The different directions in which the material can be cut is in the $\langle 110 \rangle$, $\langle 112 \rangle$ and $\langle 111 \rangle$ direction, which is visualised in figure 2.2.

In figure 2.1b, it can be seen that for the (111) plane, either indium or phosphide is on top. If the first material is on top, an A is added to the direction. If the second material is on top, a B is added. So for indium phosphide (InP) if indium is on top, an A is added and if phosphide is on top, a B is added.
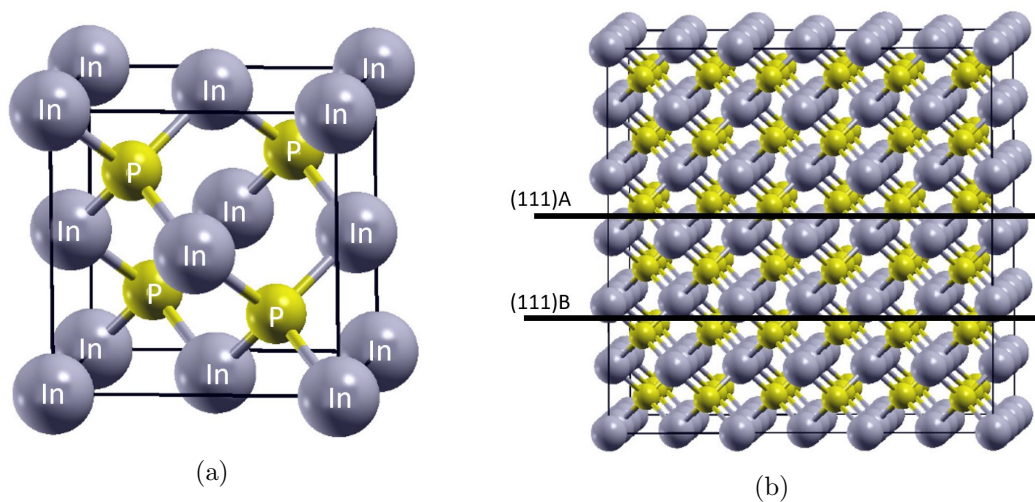

(a)

(b)

Figure 2.1: The lattice structures of InP. In figure 2.1a the zincblende structure is visualised. In 2.1b the zincblende structure is repeated and the larger structure is visible. Here, the difference in the (111)A and (111)B plane is visible. When the indium (purple) is on top, it is the (111)A plane. When the antimony (yellow) is on top, it is the (111)B plane. Image taken from reference [5].

On top of the indium phosphide, indium antimonide is grown. However, there is a mismatch between the lattice constant of the two materials. Indium phosphide has a lattice constant of 0.587 nm [6] and indium antimonide a lattice constant of 0.646 nm [7]. This gives a lattice mismatch of 10.4 %. Defects will form in the horizontal direction of the wire. The defects are scatter sites for electrons. When the nanowire is defect

free, it has a larger carrier mobility [8]. This is useful to use a nanowire in a topological quantum computer.
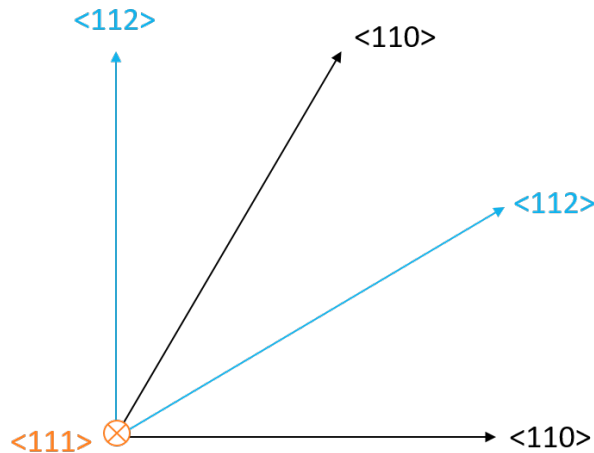

Figure 2.2: The directions within a lattice are visualised.

2.2 Nucleation Theory

The nucleation theory describes the formation for one phase to another phase. This concept can be used to support the experimental results with the theory.

The nucleation rate R is given by

$$R = C \, exp(\frac{\Delta G}{kT}) \tag{2.1}$$

were C is a constant, ΔG the difference in the Gibbs free energy, k the Boltzmann constant and T the temperature.

The Gibbs free energy as function of the radius is given figure 2.3. Here is visible that there is a radius for which the Gibbs free energy is maximum. A nucleation which is smaller as the critical radius is unstable and will fall back to its original phase. In this way, only nucleation sites larger than the critical radius will form.

The $\langle 110 \rangle$ direction has a lower surface energy as the $\langle 112 \rangle$ direction [9]. When the surface energy is high, it is more likely that a nucleation site is formed. In this way, it is expected that the number of nucleation sites per length in the $\langle 110 \rangle$ direction is higher than in the $\langle 112 \rangle$ direction.
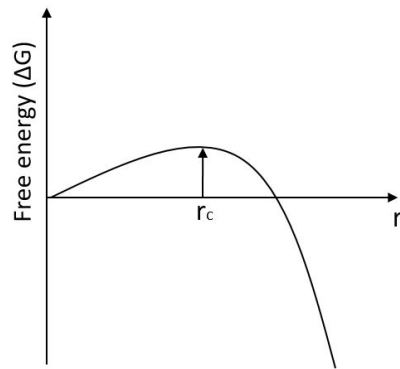
Figure 2.3: Gibbs free energy as function of the radius. Image taken from reference [10]

Chapter 3

Experimental Setup

3.1 Process of making in-plane nanowires

In this part, the process of creating in-plane nanowires is explained. The substrate which is used is indium phosphide (InP). On top of the substrate, silicon nitride (Si_xN_y) of 20 nm is added by using plasma enhanced chemical vapour deposition. On top of the silicon nitride, a resist will be applied. This is done by rotating the wafer when the polymer is added, so the resist will be uniformly distributed.

