

Evaluation of MOSFETs for Terahertz Detector Arrays

by

Gregory J. Fertig

B.S. Clemson University, 2008

A thesis submitted in partial fulfillment of the
requirements for the Degree of Master of Science in

Imaging Science

Chester F. Carlson Center for Imaging Science

College of Science

Rochester Institute of Technology

15 August 2014

Signature of the Author _____

Accepted by _____
Dr. John Kerekes, M.S. Program Coordinator

CHESTER F. CARLSON CENTER FOR IMAGING SCIENCE
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

M.S. DEGREE THESIS

The M.S. Degree Thesis of Gregory J. Fertig
has been examined and approved by the
thesis committee as satisfactory for the
thesis required for the
M.S. Degree in Imaging Science

Dr. Zoran Ninkov, Co-Advisor

Dr. Emmett J. Ientilucci, Co-Advisor

Dr. Christiaan Richter, Committee Member

Dr. Kenny Fourspring, Committee Member

Date

DISCLAIMER

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

Evaluation of MOSFETs for Terahertz Detector Arrays

by

Gregory J. Fertig

Abstract

The terahertz (THz) region of the electromagnetic spectrum is one of the last remaining regions that has yet to be fully characterized. THz imaging is one of the foremost drivers of this technology gap and has the potential to push development in the near term to a similar capability level as infrared (IR). Properties of THz radiation are introduced, along with promising current applications. Interest in array based imaging of THz radiation (T-Rays) has gained traction lately, specifically using a CMOS process due to its ease of manufacturability and the use of MOSFETs as a detection mechanism. The theory outlined explains that incident terahertz radiation on to the gate channel region of a properly configured MOSFET can be related to plasmonic response waves, which change the electron density and potential across the channel producing a photoinduced response. This work utilizes a test chip fabricated to investigate these effects. The 0.35 μm silicon CMOS MOSFETs tested contain varying structures, providing a range of detectors to analyze. Included are individual test MOSFETs for which various operating parameters and modes are studied and results presented. The focus on single transistor-antenna testing provides a path for discovering the most efficient combination for coupling 0.2 THz band energy. Specifically introduced, is a novel source region extension which is proven to improve MOSFET response. Sensitivity analysis and responsivity are described, in parallel with theoretical expectations of the plasmonic response in room temperature conditions. A maximum responsivity of 40 kV/W and corresponding NEP of $10 \text{ pW } \sqrt{\text{Hz}}$ ($\pm 10\%$ uncertainty) is demonstrated.

Acknowledgements

A science-based graduate degree can be a whirlwind. With all of the ups and downs there are many opportunities to get lost, confused, and stressed, opposed by the excitement, enthusiasm, and wonder of the next big thing. For me the community at the Carlson Center provided a wonderful, supporting environment, and steered me back on track when I might stray to far from the path. This is all part of the journey however, and in particular I'd like to thank Professor Zoran Ninkov, who first drew my attention in the classroom. I found myself having fun and working hard, and knew that this was what research is all about. He has a way of inspiring new ideas, while also being very understanding, letting you fail just enough, and most importantly providing you ownership of the research. I would also like to thank Professor Emmett Ientilucci, for guiding my quest for the right research project, and encouraging me the whole way. Professor Dave Messinger, for putting up with my many brainstorming sessions in an effort to grasp remote sensing. Thanks to Professor Christiaan Richter for patiently explaining terahertz concepts to me in his lab, and being a critical in-house resource of terahertz expertise. When it came time to start the nitty-gritty, long days in the terahertz lab, Dr. Kenny Fourspring was there to bring me up to speed, and help troubleshoot test setups. He also has an uncanny ability to remember important concepts and details when I would forget about them in the weeds. They have all taught me what a practical scientist should know, and I am very satisfied with this work after putting it all together.

Words cannot express my gratitude towards my family and parents, who sacrificed so much to put me in a position to succeed. My father instilled me with an engineering mind at a young age, fixing and building projects, and fostering my inquisitive mind. My mother is a constant voice of reason and experience in my life. She is always setting me up with the right words and advice to make my own decisions, never injecting too much opinion even when I make idiotic ones. Thanks to my brother for showing me what true heart really means, and my best friend Phil for always being there no matter how much distance or time is between us.

Finally, I want to thank my Clemson family and friends for all of their support and motivation. They provided me the means to attend undergraduate EE, allowed me to serve and defend this country, and thrust me on the path towards graduate school.

“Most people say that it is the intellect which makes a great scientist. They are wrong: it is character.” - Albert Einstein

Contents

List of Abbreviations	x
List of Terms	xi
List of Tables	xiv
List of Figures	xv
1. Introduction	1
1.1. Background	1
1.2. Motivation	3
1.3. Terahertz Radiation Properties	3
1.3.1. Non-Ionizing	4
1.3.2. Penetration	4
1.3.3. Resolution	5
1.3.4. Scattering	5
1.3.5. Safety	6
1.3.6. Atmospheric Transmission	7
1.4. Applications	9
1.4.1. Astronomy	9
1.4.2. Spectroscopy & Material Identification	10
1.4.3. Security & Defense	11
1.4.4. Communications	15
1.4.5. Medicine	17
1.4.6. Manufacturing	18
1.5. Summary	18
2. Theory	19
2.1. MOSFET Basics	19
2.1.1. Gradual Channel Approximation	21
2.1.2. Conductance & Transconductance	23

2.2. Plasmonic Detection Mechanism	27
2.2.1. Resonant Detection	29
2.2.2. Non-Resonant Detection	30
2.2.3. Theoretical Calculation	31
2.2.4. Application and Enhancement	32
3. Fabricated Test Chip Design & Evaluation	35
3.1. Research Collaborators	35
3.2. Chip & Pixel Description	35
3.3. Enclosure and Fanout Board Configuration	41
3.4. Experiment Description	44
3.5. Terahertz Signal Data Acquisition	46
4. Results & Discussion	51
4.1. MOSFET Characterization	51
4.1.1. Transconductance	51
4.1.2. Channel Conductance & Resistance	53
4.1.3. Thermal Noise	54
4.1.4. Random Telegraph Signal Noise	61
4.2. Gunn Diode Source Characterization	63
4.2.1. Power Measurements	63
4.3. MOSFET Response	73
4.3.1. Gas Laser Results	73
4.3.2. Gunn Diode Results	74
4.3.3. Response & Orientation	80
4.3.4. Responsivity & NEP	81
5. Conclusion	88
5.1. Future Work	89
A. Test Equipment Description	91
A.1. Keithley 2602A Source Measurement Unit	91
A.1.1. Noise Considerations	92
B. Wavelength Frequency Conversion Table	96
C. MATLAB Test Scripts	100
C.1. Programming the Keithley 2602	100
C.2. TSP Transconductance Code	101
C.3. Serial Port Enable Code	106

Contents

C.4. Transconductance Script	108
C.5. Drain Sweep Scripts	114
C.6. Terahertz Detection Scripts	126
Bibliography	163

List of Abbreviations

AC	Alternating Current
CDS	Correlated Double Sampling
CEIS	Center for Emerging and Innovative Sciences
CMOS	Complementary Metal Oxide Semiconductor
CT	Computerized Tomography
dBm	Decibel milli-Watts (sometimes dBmW)
DC	Direct Current
DIP	Dual Inline Package
FPA	Focal Plane Array
GCA	Gradual Channel Approximation
GDS	Graphic Data System File
HEMT	High Electron Mobility Transistor
HFET	Heterojunction Field Effect Transistor
IR	Infrared
LSD	Linear Spectral Density (Noise)
MODFET	Modulation Doped Field Effect Transistor
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
MRI	Magnetic Resonance Imaging
NPLC	Number of Power Line Cycles
NSR	Noise to Signal Ratio
NYSTAR	New York State Office of Science Technology & Academic Research
RIT	Rochester Institute of Technology
RMS	Root Mean Squared
RTS	Random Telegraph Signal (Noise)
SMU	Source Measurement Unit
SNR	Signal to Noise Ratio
THz	Terahertz
UPS	Uninterrupted Power Supply
UR	University of Rochester

List of Terms

Constants

Symbol	Value	Units	Description
ϵ_0	$8.85419 * 10^{-12}$	$F m^{-1}$	Permittivity of Vacuum
ϵ_{ox}	3.90	Unitless	Permittivity of Silicon Dioxide (Silica)
ϵ_{si}	11.70	Unitless	Permittivity of Silicon
k_B	$1.38 * 10^{-23}$	JK^{-1}	Boltzmann Constant
μ_n	250 – 1400	$cm^2 V^{-1} s^{-1}$	Electron Mobility (Typical Range)
m_e^*	0.19	Unitless	Electron Effective Mass in Silicon
m_0	$9.11 * 10^{-31}$	kg	Electron Rest Mass
n_i	$1.5 * 10^{10}$	cm^{-3}	Intrinsic Carrier Concentration of Silicon @ 300K
n_c	10^{16}	cm^{-3}	Typical Electron Concentration in MOSFET Channel for Silicon @ 300K
q	$1.62 * 10^{-19}$	C	Elementary Charge
t_{ox}	7.9	nm	Oxide Thickness

Variables

Symbol	Units	Description
α	Unitless	Pyroelectric Absorption Factor
A_d	cm^2	Area of Pyroelectric Detector
A_{pix}	m^2	Area of a Pixel
$A_{r,eff}$	cm^2	Effective Area of Receive Antenna
C	$F cm^{-2}$	Gate Channel Capacitance per Area
C_{ox}	$F cm^{-2}$	Gate Oxide Capacitance per Area
D	m	Diameter of Circular Aperture
d_r	mm	Diameter Receive Antenna
d_s	mm	Diameter of Source Aperture
d_t	mm	Diameter Transmit Antenna
η	Unitless	Ideality Factor (MOSFET Plasmon Response)
e_A	Unitless	Aperture efficiency

