THE PENNSYLVANIA STATE UNIVERSITY

DEPARTMENTS OF ENGINEERING SCIENCE AND MECHANICS AND ELECTRICAL ENGINEERING

Exploration of novel heterostructure semiconductors to create energy efficient, tunnel-based Field Effect Transistors

Chad A. E. Ostrowski

Fall 2009

a thesis
submitted in partial fulfillment
of the requirements
for baccalaureate degrees
in Engineering Science and Electrical Engineering
with interdisciplinary honors in Engineering Science and Electrical Engineering

Reviewed and approved* by the following:

Suman Datta
Associate Professor of Electrical Engineering
Thesis Supervisor

Gary Gray
Associate Professor of Engineering Science and Mechanics
Honors Advisor

Timothy Kane
Professor of Electrical Engineering
Honors Advisor

Judith A. Todd
P. B. Breneman Department Head Chair
Professor, Department of Engineering Science and Mechanics

*Signatures are on file in the Schreyer Honors College and Engineering Science and Mechanics Offices.

Abstract

A simple program was developed to calculate device characteristics of a reverse-biased, heterojunction InGaAs diode. This calculator allows for quick estimates of ideal device currents given various design parameters, as well as allowing experimental results to be plotted along with ideal results for comparison. Additionally, the calculator allows for the estimation of series resistance inherent in an actual device due to making contact to it. This calculator uses only software freely available to the general public and can be accessed from any place with an internet connection.

Table of Contents

Introduction

As it has seemed for the past two decades, the end of Moore's Law could be a mere five years away. Should this happen, computers will not get old, software developers will grow bored of the same hardware, the general populace will begrudge the fact that they will never be able to back up their brains to a computer, and millions of jobs will be lost. Accordingly, the Datta Group has taken up the task of exploring heterojunction, tunneling field-effect transistors, which we believe will sustain Moore's Law for the next ten years.

Currently, in order to test proposed device compositions, a large, expensive application such as Sentaurus must be used. While effective, it is cumbersome to utilize this power on a day-to-day basis. There are two measurements for which a faster approximation would be helpful, even if it were moderately less accurate: 1) matching experimental results to ideal results and 2) approximating the resistance occurring from device contacts.

A transistor essentially behaves as a diode when in operation. Tunneling FETs, in particular, behave as reverse-biased Zener diodes, with electrons tunneling from the valence band of the heavily-doped p side to the conduction band of the heavily doped n side. However, this simple model must have a resistance added to it in order to accurately model a real device, as making contact to the device introduces resistance into the system.

While the entire device is to be made of Indium Gallium Arsenide, InGaAs, the p side (drain) and the n side (source) will each have different proportions of indium and gallium. Indeed, these proportions, along with the amount of doping, form the bulk of what must be refined in order for TFETs to be implementable on VLSI logic chips.

Objectives

While Simulink and Sentaurus adequately compute characteristics for any semiconductor device, using them routinely throughout a process can become cumbersome. It would therefore be beneficial to have a smaller, more light-weight calculator to use explicitly for calculations involving indium gallium arsenide diodes.

Additionally, a homojunction InGaAs diode can be modeled entirely analytically; adding support for a heterojunction requires numerical methods only for solving the Fermi-Dirac integral. Thus, a light-weight, accessible calculator ought to be acceptably accurate for use in experimental verification.

My task was to build this simple InGaAs-diode reverse-bias calculator. Then while professors, Ph.D. students, and post-docs fabricate devices, they will have a fast means of determining the amount of contact resistance inherent in their measurements, as well as a quick way to verify what small parameter tweaks could boost performance.

Additionally, the calculator ought to be engaging enough that researchers enjoy it. This will aid in device discovery. Likewise, it ought also be accessible, so that no expensive, slow-to-open programs need be opened in order to use it.

Literature Review

When William Kamkwamba, a Malawian boy who built a windmill from junkyard-parts using diagrams found in library books, first had Google explained to him, he asked that they do a search for "windmill." His reaction, when seeing the millions of results generated in a fraction of a second, was "Where was this Google all this time?" (Szczys, 2009). Now he is working hard to bring reliable energy—and that vast compendium of human knowledge, the internet—to those throughout Africa who frequently do not even have access to a library, as he did. (Kamkwamba)

In the past twenty years, as the internet has exploded into popular use, it has been fueled by the hardware that makes it possible. The foundation of all of this hardware is the ability to perform logic operations with a switch. The Harvard Mark I computer made in 1944 used actual mechanical switches to this end, and was said to sound like a room "full of ladies knitting." (Alfred, 2008)

However, real progress in the field of digital knowledge-storing only began with solid-state ("solid state" means "no moving parts") switches, called transistors. In 1947, the first transistor was made by John Bardeen, Walter Brattain and William Shockley at Bell Labs (Massey). As seen in Figure 1, this device was little better than a physical switch in terms of space-saving, and furthermore, it was used to replace vacuum tubes as an amplifier, not as a switch. But this novel device lead to the first Metal-Oxide-Semiconductor Field-Effect Transistor (MOSFET) in 1960 (Massey).

In 1965, Gordon Moore proposed his famous law: in order to remain competitive in the component market, chip manufacturers must double the number of transistors on a single chip

every eighteen months (Intel). As the company he worked for, Fairchild Semiconductor, seemed reluctant to make this happen, Moore split away in 1967 to begin what would become Intel (Bellis).

Figure 1: Replica of the first transistor (Massey)

The first chip produced by Intel, the 4004 chip, contained approximately 2000 transistors and was used as a component in calculators in Japan (CNET News.com, 2006). Holding true to his creed, by 1985 the 386 chip had more than 100 times as many transistors as the 4004, the Pentium 4 had 100 times that (42 million) in 2000, and by the end of 2011, Intel will release a chip (Ivy Bridge) with 100 times the power of the Pentium 4, with 2.9 billion transistors. This progression is shown in Figure 2, the numerical trend is shown in Figure 3.

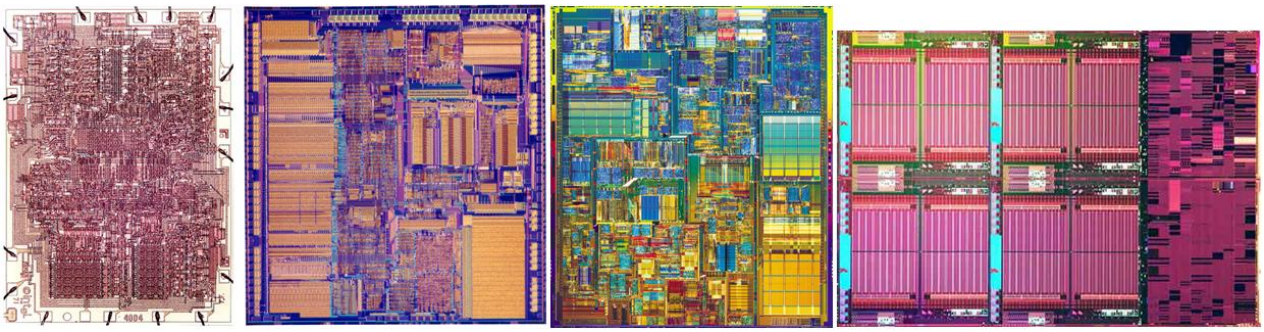


Figure 2: Moore and more transistors. Progression from the 4004 to the 386 to the Pentium 4 to the Ivy Bridge chip. First three images from CNET News.com, the Ivy Bridge picture comes from TechTree, 2009.