Using reactive ion etching, the silicon nitride can be removed in specific areas. This is shown is step 3 of figure 3.1. After making the desired pattern, the resist will be removed. This is done by a chemical lift off, followed by plasma ashing. This step is shown in step 4 of figure 3.1.

Part of the samples were annealed after the resist was removed. During the annealing process, the particles move within the lattice, lowering the number of dislocations in the material. The material will go to its equilibrium state, lowering the internal energy. The surface will be smoother, so the the probability that a nucleation site will form will be lowered.

Within the lines that where etched away, indium antimonide (InSb) will start growing on different nucleation sides. The indium antimonide is deposited on the wafer by metalorganic vapour-phase epitaxy. This process is done with different ratios of the Tri-Methyl-Indium (TMIn) and Tri-Methyl-Antimony (TMSb) pressure. This is called the V/III ratio. In this research, the TMIn is kept constant and the TMSb is changing. The indium antimonide will form nanowires by follow the direction of the lines. Indium antimonide spots will grow and eventually, two nucleation sites merge. The nuclei could have a 180 degree rotated crystallographic orientation with respect to each other. This affects the conductivity of the material. In consequence, it is tried to get as few nucleation sides as possible.
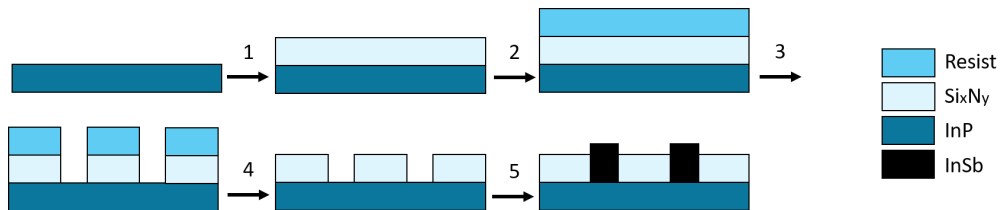

Figure 3.1: Process of making in-plane nanowires

3.2 Scanning Electron Microscope

A scanning electron microscope (SEM) of Zeiss is used to perform the measurements. With an electron beam the surface is scanned. This is done with an acceleration voltage of 3 kV.

Eight different samples were measured. Four different V/III ratios were measured and annealed and not annealed samples were compared. For every sample, the difference between the lattice orientations in the $\langle 110 \rangle$ and $\langle 112 \rangle$ direction was analysed. For every measurement, around ten to twelve lines are analysed. This is about 0.5 mm per measurement. Matlab is used to analyse the data. This Matlab file can be found in appendix B. The file gives the coordinates of the nucleation sites. After that, all images are checked to see if the Matlab file made any mistakes. With the coordinates of the nucleation sites, the distance between the nucleation sites is calculated.

In figure 3.2a an example of a measurement is shown. The lighter spots in the line are the nucleation sites. In figure 3.2b the processed image is shown. The coordinates are exported to a text file and then Excel is used to calculate the distances between the nucleation sites. With the data, the average number of nucleation sites per micrometre is calculated.

(a)

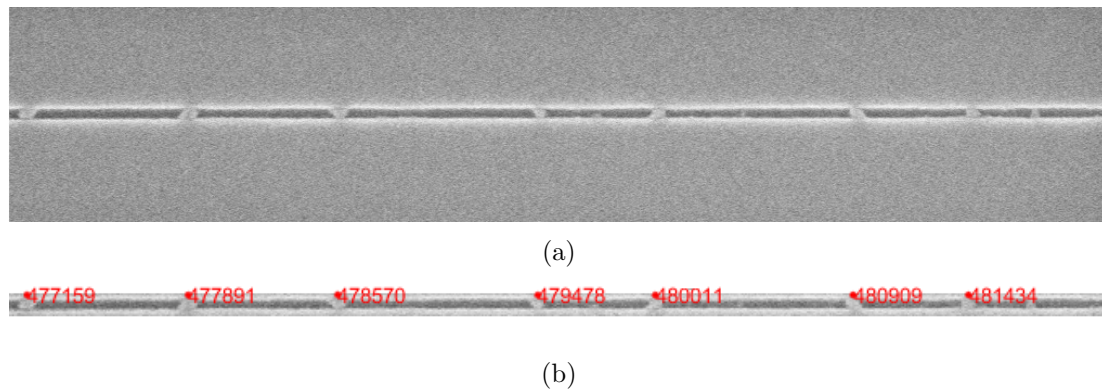

(b)

Figure 3.2: Image of an in-plane nanowire. In figure 3.2a the image made by a scanning electron microscope is shown. The lighter spots in the line are the nucleation sites. In figure 3.2b an image of the in-plane nanowire which is processed by Matlab is shown. This sample was not annealed and has a V/III ratio of 501. This measurement was preformed in the $\langle 112 \rangle$ direction.