List of Terms

Δf	Hz	Measurement Bandwidth
f	Hz	Frequency
g_o, g_d	S	Conductance of MOSFET Channel
g_m, g_{fs}	$A V^{-1}$	Transconductance
i_D	A	Drain Current
j_0	$A cm^{-2}$	Gate Leakage Current Density
κ	Unitless	Dimensionless Gate Leakage Parameter
λ	m	Wavelength
L	μm	Length of the Channel
L_s	μm	Length of the Source Extension
NEP	$pW \sqrt{Hz}^{-1}$	Noise Equivalent Power
ω_0, ω_n	Hz	Fundamental and Eigen Plasmon Frequencies
ω_{thz}	Hz	Incident Terahertz Frequency
$\omega_0 \tau_r$	Unitless	Resonance Quality Factor
P_d	mW	Total Power on the Detector
$P_{density}$	$W cm^{-2}$	Power per Unit Area of the Source
P_r	mW	Power at Receive Side
P_s	mW	Total Power at the Source
P_t	mW	Power at Transmit Side
R	Ω	Resistance
R_d	$V W^{-1}$	Pyroelectric Responsivity
R_v	$V W^{-1}$	MOSFET Responsivity
r	cm	Distance from Source
r_0	Ω	Resistance of MOSFET Channel
σ_c	S	Conductivity of MOSFET Channel
$S_{detector}$	$V W^{-1}$	GaAs Detector Sensitivity
s_p	$m s^{-1}$	Plasmon Wave Velocity
Θ	rad	Measure of Angle
τ_r	s	Electron Momentum Relaxation Time
T	K	Temperature
$\Delta U, \delta U$	V	Plasmonic Photoinduced Voltage at the Drain (DC)
U_0	V	Gate Voltage Swing ($V_{GS} - V_{TH}$)
U_a	V	Plasmonic Photoinduced Voltage at the Gate (AC)
V_{BS}	V	MOSFET Base-Source Voltage
V_{DS}	V	MOSFET Drain-Source Voltage
V_{GD}	V	MOSFET Gate-Drain Voltage
V_{GS}	V	MOSFET Gate-Source Voltage
V_{Gx}, U	V	MOSFET Gate-Channel Voltage

List of Terms

$V_{measured}$	V	Measured Voltage
V_{pix}	V	Measured Output Voltage from a Pixel
V_{TH}	V	MOSFET Threshold Voltage
W	μm	Width of the Channel

List of Tables

3.1. SMU biasing schemes. Source/measure denote whether the SMU is providing signal or measuring signal. Voltages are either fixed or swept from 0 - 3.3 V. A measurement occurs for each increment of a given sweep. For each measurement, the SMU can perform a specified number (typically 100) of sub-measurements at an integration rate (NPLC) for a given static bias condition. Test 3 and 4 produced the strongest detection signal and are described in the previous theory section.	45
--	----

List of Figures

1.1.	Electromagnetic Spectrum. Terahertz region described as 0.1 to 10 THz (3 mm to 30 μm). Adapted from [1].	1
1.2.	Diagram and equations describing Rayleigh scattering. Essentially, light of different wavelengths (colors) scatters at different angles when coming into contact with air molecules. This is why the sky looks blue and sunsets look red/orange. Reprinted from [2].	6
1.3.	Atmospheric attenuation of the THz spectrum. The boundaries of the region of interest are annotated. Adapted from [3].	7
1.4.	The “THz wall”. Calculated for a communications link using fixed gain antennas an a horizontal line-of-sight path on the ground. Good and bad weather transmitted power are shown for frequencies between 35 GHz and 3 THz. Transmitted power is shown in decibel-milliwatts, or dBm, which denotes the amount of power required at the sending side to reach the receive end at the specified distance. Adapted from [4]. Inset: Power conversion chart from decibel-milliwatts to watts.	8
1.5.	Spectral content of a typical star forming cloud in the galaxy. A Jansky is a measure of spectral flux density, equivalent to: $1 \text{ Jy} = 10^{-26} \text{ W m}^{-2} \text{ Hz}^{-1}$. Adapted from [5].	10
1.6.	Transmittance spectrum for table sugar in the THz. Reprinted from [6]. .	11
1.7.	Example ThruVision system concepts for security screening. ThruVision systems are active operating in the 0.25 THz wavelength region. Reprinted from [7]	12
1.8.	Example image from ThruVision system. The inset on the left shows an overlaid THz image and the presence of a handgun under the subject’s clothing. Reprinted from [7]	13
1.9.	Example image of T-COGNITION system. Software analysis shows the presence or absence of identified substances on a visible image. Reprinted from [8].	14
1.10.	Safe VISITOR system operates at 0.35 THz and capable of slow frame-rate video. Adapted from [9]. (a)Example camera setup. (b)THz image of resolution chart at 8 m. (c) THz image of person with concealed gun at 8 m.	15

1.11. Current communications system layout for human spaceflight missions. S (2-4 GHz), Ku (12-18 GHz), and Ka (26-40 GHz) band systems are the primary operating bands. Adapted from [10].	16
1.12. Visible and THz image of human tooth using a TeraView system. The absence of visual features on the tooth surface makes early detection of tooth decay difficult. X-rays, one of the accepted methods used to detect decay, only reveals the problem at a relatively late stage, when drilling and filling is the only method available to halt the decay. If decay can be detected early enough it is possible to reverse the process without the need for drilling by the use of either fissure sealing or remineralization. Reprinted from [11].	17
2.1. MOSFET diagram. The semiconductor substrate in an n-channel device is traditionally p-type silicon with the source and drain implants doped n+ silicon. The insulator layer below the gate is silicon dioxide, and the contacts are made of a conductive material like polysilicon or silicide. Reprinted from [12].	20
2.2. The channel region of a MOSFET under a pinch off state. The asymmetry is formed by the presence of a drain bias voltage. Too much drain bias will force the channel into a pinch off state, and the inversion layer is no longer maintained. Adapted from [13].	21
2.3. MOSFET exhibiting GCA conditions. The charge varies gradually along the channel of length, L, with a strong inversion at the source and a weak inversion at the drain. This asymmetry is due to the drain bias voltage, V_{DS} . Reprinted from [14].	22
2.4. The MOSFET Family of Curves. This plot of drain current vs drain voltage describes the MOSFETs operating characteristics. The gate voltage for each curve is annotated. The dashed line represents the saturation voltage, where $V_{GS} - V_{TH} \leq V_{DS}$. Example data is from one of the THz test MOSFETs to be described in Section 4.	23
2.5. The Conductance Curve. Represents the change in i_D vs the change in V_{DS} . V_{GS} is fixed at 1 V. V_{TH} for this device is ≈ 0.5 V. Example data from test MOSFETs.	24
2.6. The Transconductance Curve. Represents the change in i_D vs the change in V_{GS} . V_{DS} is fixed at 0.1 V. Same test MOSFET as Figure 2.5.	26

2.7. MOSFET diagram of signal detection. Under proper biasing, incoming THz radiation produces an AC signal between the gate and source region. This is rectified and detected as a DC voltage change between the source and the drain. Channel length, L , affects the response frequency and determines whether a resonant or non-resonant detection occurs. Adapted from [15].	27
2.8. Theoretical operation of MOSFET for THz detection. U_a is the photoinduced AC voltage, U_0 is the DC gate voltage bias from threshold, and dU is the rectified DC photoresponse voltage.	33
2.9. Experimental application of MOSFET for THz detection. The antenna intends to enhance response, with the bias applied through V_{GS} and i_D . Response is measured through V_{DS}	33
3.1. Responsivity of THz MOSFET vs. frequency. The triangles show the measured points for a MOSFET with a gate length of $L = 130$ nm and coupled with a bowtie style antenna. Notice the peak response near 300 GHz. For a larger MOSFET such as the 350 nm under test, the response will be lower [16]. The inset shows a diagram of the MOSFET and antenna for reference. Adapted from [17].	36
3.2. Test Chip 1 (a) 40-pin DIP package for test chip. Individual pins allow selection of test transistors without clocking electronics. (b) Micrograph of the test chip. The five test transistors on the bottom edge are being characterized in this work.	37
3.3. GDS file snapshot of T-1 pixel design. Bowtie antenna has dimensions of $70\ \mu\text{m} \times 6\ \mu\text{m}$ with a $2\ \mu\text{m}$ gap.	38
3.4. GDS snapshot of a test transistor with bowtie antenna. (a) Full pixel and antenna $5.7\ \mu\text{m} \times 68.4\ \mu\text{m}$, $2\ \mu\text{m}$ Gap). (b) Blowup view of the design with the guard rail around the edge. The orange (lower) shaded area represents the metal shield layer, with the antenna and detection MOSFET located in the top half of the image.	39
3.5. GDS snapshot of test transistor with spiral antenna. (a) Full pixel and antenna with a $72\ \mu\text{m}$ diameter. (b) Blowup view of the pixel. These pixels are identical to the bowtie antenna pixels, the only difference being the antenna.	40
3.6. MOSFET with extended source region. L_s represents the length of extension of the source from the gate.	40

3.7.	The five bowtie test MOSFETs. The only difference in the five test transistors (T-1:5) is the source region extension which varies from 0 μm - 2 μm . The T-1 transistor has the shortest extension while the T-5 transistor has the longest extension. This notation corresponds to the discussion in the results Section 4.	41
3.8.	The test chip is accessed through a fanout board. (a) Front view of the fanout board including the chip mounted in socket. (b) 9 V battery connections, feed through capacitors, and a 3.3 V regulator to provide power for the chip. Gate, Drain, and Source leads are fed directly to feedthrough capacitors on the enclosure.	42
3.9.	Enclosure Front View. A removable silicon windows and front cover provide easy access to the fanout board and chip inside. A conductive seal creates a Faraday cage around the detector.	42
3.10.	Enclosure Back View. The rotation stage is centered at the detector center-axis for rotational purposes. The switch controls 9 V power to the board.	43
3.11.	Enclosure Internal View. Standoffs and nylon washers mount the board approximately 1" from the back of the enclosure.	44
3.12.	Experimental test setup for detector characterization and beam profiling. The enclosure is translated via XYZ and rotation stages. The SMU is commanded via MATLAB serial interface for bias conditions, shutter position, and data transfer.	45
3.13.	T-5 bowtie example data. Raw voltage measurements. V_{DS} Measurements vs. Time	47
3.14.	T-5 bowtie example data. Shutter closed: V_{DS} and i_D	48
3.15.	T-5 bowtie example data. Shutter Open: V_{DS} and i_D	48
3.16.	T-5 bowtie example data. The signal ΔU vs V_{GS} (100 measurements per step).	49
3.17.	T-5 bowtie example data. The average signal ΔU and noise as σ	50
4.1.	T-1 Transconductance Curve.	52
4.2.	T-5 Transconductance Curve.	52
4.3.	T-5 Channel Resistance. The resistance of the channel as a function of drain voltage. Each series represents a different gate voltage bias.	53
4.4.	T-5 Channel Resistance. The median resistance as a function of gate voltage.	54
4.5.	T-5 Conductance, g_0 is calculated from the slope of the I-V curve from Figure 2.4.	56
4.6.	T-5 Resistance, r_0 vs V_{DS} . Point of interest is noted.	56
4.7.	Image of resistor wired on source-drain terminals inside enclosure.	57