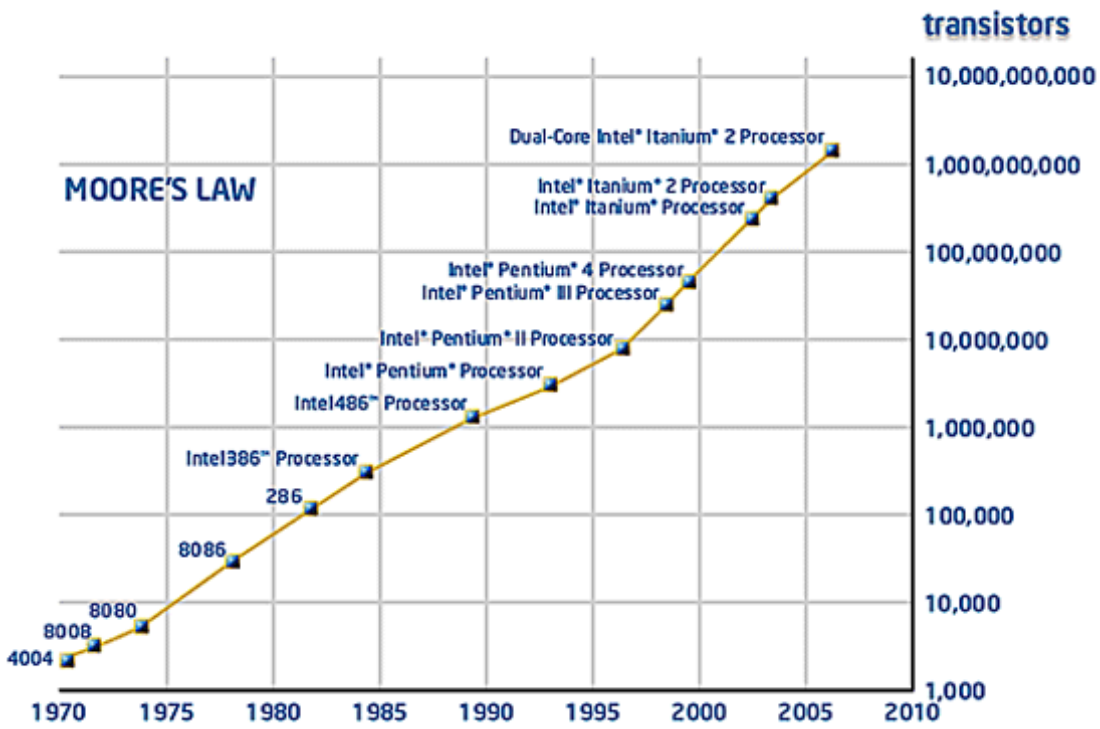


Figure 3: Number of transistors on a chip vs. time (Van Mateer, 2009)

Figures 2 and 3 show the technological explosion that now allows us to find graphs of Moore's law stored in Japan on a whim. This massive shrinking of logic technology has been accompanied by incredible growth in many fields. Processors are more powerful, storage is more massive, displays are larger, communications networks are more robust, medical procedures are safer, and the daily tasks performed in every industry have become streamlined.

However, these figures gloss over the immense amount of work that goes in to producing every new iteration of logic technology. For the past decade or more, worried scientists have been predicting the imminent end of Moore's Law (Datta). But every theorized physical limit on how small a transistor can be has been methodically proven inaccurate.

While transistors have shrunken and more have been put on each processor, they have not only crunched through more calculations, they also demand more and more power. As shown in Figure 4, some of this power use goes to leakage currents. Leakage currents occur when the width of an element in the device (here the power is plotted against the gate length, the distance between the source and the drain, explained on page 8) becomes so small that quantum mechanical tunneling occurs. These factors are obviously important and must be controlled. Indeed, the SiO_2 leakage (the purple part of the bars) has been overcome for the time being by replacing the traditional Silicon Dioxide (SiO_2) gate with Hafnium Dioxide. (An aside: hafnium, though much more expensive a material than silicon, has been incorporated in mass-manufactured processors. This teaches an important lesson: more expensive materials are economically sensible if they continue the trend of Moore's Law.) However, as is apparent in Figure 4, the largest use of power goes to the actual operation of the device, the Active power. While each single transistor uses very little power, when more than three billion are put onto a single chip, their power use is far from negligible.

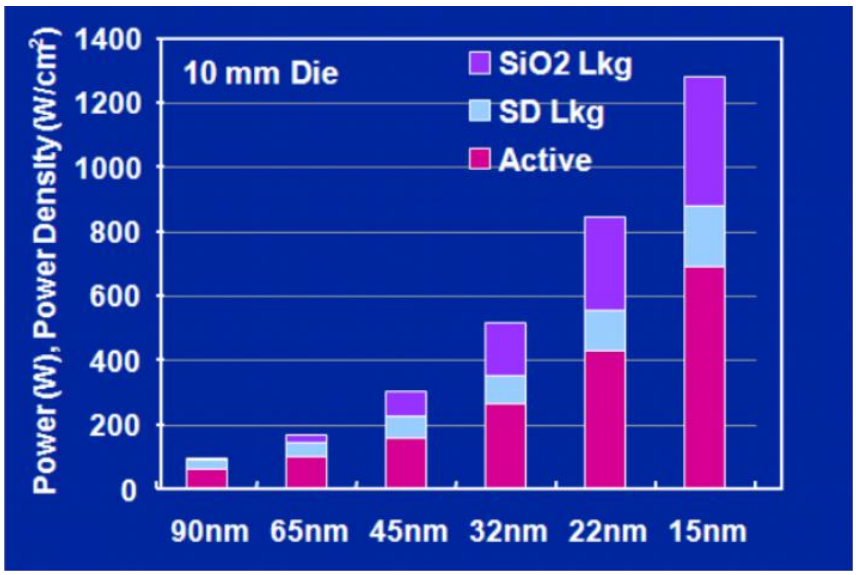


Figure 4: Power use in transistors as a function of their channel length. Courtesy of Datta et al., 2008.

Whereas a mechanical switch can be completely opened, so that no current can pass through, a solid-state device never completely turns off. However, in order to process logic operations, the difference between on-current (I_{on}) and off-current (I_{off}) must be significant. This relationship between I_{off} and I_{on} is shown in Figure 5.