Chapter 4

Results and Discussion

4.1 Number of Nucleation Sites

The average number of nucleation sites in one micrometre is calculated. The results are shown in figure 4.1. It can be seen that the higher the V/III ratio, the lower the number of nucleation sites per micrometre. So for higher V/III ratios, the nucleation probability is smaller than for lower V/III ratios. This is because the antimony decreases the surface energy on the indium phosphide [4]. So when there is more antimony, the surface energy is lower. The probability that a nucleation site will form is thus lower for higher V/III ratios.

Moreover, it can be seen that the annealed samples have fewer nucleation sites than the samples which were not annealed. This is because of a smoother surface after annealing with a lower internal energy, as written in chapter 3.1.

Lastly, there is a small difference between the $\langle 110 \rangle$ and $\langle 112 \rangle$ direction for the samples which were annealed. However, the error bars are still quite close together. Because the error bars give a 67% uncertainty, it is still possible that the measurements between the different orientations overlap. As written in 2.2 the $\langle 110 \rangle$ direction was expected to have a higher number of nucleation sites. However, there is no difference visible. So with this data, it is not possible to make a conclusion if there is, or if there is no difference between the orientations $\langle 110 \rangle$ and $\langle 112 \rangle$.

Furthermore, a histogram of the distribution of the distance between the nucleation sites was made. These distributions can be found in figures A.1 and A.2 in appendix A. Here it can be seen that for the lower V/III ratios, there is more a peak in the distribution visible. Especially when looking at the higher V/III ratios for the samples which were annealed, not distribution is visible anymore. This means that for the higher V/III ratios, the distribution is random.
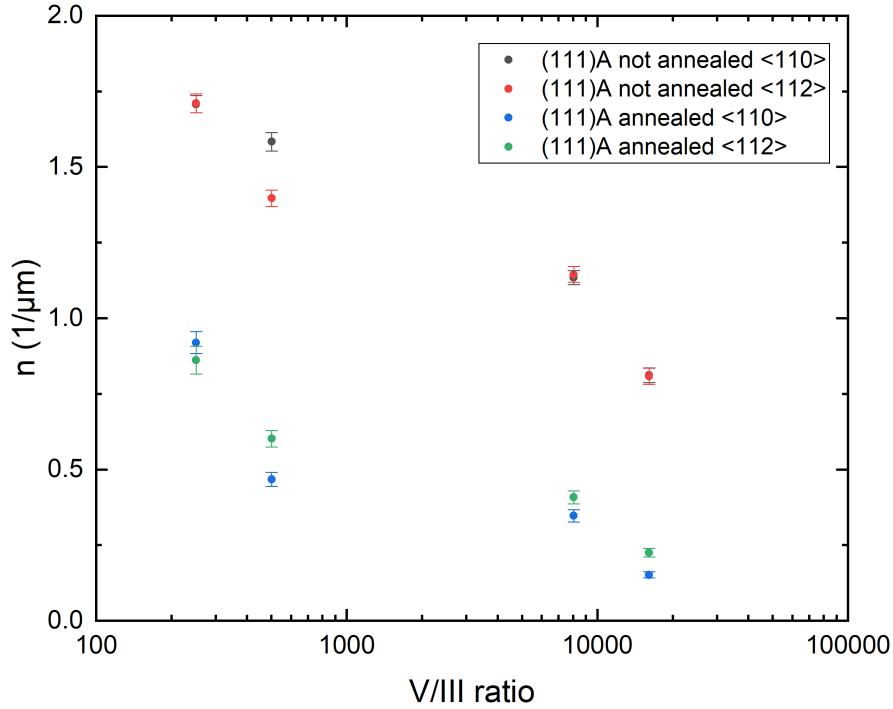
Figure 4.1: Number of nucleation sites per micrometre. The black and red dots are the not annealed measurements and the green and blue are the annealed measurements.