List of Figures

4.8. MOSFET measurements for thermal noise. Current vs time.	58
4.9. 225 k Ω resistor measurements for thermal noise. Current vs time.	59
4.10. Power spectrum of MOSFET signal from Figure 4.8.	60
4.11. Signal from Figure 4.8 w/o 60 Hz. Measurements agree with with calculated 17 pA Johnson noise.	60
4.12. RTS Noise Example. (a) and (c) show the voltage time signal and (b) and (d) show the linear spectral density for a normal vs a bad RTS pixel. Notice in (a) the random DC offset shift around two different voltage means. This is an example of RTS noise, whereas in (c), these shifts are not present. An increase of ≈ 4 mV RMS of noise is added to the signal due to RTS.	62
4.13. VDI 188 GHz Gunn diode.	63
4.14. Gunn diode power measurements using GaAs diode detector. The $1/R^2$ fit line follows the measurements precisely. The minor oscillations in the measured voltage are due to phasing reflections between the transmit and receive antennas.	64
4.15. Response Curves for 20 GHz Detector. The 200 GHz detector is approximately 8 times less sensitive than the 20 GHz detector the chart is spec'd for. This factor of 8 was provided by Pacific MM support.	67
4.16. Power calculated at transmit side (source) for each measurement along r . The average is ≈ 55 mW total integrated power. Sampling increments are 1 mm. The noise around the average is due to variations in the measured voltage output from the GaAs diode.	68
4.17. Gunn diode power measurements using the pyroelectric detector. The data and $1/R^2$ fit line are shown.	69
4.18. Radiation pattern of the Gunn diode source antenna. Provided by the manufacturer Custom Microwave, Inc. [18].	71
4.19. Detection signal as a function of gate length at 0.7 THz Reprinted from [19]	73
4.20. Experiment Setup in the UR THz Laser Lab. A mechanical chopper wheel and lock-in measurement technique are used to boost SNR. A HeNe alignment laser is used to confirm algorithm operation since the silicon chip is sensitive to visible light.	74
4.21. Bowtie T-1 Photoresponse & NSR.	75
4.22. Bowtie T-2 Photoresponse & NSR.	76
4.23. Bowtie T-3 Photoresponse & NSR.	76
4.24. Bowtie T-4 Photoresponse & NSR.	77
4.25. Bowtie T-5 Photoresponse & NSR.	77

List of Figures

4.26. Photoresponse ΔU of all five MOSFETs at peak drain bias. Errorbars represent the standard deviation of signal measurement.	78
4.27. Blowup of Figure 4.26 near peak response with NSR.	79
4.28. Diagram of polarized radiation. (Top) In the case where the electric field of the antenna is aligned with the polarization of the source, the radiation will be coupled. (Bottom) If the electric field of the antenna is perpendicular to the polarization of the source, than little to no radiation is coupled. . .	80
4.29. Bowtie T-1 Response. Horizontal vs Vertical Orientation. The polarization of the source is co-aligned with the horizontal case. The bowtie antenna and the MOSFET gate (perpendicular to channel current flow) are in the same orientation.	81
4.30. Quasi-optical system interpretation. The focal region describes the plane at which incident radiation interacts with the antenna. Reprinted from [20].	82
4.31. Example deconvolution result for NbN microbolometer pixel. (a) The intensity distribution of the incident beam, $I(x, y)$. (b) The 2-D measured scan $S(x, y)$ using the NbN pixel. (c) Deconvoluted spatial response of the pixel $B(x, y)$. The x-y projection of this response is the effective aperture (area) of the pixel which is 0.7 mm^2 , 7.4 times larger than the physical size of the detector (not including the dipole planar antenna). Reprinted from [20].	84
4.32. Pixel Area Definition. For comparison with other's results, a $100 \mu\text{m} \times 100 \mu\text{m}$ area is defined.	85
4.33. The responsivity as a function of detection voltage. The markers represent the range of detection voltages from 50 mV (T-1) to 180 mV (T-5).	86
4.34. NEP as a function of the responsivity. The markers represent the range of responsivity of the test MOSFETs from $\text{pW } \sqrt[1]{\text{Hz}}$ (T-1) to $\text{pW } \sqrt[1]{\text{Hz}}$ (T-5).	87
5.1. Next-generation test chip design. Includes a 7×7 pixel array based on the T-5 test MOSFET which includes on-chip CDS, current biasing, and gain. 15 new test MOSFETs are also included to help parametrize future pixel design.	90
A.1. Re-wiring of fanout board.	93
A.2. THz test enclosure and connections.	93
A.3. Noise comparison before and after rewire.	94

1. Introduction

1.1. Background

The terahertz (THz) band considered for this thesis is the 0.1 to 10 THz (3 mm to 30 μm) as seen in Figure 1.1.

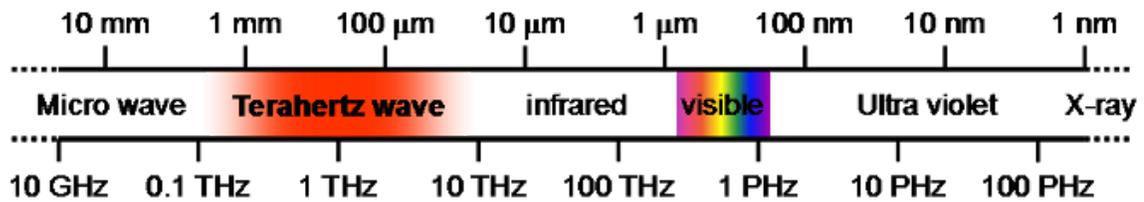


Figure 1.1.: Electromagnetic Spectrum. Terahertz region described as 0.1 to 10 THz (3 mm to 30 μm). Adapted from [1].

Use of the word terahertz is fairly recent, but the most widely accepted research in which it is first documented is the 1974 paper by J. W. Fleming[21]. There are various other terms for this region of the electromagnetic spectrum. On the low frequency end, millimeter wave, sub-millimeter band (referring to the wavelength), gigahertz, and far infrared are a few common terms. This is due to historical spectral windows used by scientists, particularly astronomers[22]. This portion of the electromagnetic spectrum is one of the last remaining regions that has not been fully characterized[15]. Several reasons for this include its poor atmospheric transmission (Figure 1.3), location between optical and microwave regimes, and lack of materials for generation and transmission of

the radiation. Because of this, technology advancement in this area is much less developed than typical with regards to the source power and sensitivity of detectors, hence the term “terahertz gap.” Interestingly, the energy in this band has penetrative properties through non-metallic materials, making it a viable application for manufacturing, security, medical imaging, communications and materials/chemicals characterization.

The THz region lies between two common technological domains, the optical regime (photonics) where experts’ intuition is expressed in terms of wavelength into the far-infrared, and the microwave (radio-electronics) regime where it is expressed in terms of frequency down to gigahertz (GHz). Because of its location in the spectrum, theory and practices from both schools of thought can be employed to utilize THz radiation, but this also comes with unique limitations. Currently, there is no ideal method to generate or detect THz waves. Each application picks different source and detection technologies. Imaging research included, has yet to find an efficient method for detection.

THz imaging has been a foremost driver of THz and could push technology development in the near-term to similar capabilities as infrared (IR). Each detection method however has its disadvantages; size, speed, sensitivity, noise, etc. Due to low sensitivity and signal levels, most systems employ a raster scanning techniques to form an image of adequate size. One of the most promising methods to mitigate this and move towards framing techniques is using silicon CMOS (Complimentary Metal Oxide Semiconductor) MOSFETs (Metal Oxide Semiconductor Field Effect Transistor). When biased properly, a MOSFET channel acts as a THz detector. Incident THz radiation disturbs the electrons in the channel creating a detectable plasma wave. Often these detectors are paired with an antenna to increase radiation coupling efficiency. Now a pixel is realized and can be utilized in the same way as digital camera framing arrays.

1.2. Motivation

This thesis outlines the background of THz radiation, theory of detection, current technologies, and how a silicon CMOS based pixel is designed and tested. The design and test of an example pixel and chip was completed as part of a Center for Emerging and Innovative Sciences (CEIS) collaboration between the University of Rochester (UR), the Rochester Institute of Technology (RIT), and Exelis Geospatial Systems. CEIS is a New York State (NYSTAR) funded advanced technology center, designed to bring together companies and university researchers who have common areas of interest[23]. Corporate sponsored research and development funding is matched by the state in order to promote collaboration.

The purpose of this THz research group is to develop a prototype focal plane array (FPA) for use in imaging systems and standoff threat detection applications[24]. CMOS technology was chosen for its low cost, commercial reliability, compact packaging, and low noise equivalent power (NEP)[15]. In order to achieve this goal, several design parameters were varied to optimize detector responsivity. This thesis supports the development and testing of initial designs for optimized pixels in CMOS, including MOSFET, antenna, and layout variations and modifications for improved response. Subsequent chip design will ultimately lead to an ideal FPA design. Testing of the technology is carried out at the Chester F. Carlson Center for Imaging Science, RIT.

1.3. Terahertz Radiation Properties

THz radiation has unique properties as compared to other portions of the spectrum which is why it is so exciting to develop. This section outlines its properties to give some perspective for the subsequent application section of this document.

1.3.1. Non-Ionizing

THz radiation is non-ionizing, meaning it does not destructively interfere with human DNA molecules. Radiation that has enough energy to free electrons from atoms or molecules during collision is considered ionizing. Obviously, this is not ideal for humans as ionizing radiation damages living tissue including DNA. In extreme doses, this results in sickness, mutation, and cancers which can lead to death. Radiation with shorter wavelengths such as some ultraviolet, x-rays, and gamma rays are ionizing, while longer wavelength radiation including visible, infrared, microwaves, THz, and radio waves are non-ionizing (Figure 1.1). When THz radiation is compared to x-rays for example, THz has photon energies which are four orders of magnitude lower which is one of the reasons why THz imaging is of interest for the partial replacement of x-ray imaging.