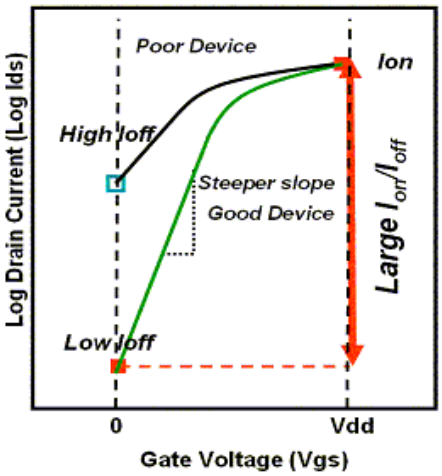


Figure 5: on/off ratio fundamentals, from eet-china.com.

The Active power use shown in Figure 4 is a product of large I_{on} and V_{dd}. However, simply lowering I_{on} in traditional MOSFETs causes two problems: the I_{on}/I_{off} ratio becomes too small, making discernment between an on-state and an off-state more difficult to decipher; and more problematically, the speed with which a device is recognized as being turned on (the switching speed) is controlled by I_{on}. That is, high I_{on} must be maintained in order to maintain switching speed. Thus, in order to reduce power use, significant I_{on}/I_{off} ratios must be achieved over a smaller voltage swing. The goal is to lower V_{dd} while maintaining the same I_{on}.

Traditional MOSFETs have been lowered to the smallest sensible V_{dd}, at just above 1 Volt. However, a paradigm shift to tunneling FETs (TFETs) would allow for a V_{dd} as low as 0.25V, as shown in Figure 6. This necessitates a move to more novel materials, as well, also highlighted in Figure 6.

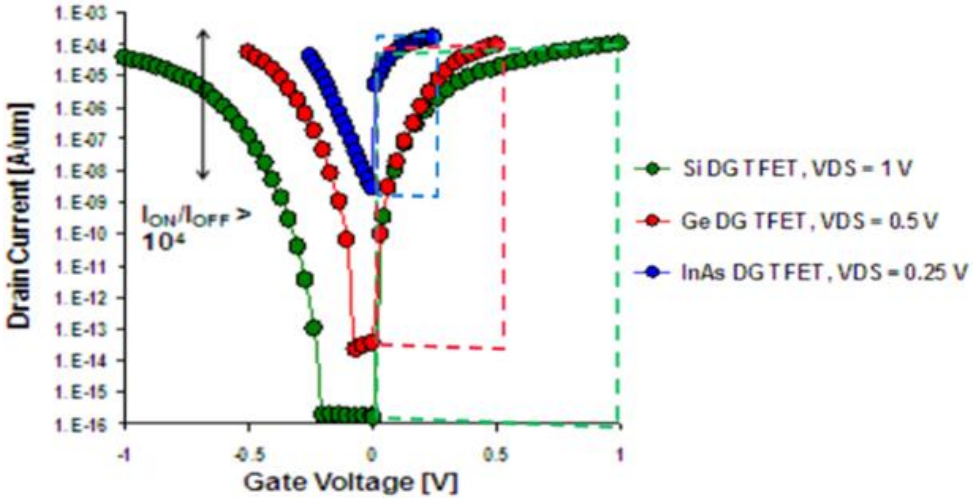


Figure 6: Tunneling Transistors provide higher Ion-Ioff ratio over a smaller voltage swing than traditional MOSFETs. Courtesy of Suman Datta.

Tunneling transistors are a big paradigm shift from traditional silicon MOSFETs not only because of their novel materials, but also because of the way in which they function. A traditional MOSFET is shown in Figure 7, along with band diagrams of the channel. This is contrasted with the TFET shown in Figure 8.

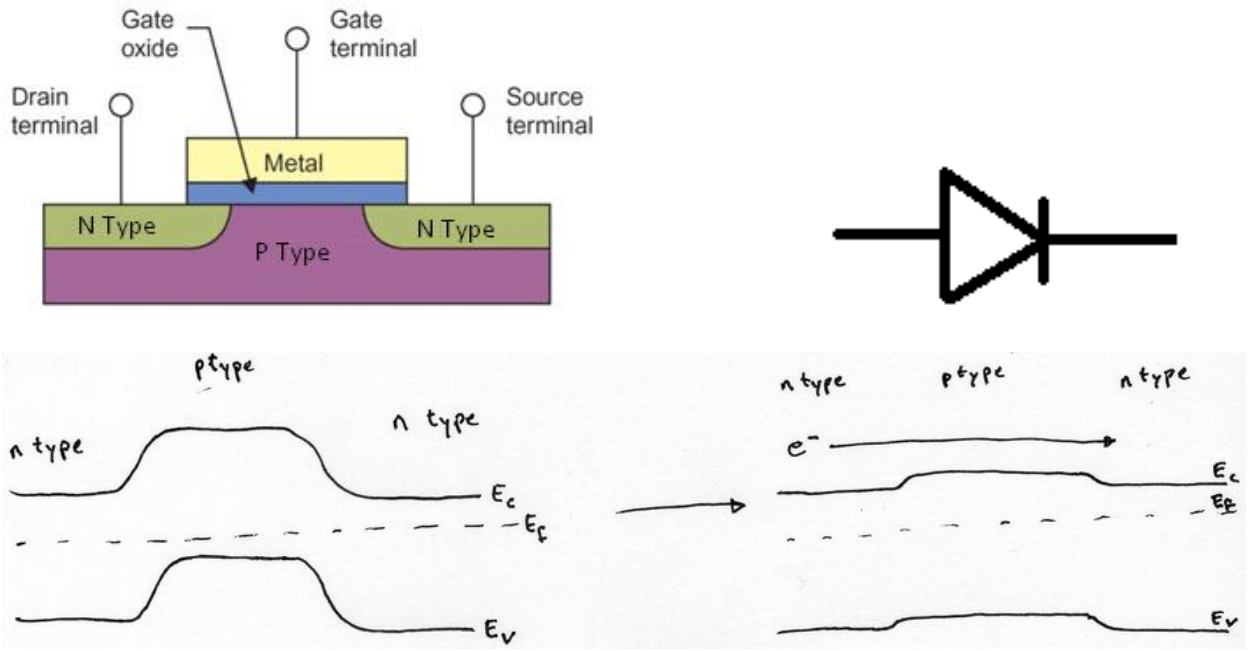


Figure 7: Basic operation of a MOSFET. It should be noted that a MOSFET is essentially a forward-biased pn junction. MOSFET diagram image courtesy of Först et al.

As shown in Figure 7, a MOSFET is a symmetric device: the source and the drain are doped identically. This essentially means that the source and the drain are interchangeable. No matter which side is being used as the source, the device always behaves a simple pn diode when on.