4.2 Difference between (111)A and (111)B plane

In figure 4.2 the number of nucleation sites per micrometre is shown for the (111)A and (111)B samples [4, 11]. It can be seen that for small V/III ratios, the (111)B samples have more nucleation sites. For larger V/III ratios, the (111)A samples have more nucleation sites. When increasing the V/III ratio, it can be seen that the (111)B samples have a large decrease in number of nucleation sites than for the (111)A samples. For the (111)A samples, the number of nucleation sites is more constant. For the (111)A samples indium is on top. When adding more antimony, part of the antimony will fit in the lattice to form a new layer in the (111)A. In this way, the Tri-Methyl-Antimony pressure is more constant for the (111)A samples than for the (111)B samples when increasing the Tri-Methyl-Antimony pressure.
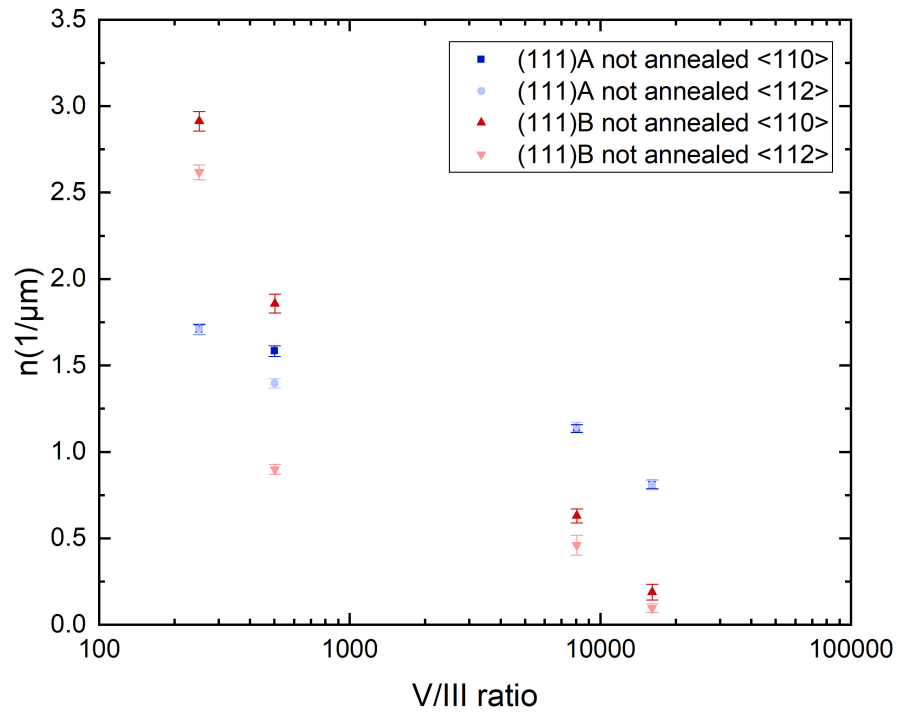
Figure 4.2: Number of nucleation sites per micrometre for different V/III ratios and for different substrates. The light and dark blue dots are the not annealed samples on a (111)A substrate. The light and dark red dots are samples on a (111)B substrate [4, 11].

Chapter 5

Conclusion

In this work, the distribution of the distance between the nucleation sites is measured. The measurements are done for different V/III ratios and different orientation angles. The higher the V/III ratio, the lower the number of nucleation sites per micrometre. This is because the antimony changes the surface energy on the indium phosphide.

Moreover, the number of nucleation sites per micrometre was lower for samples which were annealed compared to samples which were not annealed. This is because during the annealing process, the surface gets smoother. It is less likely for a nucleation site to form on a smooth surface compared to a rough surface.

Furthermore, it was seen that there was only a small difference in distance between the orientations $\langle 110 \rangle$ and $\langle 112 \rangle$ when the samples were annealed. However, the difference was too small to conclude if there is, or if there is no difference between the orientations.

This research has looked at the difference of the (111)A and (111)B substrate plane. It was seen that for the (111)A surface, the number of nucleation sites per micrometer was more constant when changing the V/III ratio than for the (111)B surface. This is because on the (111)A surface, part of the animony will form a new layer animony on the surface. This is because indium phosphide and indium animony have the same zincblende structure. In this way, the Tri-Methyl-Antimony pressure is more contant on the (111)A surface than on the (111)B surface.

Bibliography

[1] Ettore Majorana. Teoria simmetrica dell'elettrone e del positrone. *Il Nuovo Cimento*, 14(4):171–184, 1937.

[2] A. Yu. Kitaev. Quantum computations: algorithms and error correction. *Russian Mathematical Surveys*, 52(6):53–112, 1997.

[3] R. M. Lutchyn, E. P.A.M. Bakkers, L. P. Kouwenhoven, P. Krogstrup, C. M. Marcus, and Y. Oreg. Majorana zero modes in superconductor-semiconductor heterostructures. *Nature Reviews Materials*, 3(5):52–68, 2018.

[4] Roy L. M. op het Veld, Di Xu, and et al. In-plane selective area insb-al nanowire quantum networks. Unpublished.

[5] Peng, Qing, Chen, Huang, Danhong, Heller, Eric R., David A., Gao, Fei, and et al. First-principles assessment of the structure and stability of 15 intrinsic point defects in zinc-blende indium arsenide, Jan 2019.

[6] V. P. Vasil'Ev and J. C. Gachon. Thermodynamic properties of inp. *Inorganic Materials*, 42(11):1171–1175, 2006.

[7] Charles Kittel. *Introduction to solid state physics*. Wiley, 2004.

[8] M. D. Schroer and J. R. Petta. Correlating the nanostructure and electronic properties of inas nanowires. *Nano Letters*, 10(5):1618–1622, Dec 2010.

[9] Vitaly I. Kalikmanov. *Nucleation theory*. Springer, 2013.

[10] Jörg Polte. Fundamental growth principles of colloidal metal nanoparticles – a new perspective, Jun 2015.

[11] O.C.A. Bolkenbaas. Distribution of distances of insane grown insb nucleation sites. Eindhoven University of Technology, 2019.