1.3.2. Penetration

A primary advantage of THz is that many common materials are transparent or semi-transparent to its light. This includes some plastics, paper, cardboard, semiconductors, and human and biological tissues [25]. Imaging applications are ideal because of this, although further research is needed before the technologies are developed enough to compete with standardized systems such as x-ray, MRI, etc. THz radiation does not have quite the same penetrative power as millimeter wave or x-ray radiation although, which is one of the drawbacks for replacement of current systems. Efforts to combat this are usually answered with an active source system, vs a passive one. Example images utilizing this property can be seen in Section 1.4.3

1.3.3. Resolution

The Rayleigh criterion for minimum resolvable angle of a circular aperture (assuming that the aperture is large as compared to the wavelength) is:

$$\theta = 1.22 \frac{\lambda}{D} \text{ [rad]} \quad (1.1)$$

where λ is the wavelength and D is the Diameter of the circular aperture. This translates to an inverse relationship of resolution with respect to wavelength. Current millimeter wave scanner systems operate around 30 GHz and provide about 1 mm of spatial resolution at distances less than 1 m. By moving into the THz domain, not only is the resolution improved, but the aperture needed for the system can be made smaller. This makes the systems more compact, mobile, and covert if needed. The Safe VISITOR system (described in Section 1.4.3 for example, achieves a 1.5 cm resolution at 8 m with a 0.5 m diameter aperture (0.35 THz) [9, 26, 27]. X-ray systems however have better resolution based on the Raleigh criterion with a much shorter wavelength (higher frequency) than THz. A tradeoff for x-rays vs t-rays occurs with resolution versus penetration of the radiation, as well as ionizing energy considerations.

1.3.4. Scattering

THz light scatters less than that of shorter wavelength radiation. The probability of Rayleigh scattering is inversely proportional to the wavelength as λ^4 . Lord Rayleigh derived an expression for the scattered intensity as shown in Figure 1.2.

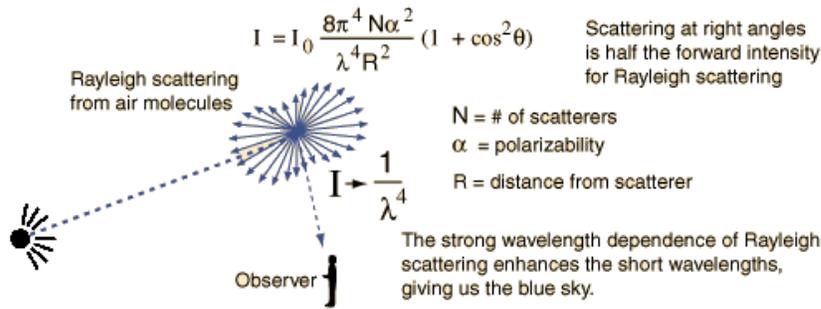


Figure 1.2.: Diagram and equations describing Rayleigh scattering. Essentially, light of different wavelengths (colors) scatters at different angles when coming into contact with air molecules. This is why the sky looks blue and sunsets look red/orange. Reprinted from [2].

For a wavelength of 1 mm (0.3 THz), a 10^{12} reduction in scattering seen as compared to visible light [25, 28]. Because of this reduction in scattering, a better image can be obtained.

1.3.5. Safety

Power levels of THz sources are continually climbing, especially with the recent technology push in the last few years. Safety and human effects of this newly developed area need to be studied further. Initially it seems as if THz radiation is completely safe since it is non-ionizing. This may not be the case however, since the technologies are fairly new and not yet fully tested against prolonged exposure. Although there have been reports of negative effects due to exposure, such as growth enhancements, wound healing, and changes in anxiety levels, further studies are needed to determine all of the effects and resulting limitations in exposure, especially with high power levels. The most recent investigations suggest that tissue damage is unlikely to occur below levels of 10 mW cm^{-2} [29].

1.3.6. Atmospheric Transmission

As previously described, developing technologies for the THz region of the spectrum proves to be a daunting task. So what are the actual capabilities of such a system should significant progress be made? Before this question can be answered, see Figure 1.3.

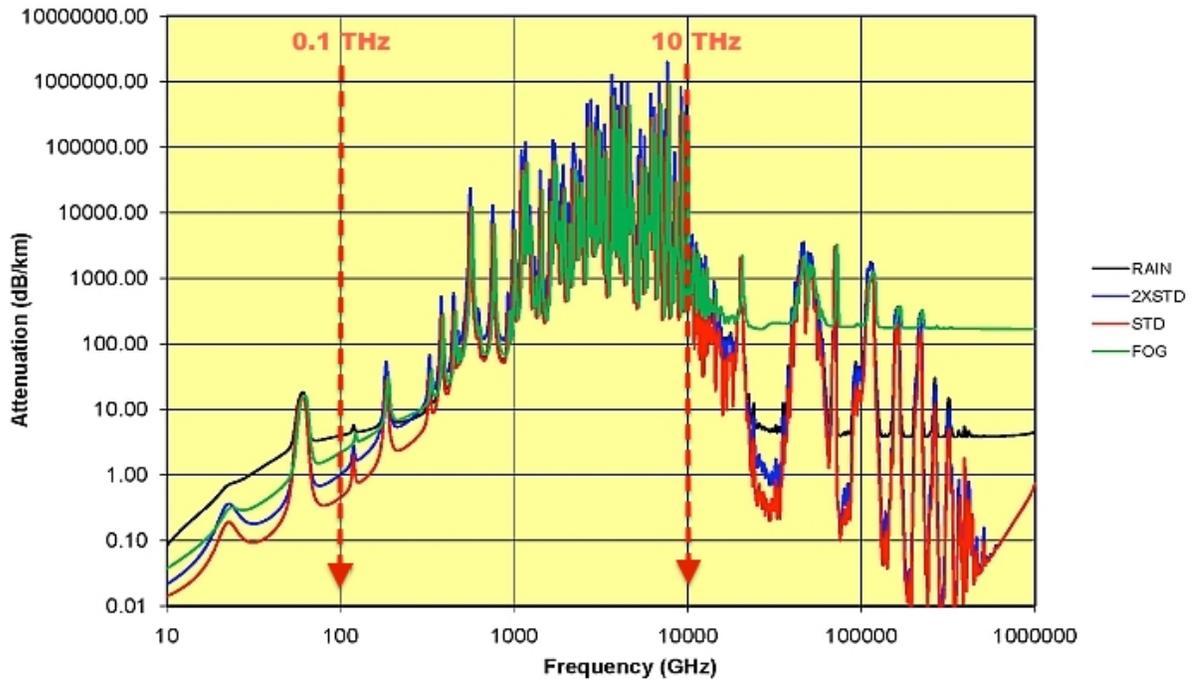


Figure 1.3.: Atmospheric attenuation of the THz spectrum. The boundaries of the region of interest are annotated. Adapted from [3].

Even if THz source power increases to much higher levels, say for an active communication system, absorption in the atmosphere is still prohibitive after a certain point. Armstrong looked at these numbers [4], and the results are telling. By analyzing the power levels and distances required for a communication system, anything further than 100 meters is near impossible for wavelengths greater than 1 THz, even with a petawatt of power. He describes this limitation as a “THz wall”, because no matter how much

you boost a signal to a receiver, essentially nothing gets through after 100 meters (see Figure 1.4). This occurs near 1 THz. Other interpretations exist however, and only time will tell once technologies develop enough to prove or disprove this theory.

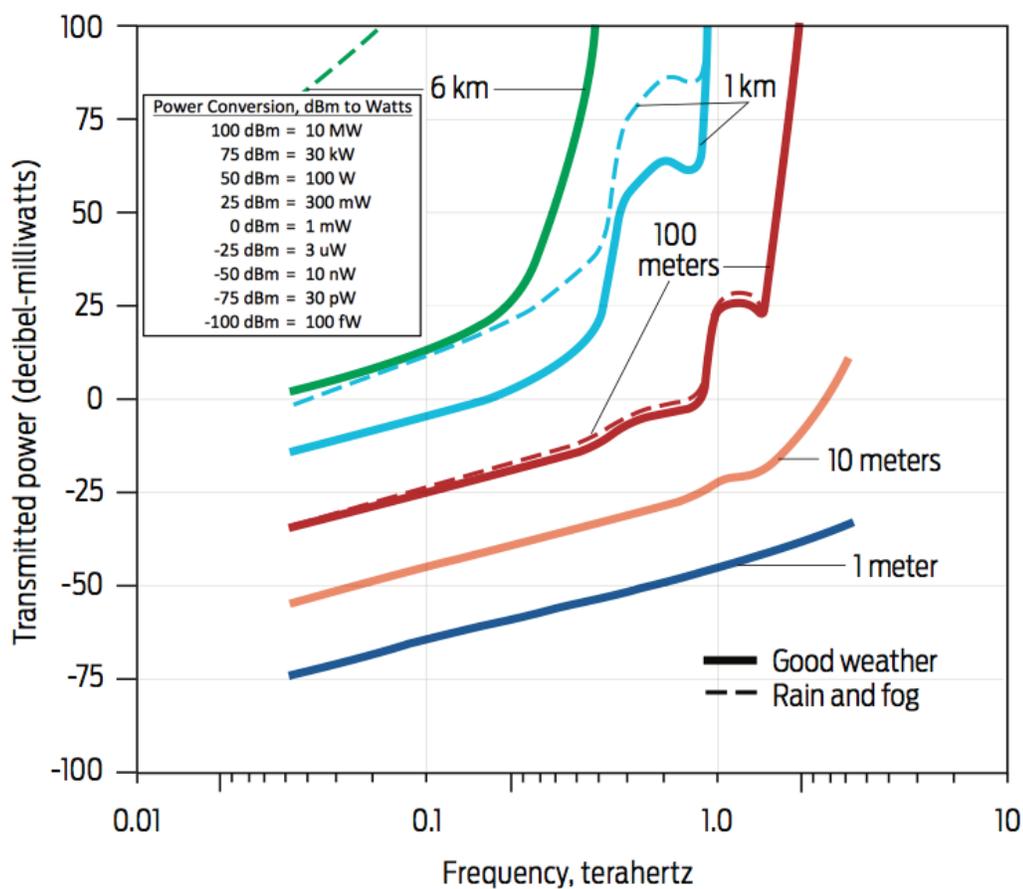


Figure 1.4.: The “THz wall”. Calculated for a communications link using fixed gain antennas and a horizontal line-of-sight path on the ground. Good and bad weather transmitted power are shown for frequencies between 35 GHz and 3 THz. Transmitted power is shown in decibel-milliwatts, or dBm, which denotes the amount of power required at the sending side to reach the receive end at the specified distance. Adapted from [4]. Inset: Power conversion chart from decibel-milliwatts to watts.