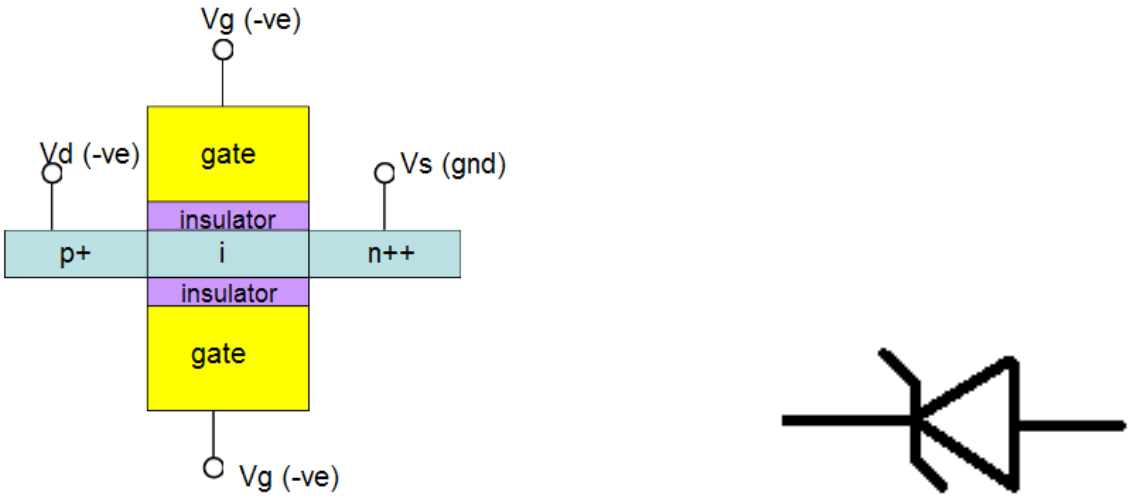


Figure 8: Basic operation of a TFET. It should be noted that a TFET is essentially a schottky diode, operating in the reverse-voltage scheme. TFET diagram image courtesy of Datta et al, 2008.

In contrast, as shown in Figure 8, a TFET is not symmetric. The drain is p+ while the source is n++. A shift to using TFETs in future processors will require a change in the circuit design of the processors; that is, circuit designers must learn the new paradigm. Fortunately, it has been shown that the 6-transistor dynamic memory configuration that holds such a

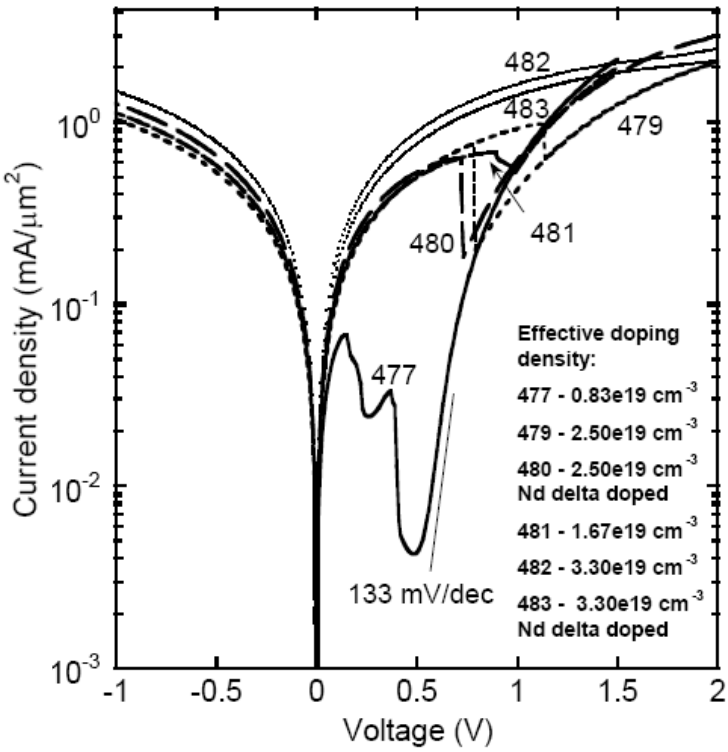
prominent place in MOSFET circuit design is still, in fact, implementable with TFETs (Datta). Despite these immense paradigm shifts away from symmetric devices and away from silicon, the ability to double the number of transistors on a chip will always be economically sensible, as was shown by the transition from a silicon gate to a hafnium gate (Datta).

Process

In the fall of 2008 when I first began what would become this project, Sajid Kabeer and Alan Seabaugh, researchers at the University of Notre Dame, had recently demonstrated the first functioning InGaAs tunnel diodes. A slideshow they had created, simply called InGaAs Tunnel Junctions, was my first in-depth introduction to the topic.

Verifying the tunneling current density equation

My first assignment was to plot InGaAs junction characteristics using equations from the slideshow created by Kabeer and Seabaugh. This could then be used to verify their equations, by comparison with the plots they obtained experimentally, shown in Figure 9.



	x 10^{19} cm^{-3}			V_P	J_P
Growth	N_A	N_D	N_{eff}	mV	mA/μm^2
477	5	1	0.91	165	1
479	5	5	1.58	445	20
480*	5	7	1.71	445	20
481	10	2	1.29	300	10
482	10	5	1.83	585	40
483*	10	7	2.03	585	40

* Si δ-doping, 1×10^{13} cm^{-2} every 1 nm, $10x + 5 \times 10^{13}$

Figure 9: characteristics of InGaAs junctions (Kabeer & Seabaugh, 2008)

Having built simple homojunction diodes with $In_{.53}Ga_{.47}As$, and having run all experiments at a temperature T of 300 Kelvin, their analytic model was developed as follows.

The only independent variables are the level of doping on each side, N_A and N_D (the amount of doping on the p side and the n side, respectively). An effective doping within the device, N_{eff}, can then be calculated in the familiar way:

$$N_{eff} = \frac{N_A N_D}{N_A + N_D} \tag{1}$$

This can then be used to find the electric field across the junction:

$$E = \sqrt{\frac{2qN_{eff}E_g}{\varepsilon_o \varepsilon_r}} \tag{2}$$

where q=1.602x10^{-19} Coulombs, the elementary charge; E_g, the bandgap of $In_{.53}Ga_{.47}As$ at 300K, is 0.74; ε_o is the permittivity of free space, ε_o=8.85418782 × 10^{-12} m^{-3} kg^{-1} s^4 A^2; and ε_r is given by NSM Archive as 13.9 for this composition and temperature of InGaAs.

This, equation, however, was discovered to contain an error: the electric field across a junction is of course dependent on the effective voltage across the junction (V_p= V_{bi}+V_r, where V_r is the reverse applied voltage and V_{bi} is the built in voltage), but the equation above does not have that factor. Plots using this equation gave bad results, but changing the bandgap factor (E_g) to V_p solved the problem agreeably. This required calculating the built in voltage of the junction, V_{bi} in the usual way:

$$V_{bi} = V_t \ln\left(\frac{N_A N_D}{n_i^2}\right) \tag{3}$$

which makes use of V_t = 26mV and , at 300K for $In_{.53}Ga_{.47}As$, n_i = 6.3x10^{11} (NSM Archive). Thus, equation (2) becomes

$$E = \sqrt{\frac{2qN_{eff}V_p}{\varepsilon_o \varepsilon_r}} \tag{4}$$

Finally, all of these values were plugged into the following equation to give the tunneling current density:

$$J_p = \frac{\sqrt{2m^*}q^3 E V_r}{4\pi^3 \hbar^2 \sqrt{E_g}} exp\left(\frac{-4\sqrt{2m^*}E_g^{\frac{3}{2}}}{3q\hbar E}\right) \quad (5)$$

In this equation, rather than attempting to calculate m*(the effective electron mass in the device) given the input parameters, we decided to make m* another input. This was due to the decidedly complicated nature of calculating m*. To our credit, the program seemed to work for the reasonable value of m*=0.042, which is the value given by the NSM Archive for 300K and N_D=2x10^{17}.