Appendix A

Distributions of nucleation sites
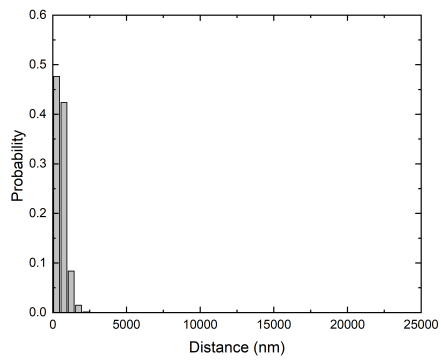
(a) V/III ratio 250.9.
$\langle 112 \rangle$.
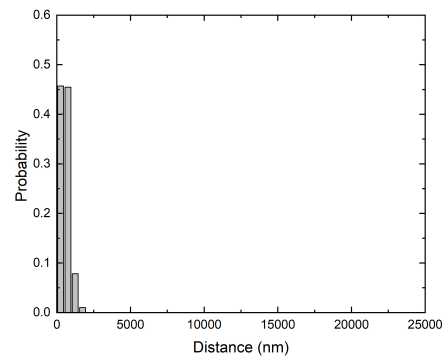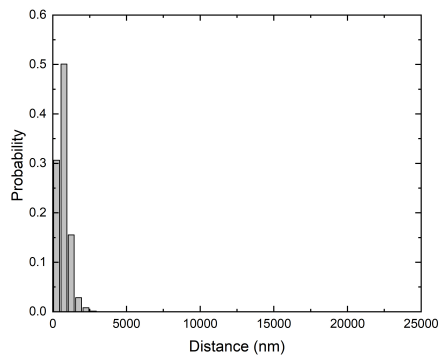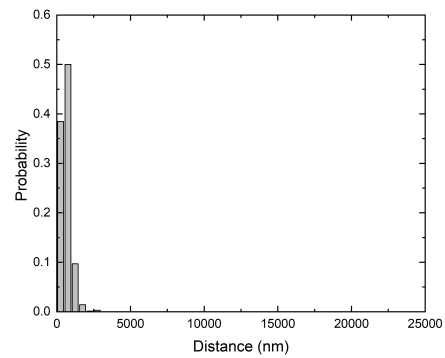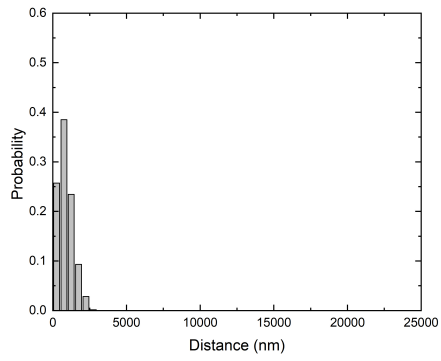
(b) V/III ratio 250.9.
$\langle 110 \rangle$.

(c) V/III ratio 501.8.
$\langle 112 \rangle$.

(d) V/III ratio 501.8.
$\langle 110 \rangle$.

(e) V/III ratio 8042.4.
⟨112⟩.
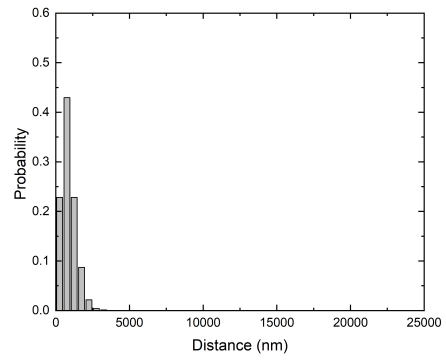
(f) V/III ratio 8042.4.
⟨110⟩.
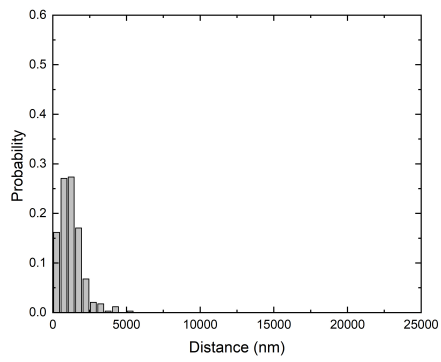
(g) V/III ratio 16084.7.
⟨112⟩.
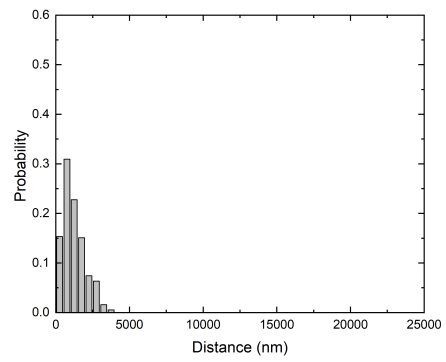
(h) V/III ratio 16084.7.
⟨110⟩.