1.4. Applications

1.4.1. Astronomy

Astronomers were some of the first scientists to become interested in THz radiation for a very practical reason. About half of the luminosity and 98% of photons released since the occurrence of the Big Bang reside in the THz portion of the spectrum, specifically the interstellar dust that sits 14-140 K below the ambient background of the earth [15, 30]. Because of this and spectral attenuation, ground based systems in the THz will always have limited utility. This application is one of the most easily employed areas since the primary limitations of THz radiation are mitigated by putting the sensors in space. In a space vacuum, all the problems with absorption and transmission nearly go away. Figure 1.5 shows the spectral content of a star forming cloud. The 30 K blackbody represents the cloud dust and gas temperatures while the cosmic background is shown at 2.7 K. At longer wavelengths, the rotation spectrum of heavy molecules dominates the gas emissions of interest. This is where a THz detector could be used since the clouds are optically thin. At shorter wavelengths into the infrared, these dust clouds are optically thick, so one cannot see through them to detect the emission lines of interest. By using THz spectroscopy in space to look at these emission lines, scientists can gather information that can answer many unknown questions regarding the formation of galaxies, stars, and planets. It is also useful for the characterization of the Earth's upper atmosphere, to detect and map ozone holes [31].

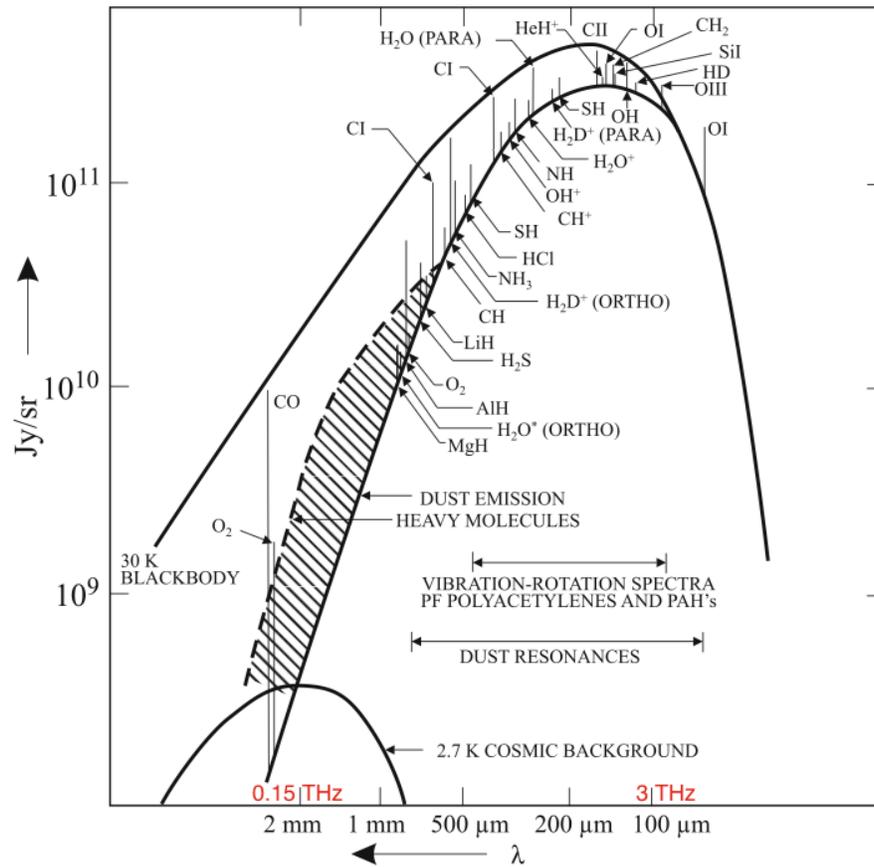


Figure 1.5.: Spectral content of a typical star forming cloud in the galaxy. A Jansky is a measure of spectral flux density, equivalent to: $1 \text{ Jy} = 10^{-26} \text{ W m}^{-2} \text{ Hz}^{-1}$. Adapted from [5].

1.4.2. Spectroscopy & Material Identification

Many non-metallic materials have strong absorption lines in the THz which allow for the detection and investigation their of physical properties. An example spectra for table sugar is shown in Figure 1.6.

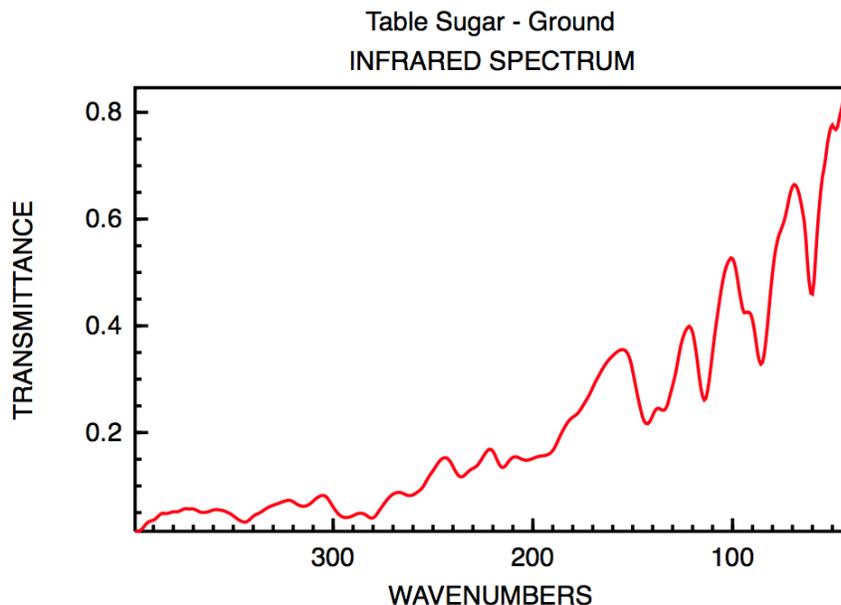


Figure 1.6.: Transmittance spectrum for table sugar in the THz. Reprinted from [6].

Coupled with the penetrative properties of THz, this opens up many applications for detection of chemicals and materials not only in terms of differentiation, but in determining their composition. Detecting pollutants, biological and chemical agents, and ancient artifact characteristics are some of the many applications.

1.4.3. Security & Defense

Law enforcement, security, and military applications are perhaps the foremost driver of THz technologies today. Airport security continues to be an ongoing problem due to increasing travel and the need for global connectivity. A THz security scanner could passively (with respect to the person) detect chemical and biological agents present on a passenger's person or in luggage, alerting security personnel to further investigate. This is a promising close range application since the radiation does not transmit well past 100 m, as previously stated in Section 1.3.6. It has the potential to replace current mm-wave

1. Introduction

and x-ray scanner systems as well, and provide a better resolution and detection product due to its ability to penetrate clothing and non-metallic materials. Digital Barriers, a UK based company has one of the first passive systems able to prove the utility of such a device as seen in Figures 1.7 and 1.8.



Figure 1.7.: Example ThruVision system concepts for security screening. ThruVision systems are active operating in the 0.25 THz wavelength region. Reprinted from [7]



Figure 1.8.: Example image from ThruVision system. The inset on the left shows an overlaid THz image and the presence of a handgun under the subject's clothing. Reprinted from [7]

Another important application is the screening of packages and mail. Dangerous substance detection is a high priority for prison systems and government facilities where large mail volumes make detection difficult. A new solution has been developed using THz detection, in a collaboration between the Fraunhofer Institute for Physical Measurement Techniques IPM, and Huber GmbH & Co.. The T-COGNITION system (Figure 1.9) uses active radiation between 0.1 and 4 THz from fiber-optic lasers and a dry air flooding system to provide a stable environment for detection. It uses a detection algorithm to perform a spectral analysis in 8 seconds on a single piece of mail [8].

1. Introduction



Figure 1.9.: Example image of T-COGNITION system. Software analysis shows the presence or absence of identified substances on a visible image. Reprinted from [8].

Additionally, a recent example of a prototype security screening system is the Safe VISITOR system, Figure 1.10. A German collaboration between the Institute of Photonic Technology, Supracon AG, and Jena-Optronik GmbH produced the system which is capable of 10 Hz video and centimeter resolution at <10 m. It uses a circular raster scan technique to gather multiple samples in order obtain good resolution.

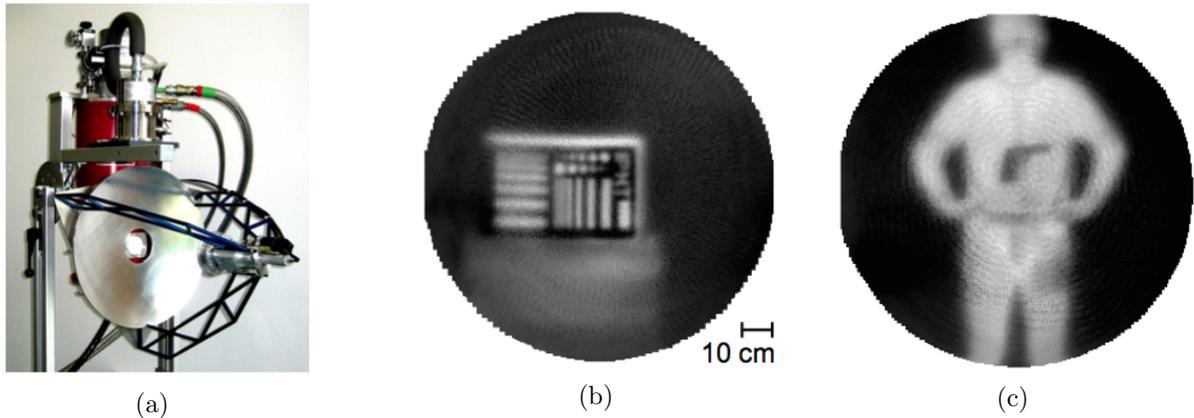


Figure 1.10.: Safe VISITOR system operates at 0.35 THz and capable of slow frame-rate video. Adapted from [9]. (a) Example camera setup. (b) THz image of resolution chart at 8 m. (c) THz image of person with concealed gun at 8 m.

Along with these novel systems, an increasingly important issue is the privacy implications of such images. THz images tend to reveal more detail underneath clothing than the general public would like. This is a tradeoff between security and privacy. The same concern exists with millimeter-wave systems as well, and software adaptations of the images have been used to make such security screening systems more ethically acceptable [8].

1.4.4. Communications

The field of communications and its continually increasing need for more bandwidth will always push technologies to find new ways of meeting this demand. Channel congestion in ground based systems is also a problem, so new frequency options would provide some relief. Typically, device bandwidth is about 10% of the frequency on which it operates [28]. This means that the higher the operating frequency, the more potential bandwidth is available. Current indoor and outdoor systems are not meeting bandwidth demand, and in most cases, wired connections are still used. Operating at THz frequencies has

1. Introduction

the potential to meet this demand in order to move to an ideal target bandwidth of 100 Gbps for a wireless link [32].