In order to plot this, a simple MATLAB function was used and an array of values for J_p was made for a whole range of voltages, V_r. The former was then plotted on a vertical log axis and the latter on a horizontal linear axis. However, comparison with the experimental results shown in Figure 9 was difficult, so a program called Data Thief was used to turn the 482 plot-line into raw data. This was then plotted on the same chart as the theoretical results. This revealed that the above series of equations were acceptably accurate. The only discrepancy occurred for large values of V_r, which was due to series resistance inherent in the experiment but not modeled in the equations. However, at this point in the process we were not concerned with modeling the series resistance.

Generalizing to arbitrary temperature and composition

After verifying that this simple series of equations worked, the goal was to generalize the MATLAB code to work for different temperatures and compositions of $In_{1-x}Ga_xAs$. The

parameters affected by these generalizations were E_g, ε_r, V_t, and n_i. These were given the following equations, in which x is the fraction of gallium:

$$E_g(x,T) = 0.42 + 0.625x - \left(\frac{5.8}{T+300} - \frac{4.19}{T+271}\right) \cdot 10^{-4}T^2x$$

$$-4.19 \times \frac{10^{-4}T^2}{T+271} + 0.475x^2 (in\ eV) \quad (6)$$

$$\varepsilon_r(x) = 15.1 - 2.87x + 0.67x^2 \quad (7)$$

$$n_i(N_c, N_v, T) = \sqrt{N_c N_v} \exp\left(-\frac{E_g}{2kT}\right) (in\ cm^{-3}) \quad (8)$$

all from NSM Archive, and the common equation

$$V_t(x) = \frac{kT}{q} \quad (9)$$

In equations (8) and (9), k is the Boltzmann constant, k= 1.38×10^{-23}J/K. It will also be noticed that the equation for n_i requires the calculation of N_c and N_v, the density of states in the conduction band and the valence band, respectively. These are given (also from the NSM Archive) by the following:

$$N_c(x,T) = 4.82 \times 10^{15}(0.023 + 0.037x + 0.003x^2)^{\frac{3}{2}}T^{\frac{3}{2}} \quad (10)$$

$$N_v(x,T) = 4.82 \times 10^{15}(0.41 - 0.1x)^{\frac{3}{2}}T^{\frac{3}{2}} \quad (11)$$

Plotting against electric field

After implementing all of these, the original data still fit the theoretical plots well. However, using new data would give inexplicable mismatches. Part of the problem, suggested

Professor Datta, could be the fact that we were plotting against voltage rather than electric field.

Plotting against electric field was a simple task for the ideal results, as the electric field was being calculated at every step already. Plotting the experimental results against electric field required calculating an electric field for every voltage value that had been measured. This was done using the following method, given in pseudo code:

```
for j from 1 to numberOfVoltagePointsInFile
    Vp(j)=ExperimentalVoltage(j)+Vbi;
    E(j)=sqrt(2*q*(Neff)*(Vp(j))/(eo*er));
end
```

After implementing this, there was seen better matching between the ideal results and the experimental results.

Generalizing to a heterojunction

After achieving all of this, my next task was to calculate the voltage across a heterojunction diode. Due to a different fraction of gallium, x, on each side of the diode, several slight modifications had to be made and a couple of large ones, as well. The small modifications were as follows:

- E_g needed to be calculated for each side
- ε_r needed to be calculated for each side
- N_c was now only relevant on the n side, and only had to use x_n
- N_V was now only relevant on the p side, and only had to use x_p

Calculating V_{bi}

These were all simple fixes, but V_{bi} required an entirely new calculation. Equation (3) is only relevant for homostructures. In order to calculate V_{bi} for a heterojunction, the actual Fermi

integral must be solved in order to find the difference in energy between the conduction band on each side of the junction. This calculation is shown schematically in Figure 10, where it can also be seen that the built in voltage can be calculated with three other parameters, one of which has already been calculated.

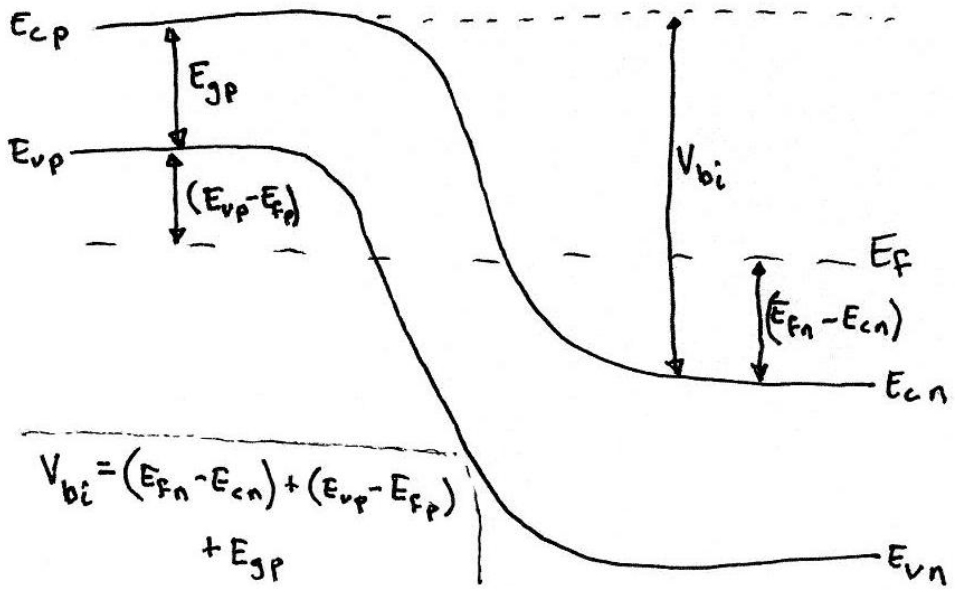


Figure 10: Band diagram of heterojunction showing calculation of Vbi.

These other values can be given by

$$\left(E_{fn}-E_{cn}\right)=\eta_n kT \quad \text{and} \quad (E_{vp}-E_{fp})=\eta_p kT \qquad \text{(12 and 13)}$$

where $\eta_{n,p}$ are parameters of the Fermi integral, whose solutions are given by the following

$$n/N_c = fermi(\eta_n) \quad \text{and} \quad p/N_v = fermi(\eta_v) \qquad \text{(14 and 15)}$$

In the above equations, "fermi" is the Fermi integral. Rather than spending the effort to numerically model this integral myself, I was able to use a MATLAB m-file for the evaluation of half-order Fermi-Dirac integrals called fermi.m (Mohankumar, 2007).