Figure A.1: (111)A distribution for the not annealed samples.
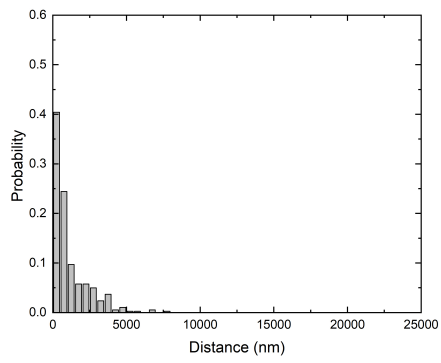
(a) V/III ratio 250.9.
⟨112⟩
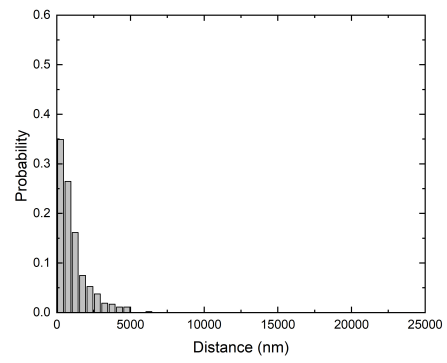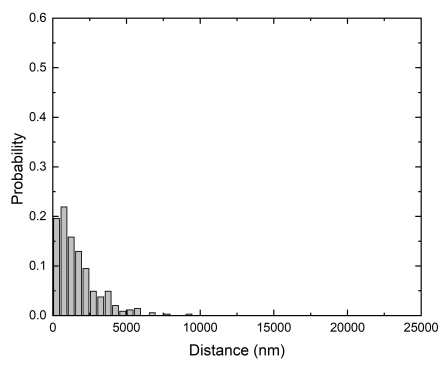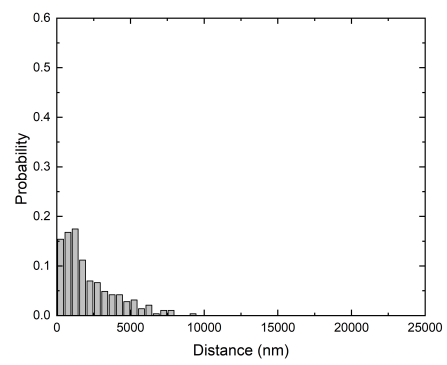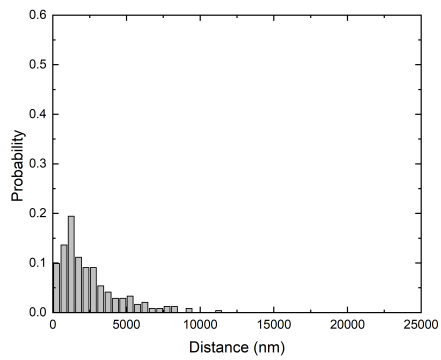
(b) V/III ratio 250.9.
⟨110⟩.

(c) V/III ratio 501.8.
⟨112⟩.

(d) V/III ratio 501.8.
⟨110⟩.

(e) V/III ratio 8042.4.
⟨112⟩.
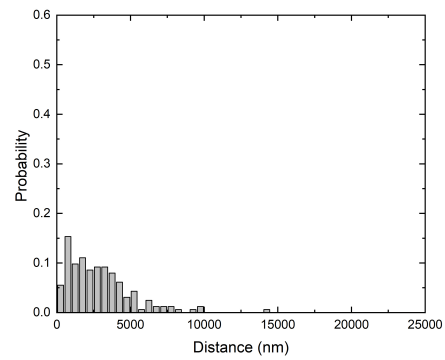
(f) V/III ratio 8042.4.
⟨110⟩.
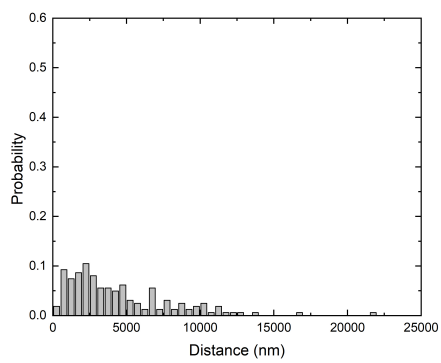
(g) V/III ratio 16084.7.
⟨112⟩.
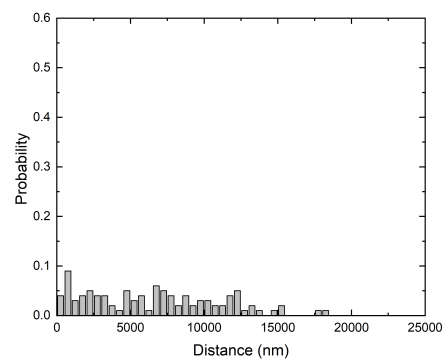
(h) V/III ratio 16084.7.
⟨110⟩.

Figure A.2: (111)A distribution for the annealed samples.

Appendix B

Matlab file

The Matlab file which is used to detect the nucleation sites is given below. The file is taken from reference [11] and improved.

```matlab
 1  clear;
 2  clc;
 3  close all;
 4
 5  %Open the tif files from the SEM. One trench at a time, then run the ...
        script
 6  %again for the next wire
 7  [file,path] = uigetfile('*.tif','Select One or More Files', ...
        'MultiSelect', 'on');
 8  filename=string(fullfile(path,file));
 9  Size=size(filename);
10  NumIm=Size(2); %number of images
11  Ndist=0; %initial number of distances between the nucleation sites
12
13  %Four variables used to improve working of the program a detailed
14  %explanation is given in the appendix of this BEP report
15  %    -rot for rotating the image
16  %    -sens for the sensitivity of the imbinarize function
17  %    -noise_limit for the minimum area of a nucleus in pixels
18  %    -res for the pixelsize of the image
19  rot=0;
20  sens=0.3;
21  noise_limit=30;
22  res=4.522;
23
24  size_total_pictures=0; %%start value
25
26  %Image analysis for-loop (repeated for every image):
27  %    -The user crops the image is cropped to the size of the trench in
28  %     which the nucleation sites can be found.
29  %    -Image is converted into a binary image
30  %    -Noise is reduced, single or small groups of white pixels are removed.
31  %    -The centres of the nucleation sites are found and shown in a plot.
32  %     This plot can be used to check the program
33  %    -The distances between the centres are computed and saved in a matlab
34  %     file so all distances from every image can be put in the same array
35  %     later
36
37  disp('The script is started at')
38  disp(datestr(now,'mm-dd-yyyy HH-MM'))
39  disp(filename(1,1))
40
```