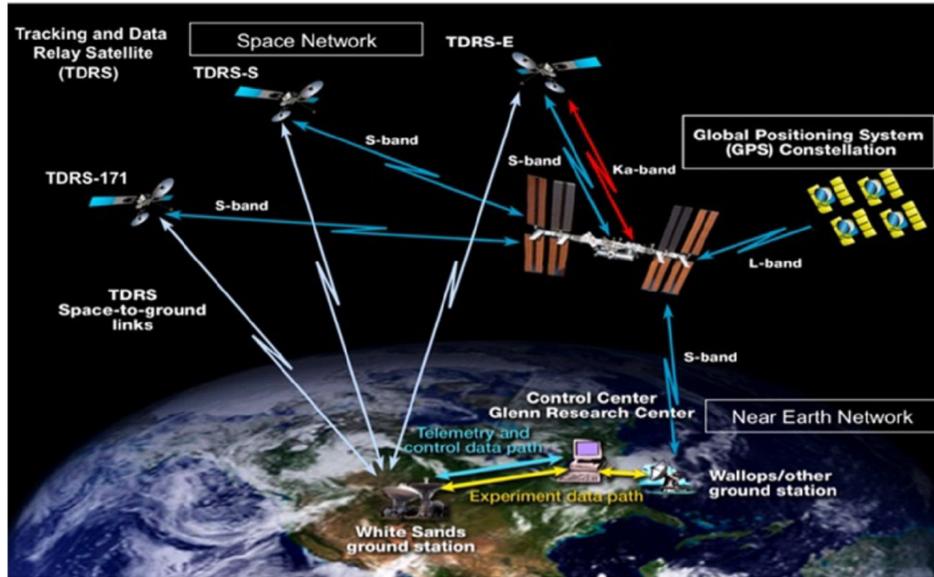


Figure 1.11.: Current communications system layout for human spaceflight missions. S (2-4 GHz), Ku (12-18 GHz), and Ka (26-40 GHz) band systems are the primary operating bands. Adapted from [10].

Space-based communications is another application for THz communications, as shown in Figure 1.11. Again, because of the space environment, atmospheric attenuation is not an issue. Currently, systems have to parse information on-board the spacecraft and choose what to send over the link. Increased bandwidth of a THz link would allow for more information to flow between systems. THz systems would also have the advantage of simplified design compared to current S (2-4 GHz), Ku (12-18 GHz), and Ka (26-40 GHz) band systems [10]. These links have a highly complex design in order to push the required data over limited channels. THz systems and the inherent physics of the wavelength would allow for a smaller form factor, which is always attractive to space applications for the never ending chase of weight savings.

1.4.5. Medicine

THz technologies have the potential to replace x-ray and CT scan based medical imaging in many areas. Dental x-rays, detection of skin conditions, cancer, and biological tissue identification are some of the uses. Figure 1.12 shows a TeraView system image of a tooth cavity.

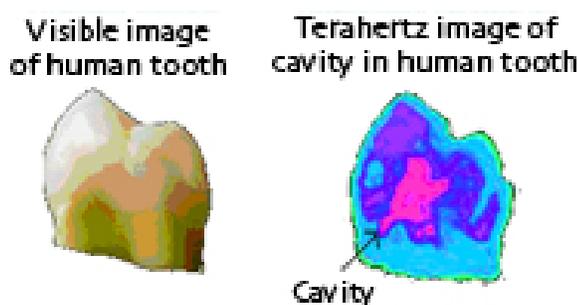


Figure 1.12.: Visible and THz image of human tooth using a TeraView system. The absence of visual features on the tooth surface makes early detection of tooth decay difficult. X-rays, one of the accepted methods used to detect decay, only reveals the problem at a relatively late stage, when drilling and filling is the only method available to halt the decay. If decay can be detected early enough it is possible to reverse the process without the need for drilling by the use of either fissure sealing or remineralization. Reprinted from [11].

THz radiation is non-ionizing, meaning it will not destructively interfere with human tissue (see Section 1.3.1). Also, it can only penetrate a limited distance due to the strong H_2O content in the human body. In-vivo non-invasive blood glucose monitoring is another interesting use of THz for medicine. Because of unique absorption features, a THz system could potentially replace pesky finger pricking glucose systems by remotely sensing the content in blood [33].

1.4.6. Manufacturing

Manufacturing and quality control are a good fit for THz systems. Using its penetrative properties, a THz imager can detect defects in semiconductors, plastics, or other manufactured materials that would not be otherwise visible. One example would be for detecting the thickness of tar applied to roofing shingles. Currently tar thickness is physically measured after the application process, deeming product runs with an incorrect thickness as unusable. If the process does not apply enough tar, the product fails. If too much tar is applied, than the manufacturer is losing profit. By monitoring this thickness in realtime with THz sensors, the thickness can be adjusted accordingly. This is possible because of the THz's unique penetrative properties. (This application was offered by Traycer Systems, Inc).

1.5. Summary

Using THz radiation as a new regime for imaging and remote sensing has been sought after for many decades. Recently though, technology breakthroughs such as the decreasing feature size of semiconductor devices, has allowed for first practices of THz concepts. Unique properties and spectral profiles of THz radiation allows us to realize potential applications. Security, defense, and communications are the primary technology paths that are driving this development, but a broad range of applications is what makes the THz research field full of energy and promise. The remainder of this thesis will focus on the specific application of THz detection in MOSFETs and efforts to develop an FPA imaging system applicable to many of the aforementioned applications.

2. Theory

This section will explain the theory and mathematics aimed at describing THz detection using the channel region of a MOSFET. First though, some basic concepts must be understood.

2.1. MOSFET Basics

The concept of the field effect transistor was first patented in 1925 by Julius Lilienfeld. A MOSFET is a four terminal semiconductor device (source, drain, gate, and base) which operates in the simplest form as a switch, Figure 2.1. Charge carrier (electrons in our case) density under the gate is controlled by applying a voltage to the gate contact. This forms a conductive semiconductor channel (inversion layer) between the source and the drain. Typically the substrate (or base) is grounded with the source (common-source mode) and the device is controlled with the three other contacts. The channel length, L , of these devices were once on the order of a few micro-meters (micron), but improvements in production technology have scaled this down to tens of nano-meters. The latest Intel processors for example, are 22 nm which allows for more transistors to be packed on a chip to provide better performance, latency, and power consumption[34]. The increase of this transistor density is commonly known as Moore's Law, which predicts a factor of 2 increase every two years (this is commonly misquoted as 18 months due to Intel executive David House referring to effective computer performance gains [35]).

2. Theory

For an n-channel device (electrons as carriers vs holes) in the off state (no gate voltage), the source and drain contacts are isolated from each other, and inversely in the on state (when the gate voltage is above a threshold) the conducting channel forms allowing current to flow between the source and drain. The electrons enter and exit the channel at the source and drain contacts [12].

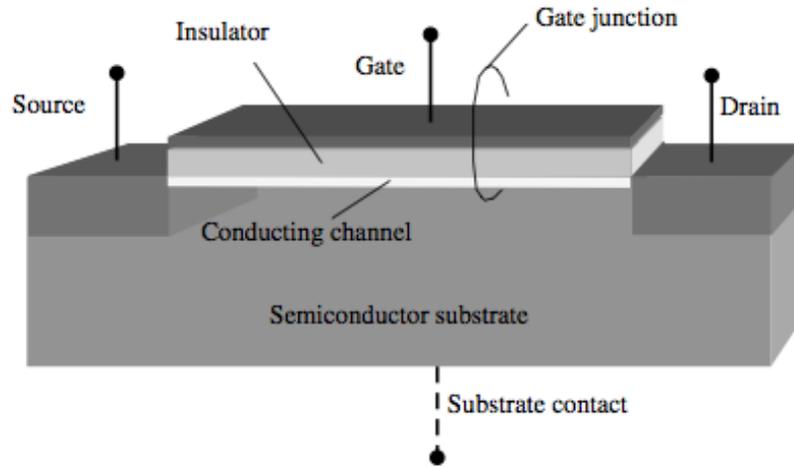


Figure 2.1.: MOSFET diagram. The semiconductor substrate in an n-channel device is traditionally p-type silicon with the source and drain implants doped n+ silicon. The insulator layer below the gate is silicon dioxide, and the contacts are made of a conductive material like polysilicon or silicide. Reprinted from [12].

The gate threshold voltage, V_{TH} is defined as the voltage at which a strong inversion occurs, therefore forming the conductive channel. In some cases, a drain voltage, V_{DS} is applied as another biasing parameter. Assuming the gate voltage is above threshold, $V_{GS} > V_{TH}$, this produces an asymmetric channel region, as shown in Figure 2.2.

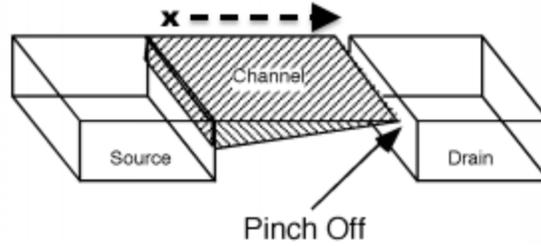


Figure 2.2.: The channel region of a MOSFET under a pinch off state. The asymmetry is formed by the presence of a drain bias voltage. Too much drain bias will force the channel into a pinch off state, and the inversion layer is no longer maintained. Adapted from [13].

As you travel linearly across the channel from the source to the drain, the channel-source voltage increases and reaches V_{DS} at the drain terminal. Corresponding to this voltage increase along the channel, is a decrease in the gate channel bias V_{Gx} , where x represents the position along the channel. At the drain, the gate-channel voltage is equal to the difference between gate and drain voltage, $V_{Gx} = V_{GD} = V_{GS} - V_{DS}$.

2.1.1. Gradual Channel Approximation

In order to explain the plasmonic detection mechanism, William Shockley's Gradual Channel Approximation (GCA) of the MOSFET must first be understood [14, 36]. It is termed GCA because of the assumption that the voltages vary gradually along the channel of length, L , from the drain to the source. Vertically they vary quickly from the gate through the channel to the substrate. This approximation is used in order to model the I-V characteristics of the MOSFET device. The GCA holds for long channel devices where the gate length to channel depth ratio is large, i.e. $L/h(x) \gg 1$. However, newer short channel devices have more complex characteristics, but the approximation is still useful for modeling and analysis. Figure 2.3 shows an example MOSFET exhibiting the GCA conditions.