In the above equations it is apparent that n/N_c and p/N_v are known values (where $n=N_D$ and $p=N_A$), but in order to solve for η_n and η_p, it was necessary to solve the reverse Fermi

integral. As this is not possible with current analytical math, I simply plugged in a wide range of values for η_n and η_p and used the one that resulted in a value in closest agreement with n/N_c and p/N_v. The code for this is given below:

```
minn=1e6;minp=1e6;
for eta from -10 to 60
    sol=fermi(eta);
    if abs(sol-ND/Nc) < minn
        etan=eta;
        minn=abs(sol-ND/Nc);
    end
    if abs(sol-NA/Nv) < minp
        etap=eta;
        minp=abs(sol-NA/Nv);
    end
end
```

Calculating the difference in sizes of the bandgaps, ΔE_c

Now that the built-in voltage was calculated and functioning well, the next challenge to tackle was the difference in sizes of the band gap from one side of the junction to the other. That is, as shown in Figure 11, when materials have different compositions, their bandgaps are different sizes. This size difference must be accounted for in the calculation of the electric field and, thus, the current through the diode.

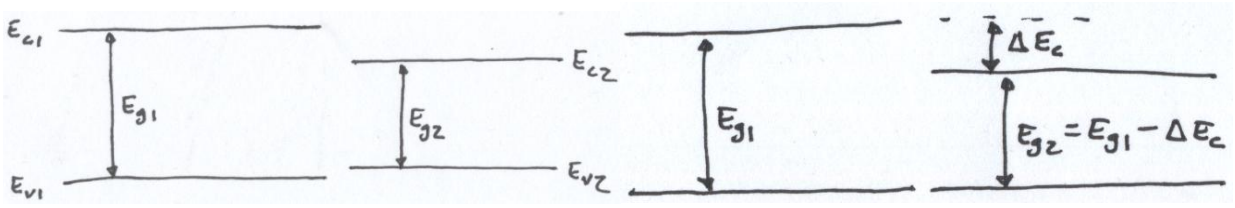


Figure 11: Difference in bandgap sizes and a simple metric to quantify them.

This difference can be modeled as simply the difference in location of the conduction band edge, as also demonstrated in Figure 11. By thinking of the valence-band edges as being aligned, one need only to calculate the ΔE_c between the two materials to account for the difference.

Calculating ΔE_c to zeroeth-order accuracy is a straight-forward calculation (Lundstrom, 1995):

$$\Delta E_c = \left|(4.9 - 0.83x_n) - (4.9 - 0.83 * x_p)\right| \quad (16)$$

However, after implementing this in the program, it was found to be too inaccurate. Unfortunately, calculating ΔE_c to a higher level of accuracy required more programming intensity than time allowed for in this project, and so it was decided that a better option was to make this unit user-definable, as well.

Calculating electric field using ΔE_c

As this difference in bandgap-sizes affects the electric field, it was necessary to find an equation for the electric field that took it into account. According to Bedair et al., 2000, the width of the depletion region is given as

$$W = \sqrt{\frac{2\varepsilon_{r,n}\varepsilon_{r,p}V_p(N_A+N_D)^2}{q(\varepsilon_{r,n}N_D+\varepsilon_{r,p}N_A)N_AN_D}} \quad (17)$$

by which it follows simply that the equation for the electric field is

$$E = \frac{V_p+\Delta E_c}{W} \quad (18)$$

This was then plugged into the original current for J_p, which was found to give reasonable results.

Calculating the series resistance

My final task in creating the InGaAs heterojunction diode reverse-bias current calculator was to accurately model the amount by which the current would be reduced given a user-input value for the contact resistance. As a review, this resistance occurs not because of the device itself, but because of the imperfect contact made to the device in order to test its

characteristics. This is shown schematically in Figure 12, where the on-state device is shown simply as a Zener diode. It is evident that, since this resistance is in series with the device, they will have identical currents flowing through them.

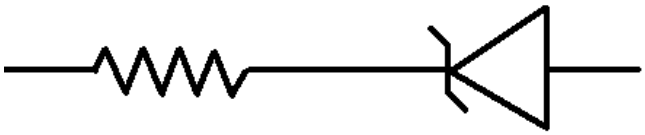


Figure 12: schematic representation of series resistance

All that needed to be done to calculate the voltage drop across this resistor was to multiply the current density by the area of the device, giving current, which then gives voltage by V=IxR. This can then be subtracted from V_p, the voltage across the diode. However, two current densities have been loaded into the program so far: the experimental current density and the ideal current density. Rather than use a complicated process to iteratively solve for ideal reduced voltage (and thus reduced current), we assumed that the ideal and the experimental ought to match well enough that the experimental could be used. Indeed, using the experimental current density values in the calculation of the ideal current with series resistance gave excellent agreement with the experimental results, which will be shown shortly.

Calculating a new E_g

In equation (5), the tunneling current density is calculated using the bandgap, E_g. However, there are now two bandgaps, one on each side of the junction. Rather than modifying the entire equation for J_p, it was found that simply using the average of the two bandgaps gave good experimental agreement.

Graphical User Interface

Though somewhat hidden throughout the text of the document, there are 12 different inputs to the calculator. Originally, this program was created to simply use text commands in MATLAB. These were entered in the form below:

```
ingaasJunction(1e20,5e19,.47,.47,300,0,1,'experimental/Kabeer.txt',.042,0,5,12000)
```

where the inputs are, in order, N_A, N_D, x_n, x_p,T, $V_{r,low}$, $V_{r,high}$, location of experimental results to use, m*, ΔE_c, series resistance in Ω, and area of the device in μm^2. This defeats the purpose of the program, since it is supposed to be easy to use. Thus, I decided to learn how to create a graphical user interface (GUI) in MATLAB. The result is shown in Figure 13.