18

```matlab
41   for p=1:1:NumIm %%the start of the large for-loop
42       %To make sure no errors occur when only one image is selected.
43       if NumIm==1
44           I=imread(filename);
45       else
46           fn=filename(1,p);
47           I=imread(fn);
48       end
49
50       Irot=imrotate(I,rot);
51
52       %%
53
54       image_imported=Irot;
55
56       image_cropped = image_imported(1:end-100,:);
57
58       image_size = size(image_cropped);
59
60       pixels_vertical = image_size(1);
61       pixels_horizontal = image_size(2);
62       clear image_size;
63
64       %% Vertical
65
66       horizontal_mid = round(pixels_horizontal/2);
67
68       white_pixels_vertical = [];
69
70       for i = 1:pixels_vertical
71           if image_cropped(i,horizontal_mid)==255 && ...
72                   image_cropped(i,horizontal_mid-1)==255 && ...
73                   image_cropped(i,horizontal_mid-2)==255 && ...
74                   image_cropped(i,horizontal_mid+1)==255 && ...
75                   image_cropped(i,horizontal_mid+2)==255
76
77               white_pixels_vertical = [white_pixels_vertical, i];
78           end
79       end
80
81       if length(white_pixels_vertical)<2 || length(white_pixels_vertical)>4
82           error('White pixels vertical')
83       end
84
85       boundary_upper = white_pixels_vertical(1);
86       upper = 1;
87       for i = 2:length(white_pixels_vertical)
88           if white_pixels_vertical(i) - white_pixels_vertical(i-1) == 1
89               boundary_upper = white_pixels_vertical(i);
90               upper = i;
91               break;
92           end
93       end
```

```matlab
94
95        boundary_lower = white_pixels_vertical(upper+1);
96        for i = upper+1:length(white_pixels_vertical)−1
97            if white_pixels_vertical(i+1) − white_pixels_vertical(i) == 1
98                boundary_lower = white_pixels_vertical(i);
99                break;
100           end
101       end
102
103       image_cropped = image_cropped(boundary_upper+1:boundary_lower−1,:);
104
105       %% Horizontal
106
107       image_size_vertical = size(image_cropped);
108
109       pixels_vertical = image_size_vertical(1);
110       pixels_horizontal = image_size_vertical(2);
111       clear image_size_vertical;
112
113       vertical_mid = round(pixels_vertical/2);
114
115       white_pixels_horizontal = [];
116
117       for j = 1:pixels_horizontal
118           if image_cropped(vertical_mid,j)==255 && ...
119                   image_cropped(vertical_mid−1,j)==255 && ...
120                   image_cropped(vertical_mid−2,j)==255 && ...
121                   image_cropped(vertical_mid+1,j)==255 && ...
122                   image_cropped(vertical_mid+2,j)==255
123
124               white_pixels_horizontal = [white_pixels_horizontal, j];
125           end
126       end
127
128       if length(white_pixels_horizontal)<2 || ...
                length(white_pixels_horizontal)>4
129           error('White pixels horizontal')
130       end
131
132       boundary_left = white_pixels_horizontal(1);
133       left = 1;
134       if white_pixels_horizontal(left+1) − white_pixels_horizontal(left) ...
                == 1
135           boundary_left = white_pixels_horizontal(left+1);
136           left = left+1;
137       end
138
139       boundary_right = white_pixels_horizontal(left+1);
140
141       image_cropped = image_cropped(:,boundary_left+1:boundary_right−1);
142
143       image_edges=edge(image_cropped,'canny',0.7);
144
```

```matlab
145       %%
146
147       image_size = size(image_edges);
148
149       pixels_vertical = image_size(1);
150       pixels_horizontal = image_size(2);
151       clear image_size;
152
153       sampling_sites = pixels_horizontal * [1/10 2/10 3/10 4/10 5/10 ...
              6/10 7/10 8/10 9/10];
154       sampling_sites = round(sampling_sites);
155
156
157
158       boundaries_total=[];
159       num_succes=0;
160       for i=1:length(sampling_sites)
161           boundaries=[];
162           for j=1:pixels_vertical
163               if image_edges(j,sampling_sites(i))==1
164                   boundaries = [boundaries,j];
165               end
166           end
167           if length(boundaries)==2
168               num_succes = num_succes+1;
169               boundaries_total(num_succes,1)=boundaries(1);
170               boundaries_total(num_succes,2)=boundaries(2);
171           end
172       end
173
174
175       hight_groeve_average=round(mean(mean(boundaries_total)));
176
177       image_cropped_line = image_cropped(hight_groeve_average-1:hight_ ...
              groeve_average+1,:);        %%%%%%%%%%dikte van de wire ...
              cropped%%%%%%%%%%%%%%
178       J=image_cropped_line;
179
180
181       image_cropped_line_groot = image_cropped(hight_groeve_average-10: ...
              hight_groeve_average+10,:);
182
183       %binarize the image for detection of the nuclei
184       se=strel('line', 300,0);
185       background=imopen(J,se);
186       I2=J-background;
187       I3=imadjust(I2);
188       bw=imbinarize(I3,'adaptive','Sensitivity',sens); %%hier wordt de ...
              afbeelding binair gemaakt, dus 0tjes of 1tjes
189
190       %Find all objects in the image and identify them