2. Theory

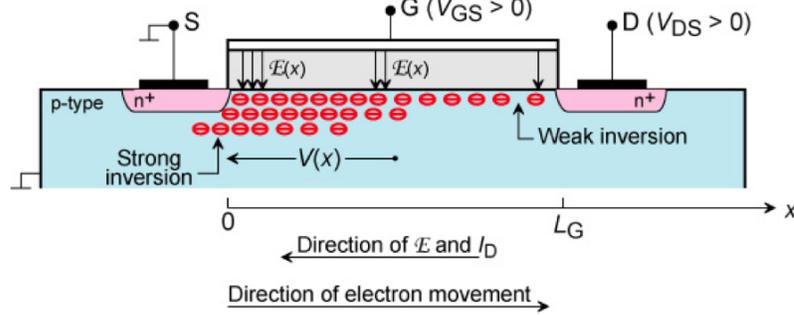


Figure 2.3.: MOSFET exhibiting GCA conditions. The charge varies gradually along the channel of length, L , with a strong inversion at the source and a weak inversion at the drain. This asymmetry is due to the drain bias voltage, V_{DS} . Reprinted from [14].

An important assumption for the GCA is that the current through the base and gate is approximately zero, $i_B = i_G \approx 0$. Therefore the drain voltage and base voltage as $V_{DS} \geq 0$ and $V_{BS} \leq 0$ provide bounds for the validity of the GCA. With the GCA, the problem reduces to simply finding the current through the channel, i_D . The drain current, i_D now has three regions (modes) of operation. They are cutoff, linear, and saturation mode, as shown in Figure 2.4. Linear mode is also referred to as the ohmic or triode region. In cutoff, current is not able to flow through the channel, $i_D = 0$. This is the case when the gate voltage is below threshold (also known as sub-threshold), so a complete inversion layer is not present, $V_{GS} < V_{TH}$. In linear mode, the MOSFET operates like a resistor (hence ohmic mode), and the gate voltage controls the amount of current flowing through the channel as:

$$i_D = \mu_e C_{ox} \frac{W}{L} \left((V_G - V_{TH}) V_{DS} - \frac{V_{DS}^2}{2} \right) \quad (2.1)$$

where μ_e is the electron mobility, C_{ox} is the capacitance of the gate oxide per unit area, W is the gate width, and L is the gate length. In saturation (also active mode), a strong

2. Theory

inversion takes place and completely forms the channel. Current flow now reaches a limit, and no longer depends on V_{DS} . The gate voltage is now the primary control for current flow.

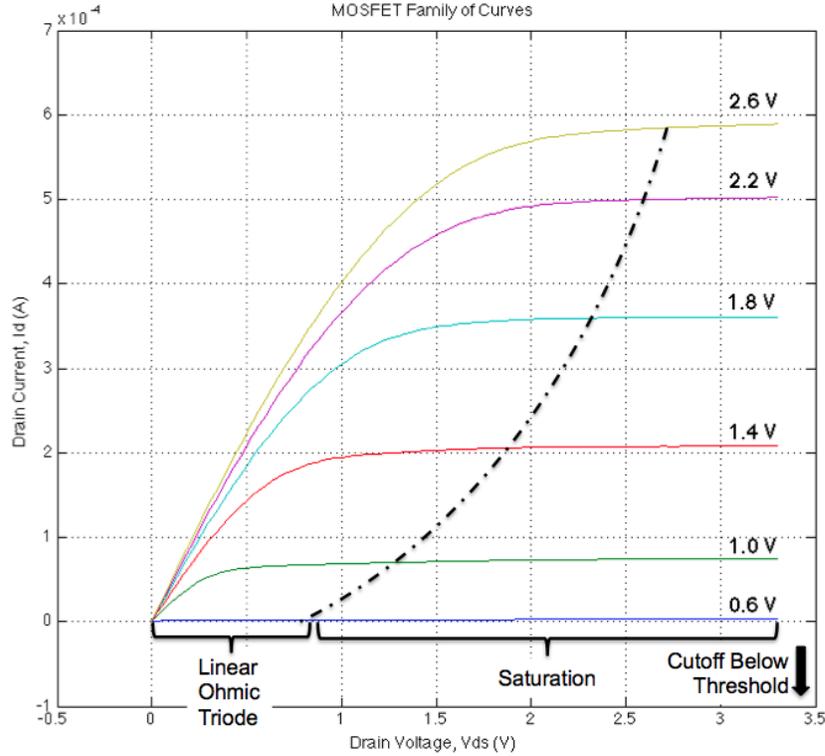


Figure 2.4.: The MOSFET Family of Curves. This plot of drain current vs drain voltage describes the MOSFETs operating characteristics. The gate voltage for each curve is annotated. The dashed line represents the saturation voltage, where $V_{GS} - V_{TH} \leq V_{DS}$. Example data is from one of the THz test MOSFETs to be described in Section 4.

2.1.2. Conductance & Transconductance

Two important parameters of a MOSFET's operation are the conductance and the transconductance. These are used to characterize operation, and also to track changes in operation over time. The conductance, g_d (the reciprocal of resistance) is the ratio of

2. Theory

the change in drain current to change in drain voltage for a fixed gate voltage:

$$g_d = \left. \frac{\delta i_D}{\delta V_{DS}} \right|_{V_{GS}} \quad (2.2)$$

$$g_{d,triode} = \mu_e C_{ox} \frac{W}{L} (V_{GS} - V_{TH} - V_{DS}) \quad (2.3)$$

$$g_{d,sat} = 0 \quad (2.4)$$

An example of conductance is shown in Figure 2.5 for the T-5 bowtie MOSFET described in Section 4.

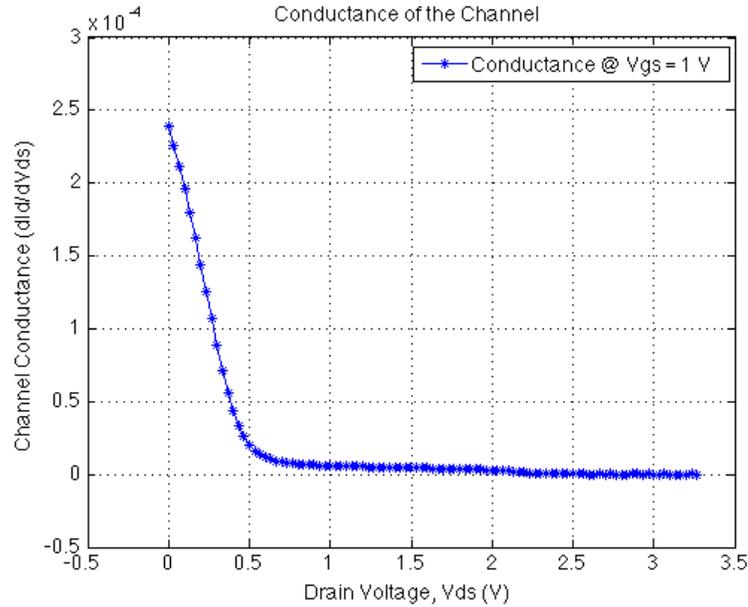


Figure 2.5.: The Conductance Curve. Represents the change in i_D vs the change in V_{DS} . V_{GS} is fixed at 1 V. V_{TH} for this device is ≈ 0.5 V. Example data from test MOSFETs.

$g_{d,triode}$ is the conductance in the triode regime and approaches zero in the saturation regime, represented by $g_{d,sat}$. In triode, the drain current changes rapidly with any change in V_{DS} . In saturation, there is almost no change. Since the channel resistance

2. Theory

is the reciprocal of conductance, it is easy to see that in saturation a nominal channel resistance is achieved, but in triode this inversely proportional to V_{DS} .

Similar to conductance, the transconductance, g_m is the ratio of the change in drain current to the change in gate voltage for a fixed drain voltage:

$$g_m = \left. \frac{\delta i_D}{\delta V_G} \right|_{V_{DS}} \quad (2.5)$$

$$g_{m,subthreshold} = 0 \quad (2.6)$$

$$g_{m,triode} = \mu_e C_{ox} \frac{W}{L} V_{DS} \quad (2.7)$$

$$g_{m,sat} = \mu_e C_{ox} \frac{W}{L} (V_{GS} - V_{TH}) \quad (2.8)$$

Figure 2.6 shows an example of transconductance from the test MOSFETs described in Section 4. where g_m has three regimes. Below the threshold of the device, V_{TH} , g_m is zero, but swiftly increases in the triode regime. As the channel begins to saturate, an exponential like decrease appears represented by $g_{m,sat}$.

Conductance and transconductance are both given in units of inverse ohms, or sometimes mhos or siemens:

$$1 \text{ } \mathcal{U} = 1 \text{ } S = 1 \frac{A}{V} \quad (2.9)$$

2. Theory

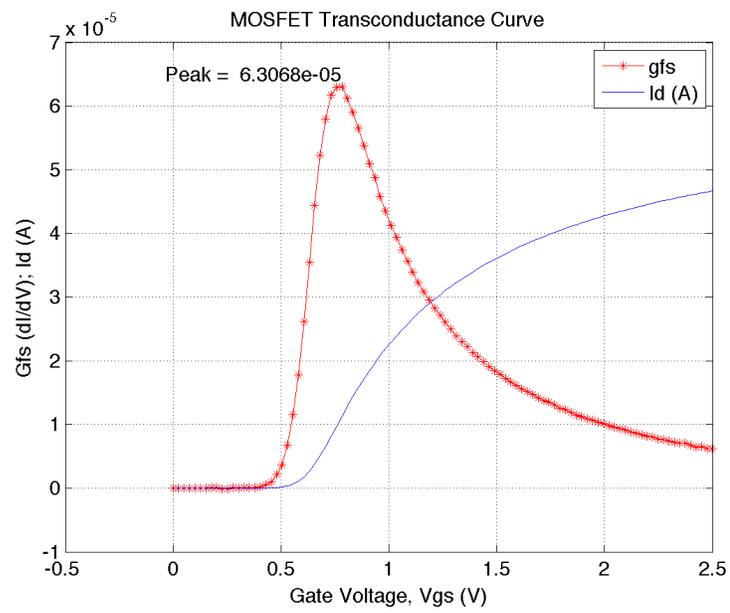


Figure 2.6.: The Transconductance Curve. Represents the change in i_D vs the change in V_{GS} . V_{DS} is fixed at 0.1 V. Same test MOSFET as Figure 2.5.