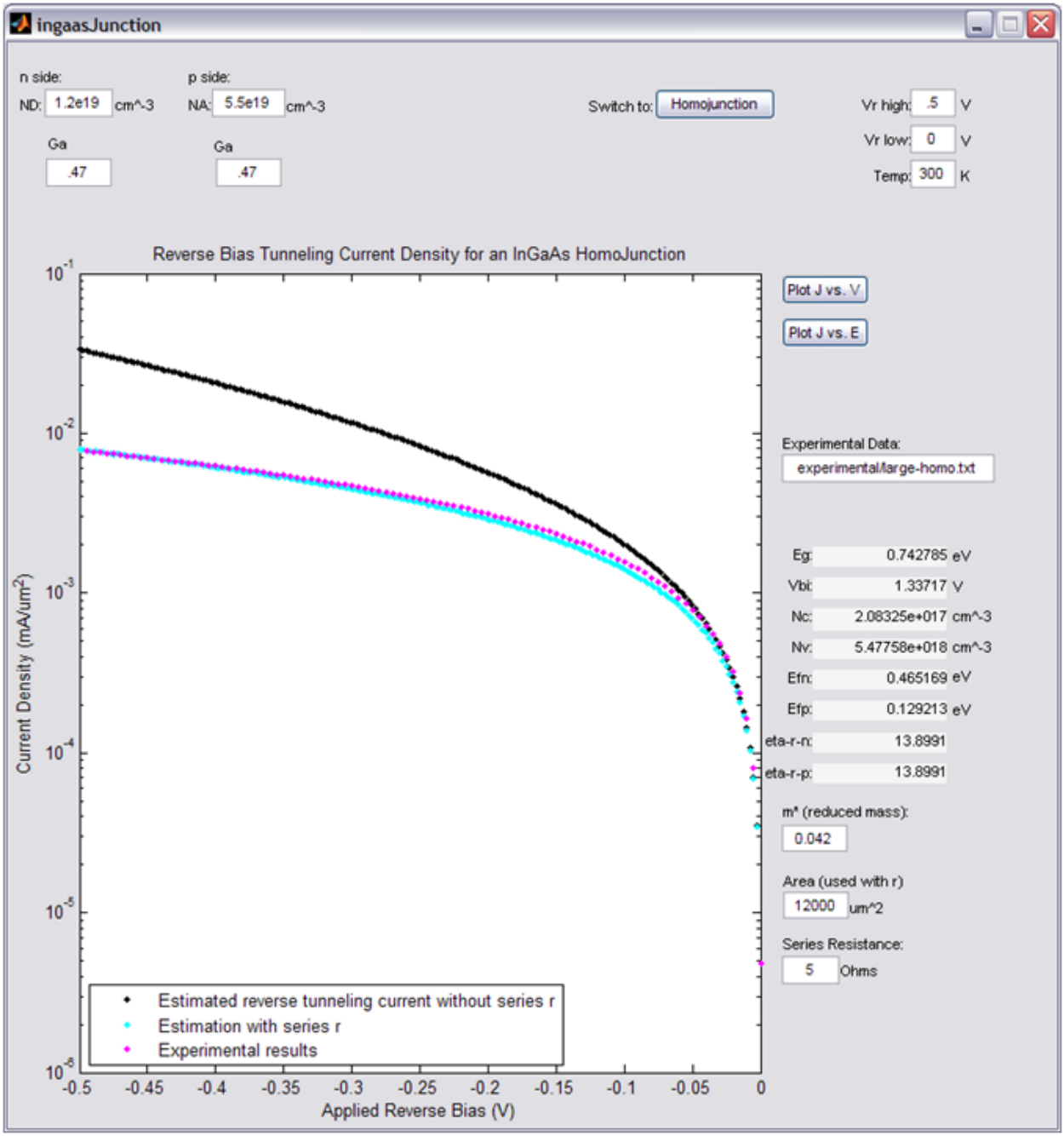


Figure 13: The GUI made in MATLAB

Conversion to PHP and AJAX

However, while one goal of the program was to have it be easy to use, the other was to have it be easy to access. Making it in MATLAB requires that those using it have MATLAB installed. While this may be the case for all researchers using the program, they will also then

need to download the calculator program itself and learn how to open it properly (it is not a straightforward task to open GUIs in MATLAB). Thus, I decided to learn how to implement this on the web, using PHP to do the calculations and using JavaScript and HTML to create the interface. This also added functionality that MATLAB did not have, such as the ability to mouse-over a point and be told its coordinates, as well as to zoom in on the graph. The interface for this is shown in Figure 14.

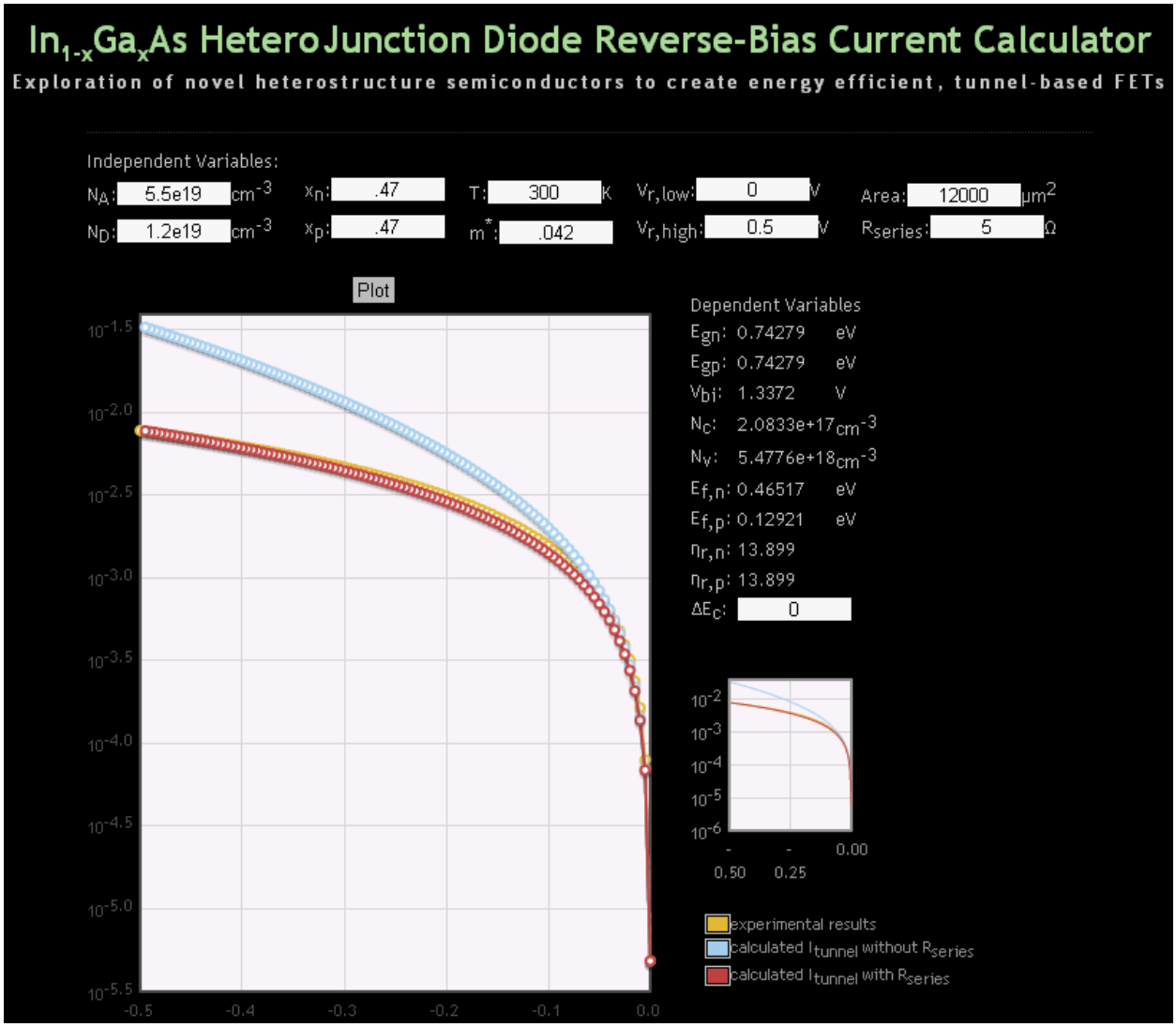


Figure 14: web-based calculator

This is the final product, which can be used online at www.chad-oh.com/thesis.