191       cc=bwconncomp(bw); %finds the white areas in the image
192       Tot=cc.NumObjects; %number of objects that were found
```

```matlab
193
194        %filter out the noise by removing the white areas with small areas.
195        %change 'noise_limit' to addapt the filtering
196
197        %Finding the white areas larger than the noise_limit variable.
198        nucleationdata=regionprops(cc,'basic');
199        areas = [nucleationdata.Area];
200        n_site = areas>noise_limit;
201        N=areas.*n_site;
202
203        %Define variables to be used in noise reduction for loop.
204        [m,n]=size(bw);
205        Ns=zeros(m,n);
206        NucleationSites=zeros(m,n);
207
208        %Noise reduction for-loop:
209        %   -Find the objects with an area equal to those in the array 'N',
210        %    therefore the areas larger than the 'noise_limit' (if-loop)
211        %   -Add all objects which meet this requirement to the image with
212        %    only nucleation sites.
213        for i=1:1:Tot
214            if areas(i)== N(i)
215                Ns = false(size(bw));
216                Ns(cc.PixelIdxList{i}) = true;
217            end
218            NucleationSites=NucleationSites+Ns;
219        end
220
221        %Find all nucleation sites in the new image and define their centres
222        %and the number of nucleation sites in the image.
223        cc2 = bwconncomp(NucleationSites);
224        nucleationcentre = regionprops(cc2,'Centroid');
225        NN = cc2.NumObjects; %number of nucleation sites
226
227
228 %%
229        %Create an array containing all the coordinates for the centres of ...
                the nucleation sites
230        Centre=zeros(NN,2);
231        for j=1:1:NN
232            Centre(j,1)=nucleationcentre(j).Centroid(1); %%%%%%%%%% ...
                    x-coridinaat voor de nucleation site
233            Centre(j,2)=nucleationcentre(j).Centroid(2); %%%%%%%%%% ...
                    y-coridinaat voor de nucleation site
234        end
235
236
237
238        x_coordinate_nucleationsites_in_image=Centre(:,1);
239        size_one_picture_J= size(J,2);
240        x_coordinate_nucleationsites_total= ...
                x_coordinate_nucleationsites_in_image + size_total_pictures;
241        size_total_pictures=size_total_pictures+size_one_picture_J
```

```matlab
242
243        %Figure showing the final black and white image and the calculated
244        %centres as red stars next to the grayscale image with the same centre
245        %overlays.
246        %These can be used to see whether nucleationsites and their centres
247        %were found correctly.
248        fig=figure('visible','off');
249        %figure
250            subplot(2,1,1), imshow(NucleationSites)
251            hold on
252                    plot(Centre(:,1),Centre(:,2),'r.')
253            hold off
254                    title('Nucleation sites in black and white')
255            subplot(2,1,2), imshow(image_cropped_line_groot)
256            hold on
257                    plot(Centre(:,1),Centre(:,2)+5,'r.')
258            hold on
259                    text(Centre(:,1),Centre(:,2)-10, num2str(round(res* ...
                            x_coordinate_nucleationsites_total)), ...
                            'FontSize',6,'Color','red')
260
261            hold off
262        title('Nucleation sites in grayscale')
263
264
265
266        image_name=strcat(filename(1,p),'matlab_img.png');
267        saveas(gcf,image_name) %%file opslaan
268        %%
269
270        d_pix=zeros(NN-1,1);
271
272        %calculate distance in pixels using pythagoras. Then saving these
273        %values into a matlab file (will be overwritten if the script is run
274        %again)
275        for k=1:1:NN-1
276            d_pix(k)=sqrt((Centre(k,1)-Centre(k+1,1)).^2+ ...
                    (Centre(k,2)-Centre(k+1,2)).^2);
277        end
278
279        fnm = sprintf('file_%d.mat',p);
280        save(fnm,'x_coordinate_nucleationsites_total', 'NN')
281
282        %Display the results per image
283        %disp(['In image ', num2str(p),' ', num2str(NN), ' nucleation ...
                sites have been found.'])
284        %disp('The distances in nm are given by ')
285        %disp(round(d_pix)*res)
286
287
288        SizeD=size(d_pix);
289        Ndist=Ndist+SizeD(1);
290
```

```matlab
291  end %the end for the large for loop
292
293  %Distances are put in the same array and then rounded of to integer values
294  %for pixels. Then the resolution is used to compute the distances in nm.
295  %These are then saved into an ASCII formatted txt file named using the
296  %current date and time so no old data will be overwritten.
297
298  AllD=zeros(Ndist,1);
299  Nd=zeros(NumIm+1,1);
300  Nc=0;
301
302  for n=1:1:NumIm
303      fnm = sprintf('file_%d.mat',n);
304      load(fnm)
305      Nd(n+1)=NN;
306      Nc=Nc+Nd(n);
307      for m=Nc+1:1:Nd(n+1)+Nc
308          AllD(m)=x_coordinate_nucleationsites_total(m-Nc);
309      end
310  end
311
312  AllD_nm=res*round(AllD);
313
314  save(sprintf('distances2_%s.txt', datestr(now,'mm-dd-yyyy HH-MM')), ...
315      'AllD_nm', '-ascii')
316
317  disp('The script is finished at')
318  disp(datestr(now,'mm-dd-yyyy HH-MM'))
319  disp(filename(1,1))
```