Conclusion

The web-based InGaAs heterojunction diode reverse-bias current calculator created for this project will aid in the research of these devices. While researchers create devices, they will have a quick, simple means of verifying their results, of seeing where they could improve, and of checking their series resistance. This will allow for a faster creation of fully-functional and manufacturable InGaAs tunneling FETs. Which will, in turn, allow for the continuation of Moore's Law for another several prosperous years.

Future Work

Future work in this specific area would be the creation of a more robust calculator. Several aspects were mentioned throughout the description that were too complicated to include for now:

- Equation (4) could be modified to more accurately account for the difference in bandgap on each side of the junction, rather than simply averaging the two.
- The series resistance could be obtained by recursively solving for the ideal reduced-current, rather than using the experimental current
- ΔE_c could be calculated more accurately by the program, rather than being user-specified
- A more sophisticated approximation could be used than that $n=N_D$ and $p=N_A$

More important than any of these, though, would be the ability to model an actual three-terminal device. As the Datta Research Group has produced functioning InGaAs FETs, and not simply recreated the diodes that Kabeer and Seabaugh demonstrated over a year ago, it would be very useful to model this important aspect of the devices.

In order to make these goals possible, I have attempted to comment all of my code well, and all of the source code is available on the web page where the program resides. Additionally, the original MATLAB code can also be downloaded.

Works Cited

Alfred, R. (2008, August 7). *Aug. 7, 1944: Still a Few Bugs in the System*. Retrieved November 14, 2009, from http://www.wired.com/science/discoveries/news/2008/08/dayintech_0807

Bedair, S. M., Roberts, J. C., Jung, D., Moody, B. F., El-Masry, N. A., & Katsuyama, T. (2001). Analysis of p+-AlGaAs/n+-InGaP tunneljunction for high solar concentration cascade solar cells. *Conference Record of the Twenty-Eighth IEEE Photovoltaic Specialists Conference-2000* (pp. 1154-1156). Piscawtaway, NJ: Institute Of Electrical & Electronics Engineers.

Bellis, M. (n.d.). *Intel 4004 - The World's First Single Chip Microprocessor*. Retrieved November 14, 2009, from http://inventors.about.com/od/mstartinventions/a/microprocessor.htm

CNET News.com. (2006, November 17). *Photo Gallery: 35 Years of Intel Chip Design*. Retrieved November 14, 2009, from http://www.zdnet.com.au/insight/hardware/soa/Photo-gallery-35-years-of-Intel-chip-design/0,139023759,339272257,00.htm

Datta, S. (n.d.). Various personal conversations throughout 2009.

Datta, S., Ho, T., Mayer, T., Mookerjea, S., Narayanan, V., Schlom, D. & Vallet, A. (2008)*Vertical Tunnel Transistors: Simulation, Design and Fabrication*. Midwest Institute of Nanoelectronics Discovery.

Först, C., Ashman, C., Schwarz, K., & Blöchl, P. (n.d.). The interface between silicon and a high-k oxide.*TU Wien*. Retrieved November 14, 2009, from http://info.tuwien.ac.at/theochem/si-srtio3_interface/si-srtio3.html

Intel. (n.d.). *Moore's Law: Made Real by Intel Innovation*. Retrieved November 14, 2009, from http://www.intel.com/technology/mooreslaw/

Kabeer, S., & Seabaugh, A. (2008, September 26) *InGaAs Tunnel Junctions.* Midwest Institute of Nanoelectronics Discovery. University of Notre Dame.

Kamkwamba, W. (Narrator). *Moving Windmills: The William Kamkwamba Story* [Online video]. YouTube. Retrieved November 14, 2009, from http://www.youtube.com/watch?v=arD374MFk4w

Lundstrom, M. (Director) (1995 Fall). Notes on Heterostructure Fundamentals.*EE 650Y*. Lecture conducted from School of Electrical and Computer Engineering and the NSF MRSEC for Technology-Enabling Heterostructures, West Lafayette, Indiana 47907.

Massey, D. (n.d.). *History of the Transistor (the "Crystal Triode")*. Retrieved November 14, 2009, from http://www.porticus.org/bell/belllabs_transistor.html

Mohankumar, Nandagopal. (2007, January 10). *fermi.m: Matlab m file to evaluate the Half-order Fermi-Dirac integral for Semiconductor/Solid State Applications*. From *MATLAB Central: An open exchange for the MATLAB and Simulink user community* <http://www.mathworks.se/matlabcentral/fileexchange/13616-fermi?controller=file_infos>. Retrieved 10 June 2009.

NSM Archive – Gallium Indium Arsenide (GaInAs) – Band structure. (n.d.). *Electronic archive: New Semiconductor Materials. Characteristics and Properties*. Retrieved November 15, 2009, from http://www.ioffe.rssi.ru/SVA/NSM/Semicond/GaInAs/bandstr.html

Szczys, M. (2009, October 8). *William Kamkwamba talks windmills with Jon Stewart*. Retrieved November 14, 2009, from http://hackaday.com/2009/10/08/william-kamkwamba-talks-windmills-with-jon-stewart/

TechTree News Staff. (2009, September 23). *IDF09: Intel Shows Working 22nm Chips*. Retrieved November 14, 2009, from http://www.techtree.com/India/News/IDF09_Intel_Shows_Off_Working_22nm_Chips/551-106501-581.html

Van Meter, R. (2009, April 1). *OS History and Basic Concepts*. Retrieved November 14, 2009, from http://web.sfc.keio.ac.jp/~rdv/keio/sfc/teaching/system-software/system-software-2009-prod/lec01.html

Academic Vita

Education

The Pennsylvania State University — University Park, PA
B.S. in **Engineering Science**, Engineering Honors Curriculum — Graduation: Dec. 2009
B.S. in **Electrical Engineering** — Cumulative GPA: **3.64**/4.0
Schreyer Honors College
Thesis: Exploration of Novel Heterostructure Semiconductors to Create Energy Efficient TFETs

Experience

Penn State Physics Department — University Park, PA
Staff Operations Systems Consultant — March 2009 – Present

- Sorted and interviewed applicants
- Collaborated with manager and developers to design efficient means of managing data

Penn State Engineering Science Department — State College, PA
Teaching Intern — September 2009 – Present

- Prepared lesson plans and taught two classes
- Discussed educational philosophy and the content of the course each week with the instructor

ITS Lab Consulting — University Park, PA
Lab Consultant — January 2006 – July 2009

- Guided users through troubleshooting computer problems
- Contributed to an online open-source (wiki) knowledge center for use by other consultants

Bridge Semiconductor — State College, PA
Intern — September 2008 – January 2009

- Maintained online database for analysis of wafers and success of fabrication techniques
- Aligned poles of PZT layer in wafers to move fabrication process towards completion

Pitney Bowes — Shelton, CT
Electrical Engineering Intern — May 2008 – August 2008

- Designed and built reaction timer with coworker
- Constructed prototype board to creatively meet needs of customer

Teamwork

- Collaborated regularly with Ph.D. students towards completion of undergraduate thesis
- Spearheaded artificial neural network project to analyze music genres with classmate
- Oversaw team development of hypersonic directional speakers
- Created organizational database with a developer for a campus literary magazine
- Organized large service project towards the completion of Eagle Scout requirements