2.2. Plasmonic Detection Mechanism

The following analysis of plasmonic detection will follow the theory initially provided by Dyakonov and Shur in the early 1990s [37, 38]. In the MOSFET channel under proper biasing conditions, the electron system, termed electron fluid, can act as a resonator for THz radiation in the form of plasmon waves as seen in Figure 2.7.

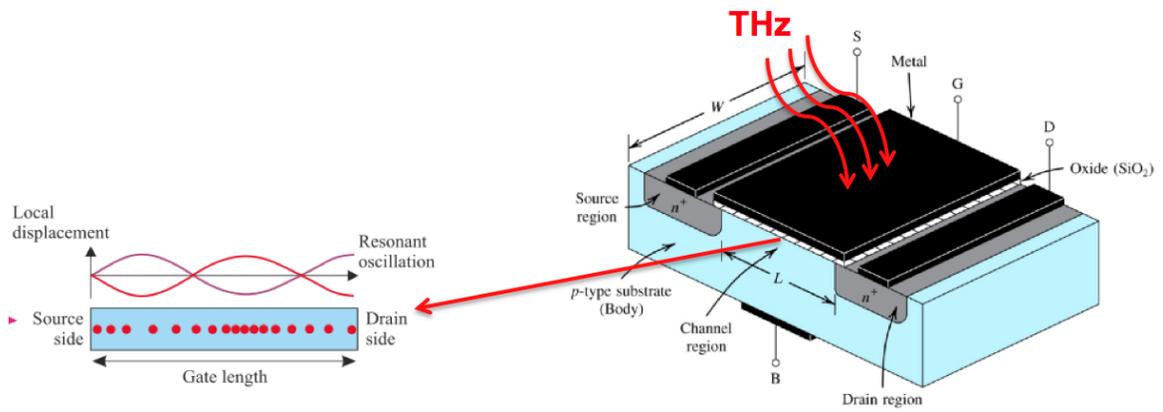


Figure 2.7.: MOSFET diagram of signal detection. Under proper biasing, incoming THz radiation produces an AC signal between the gate and source region. This is rectified and detected as a DC voltage change between the source and the drain. Channel length, L , affects the response frequency and determines whether a resonant or non-resonant detection occurs. Adapted from [15].

The current flow through the MOSFET is then disturbed from these plasmon waves and can be used to quantize the incoming radiation. Because of Dyakonov and Shur's analogy to shallow water waves, hydrodynamic equations for fluid can be used to describe the motion [39]. Electron concentration in the channel is described by:

$$n_c = \frac{CU}{q} \quad (2.10)$$

where n_c is the electron concentration in the channel, C is the gate-channel capacitance

2. Theory

per unit area, U is the gate-channel voltage, and q is the elementary charge. The Euler equation of motion for 2D electron fluid is given by [38]:

$$\frac{\delta v}{\delta t} + v \frac{\delta v}{\delta x} + \frac{q}{m} \frac{\delta U}{\delta x} + \frac{v}{\tau_r} = 0 \quad (2.11)$$

where $\delta U/\delta x$ is the longitudinal electric field in the channel, $v(x, t)$ is the local electron velocity, q is the elementary charge of an electron, and τ_r is the electron momentum relaxation time. Equation 2.11 is solved in conjunction with the continuity equation (local energy conservation) below with the gradual channel approximation from Section 2.1.1:

$$\frac{\delta U}{\delta t} + \frac{\delta}{\delta x}(Uv) = 0 \quad (2.12)$$

where Equation 2.10 provides a replacement for the electron concentration n_c with the gate-channel voltage U . An expression for the photoinduced response is then derived [40, 41]. This is dependent however, on whether the MOSFET is operating in a resonant or non-resonant regime. First, the following parameters and values are defined:

The electron momentum relaxation time represents the average free time between ionic collisions, τ_r as [19]:

$$\tau_r = \frac{\mu_n m_e^* m_0}{q} \text{ [s]} \quad (2.13)$$

where $\mu_n [\frac{cm^2}{Vs}]$ is the electron mobility in the channel, $m_e^* = 0.19$ is the electron effective mass, $m_0 = 9.11E - 31[kg]$ is the electron rest mass, and $q = 1.6E - 19[coul]$ is the elementary charge of an electron. Values of μ_n can range from 250-1400 $[\frac{cm^2}{Vs}]$ for silicon, but a typical value is 500.

The plasmon wave velocity s_p is defined through [16, 19]:

$$s_0 = \sqrt{\frac{\eta k_B T}{m_e^* m_0}} \left[\frac{m}{s} \right] \quad (2.14)$$

$$s_1 = \frac{q U_0}{\eta k_B T} \quad (2.15)$$

$$s_p = \sqrt{s_0^2 (1 + e^{-s_1}) \ln(1 + e^{s_1})} \left[\frac{m}{s} \right] \quad (2.16)$$

where $\eta = 1.5$ is the ideality factor, $k_B = 1.38E - 23[\frac{J}{K}]$ is the Boltzmann constant, $T = 300[K]$ is the temperature, $U_0 = V_{GS} - V_{TH}[V]$ is the voltage swing between V_{GS} , the gate voltage, and V_{TH} , the threshold voltage.

The conductivity of the channel, σ_c is driven by τ_r through[39]:

$$\sigma_c = \frac{n_c \tau_r q^2}{m_e^* m_0} [S] \quad (2.17)$$

2.2.1. Resonant Detection

A sufficiently short channel length MOSFET will act as a resonant detector with the fundamental plasmon wave frequency, ω_0 , and eigen plasmon wave frequencies, ω_n are given as:

$$\omega_0 = \frac{\pi}{2L} \sqrt{\frac{q(V_{GS} - V_{TH})}{m_e^*}} = \frac{\pi s_p}{2L} [Hz] \quad (2.18)$$

$$\omega_n = \omega_0(1 + 2n) \quad n = 1, 2, 3, \dots \quad (2.19)$$

Resonant detection is realized when the resonance quality factor $\omega_0 \tau_r > 1$. In this regime, the channel length is short enough that a standing plasmon wave is formed in the channel promoting signal amplification. The wave propagates between the source and the drain

regions being reflected back and forth. For longer channel length devices, the wave decays before it reaches the drain side and the AC induced signal only exists near the source [39].

A resonant device can have a high responsivity due to increase signal amplification in the channel. Also, the response is narrowband, so response to particular frequencies can be tuned. Unfortunately, parameters for resonant response in modern silicon MOSFETs is driven by the short gate length and high mobility. The gate length is manufacturable, but electron mobilities are too low to support this resonant response at room temperature conditions. Other semiconductor applications using high electron mobility transistors (HEMT, HFET, MODFET) are more suited for resonant detection [40].

2.2.2. Non-Resonant Detection

When the resonance quality factor $\omega_0\tau_r \ll 1$, a non-resonant detector response is realized. Here the plasmon wave is dampened before reaching the drain side of the MOSFET channel (overdamped), resulting in a broadband response of lower responsivity. This is the case in our application with a larger gate length, $L = 350$ nm. Several mathematical applications of this case predict voltage response ([15–17, 39–42]) and a generalized equation for the non-resonant case incorporating below and above threshold response is provided in [16, 39].

The resulting photoinduced voltage, ΔU is determined through:

$$Q = \frac{L}{s_p} \sqrt{\frac{\omega_{thz}}{2\tau_r}} \quad (2.20)$$

$$\Delta U = \frac{qU_a^2}{4m_e^*m_0s_p^2} \left[\frac{1}{1 + \kappa e^{-s_1}} - \frac{1}{[1 + \kappa e^{-s_1}]^2 [\sinh^2(Q) + \cos^2(Q)]} \right] [V] \quad (2.21)$$

where κ is a dimensionless parameter related to the gate leakage current ($\kappa \ll 1$), and

Q is a simplification term given by:

$$\kappa = \frac{j_0 L^2 m_e^* m_0 q}{2 C_{ox} \tau_r \eta^2 k_B^2 T^2} \quad (2.22)$$

where j_0 is the gate leakage current density and C_{ox} is the gate-oxide capacitance per unit area.

2.2.3. Theoretical Calculation

Using Equation 2.21 and supporting parameters, a theoretical calculation for reponse can be obtained. The unknown variable in this calculation however, is the AC-induced signal on the gate, U_a , which is coupled via the antenna. This must be interpreted and estimated.

First, the following parameters are defined:

<u>Symbol</u>	<u>Value</u>	<u>Units</u>	<u>Description</u>
η	1.5	Unitless	Ideality Factor
κ	0.01	Unitless	Dimensionless Gate Leakage Parameter
k_B	$1.38 * 10^{-23}$	J K^{-1}	Boltzmann Constant
m_e^*	0.19	Unitless	Electron Effective Mass in Silicon
m_0	$9.11 * 10^{-31}$	kg	Electron Rest Mass
q	$1.62 * 10^{-19}$	C	Elementary Charge
T	300	K	Temperature
μ_n	500	$\text{cm}^2 \text{V}^{-1} \text{s}^{-1}$	Electron Mobility
U_a	60	mV	AC Photoinduced Voltage
V_{GS}	0.5	V	MOSFET Gate-Source Voltage
V_{TH}	0.485	V	MOSFET Threshold Voltage

Next, the following results are calculated using Equations 2.13 - 2.22:

Symbol	Value	Units	Description
s_p	$2.14 * 10^5$	m s^{-1}	Plasmon Wave Velocity
τ_r	$5.4 * 10^{-14}$	s	Electron Momentum Relaxation Time
U_0	-0.015	V	Gate Voltage Swing ($V_{GS} - V_{TH}$)
ω_0	0.962	THz	Fundamental Plasmon Wave Frequency

Using a U_a of 60 mV, a theoretical response $\underline{\Delta U = 59.1 \text{ mV}}$ is realized. This agrees with the T-1 results from Section 4. The method of radiation coupling through the antenna, coupling efficiency, and incident power on the effective aperture of the antenna are what is missing from these calculations. The front end portion of the detection chain is what is missing from the aforementioned theory as well. This also does not account for any of the source region extensions outlined in Section 3.

2.2.4. Application and Enhancement

Several methods exist for enhancement of the rectified signal in experimental applications. An antenna is commonly used to enhance response to the frequency of operation, as well as modified biasing schemes. Developing theory for these more complicated configurations is ongoing [43–45], but tends to be application specific. A simplified explanation for the approach in this section is that the incoming radiation of frequency ω_{thz} hits the MOSFET channel and induces a plasmonic oscillation, and corresponding AC voltage, U_a in the gate. Given that the biasing and dimensional parameters are constructive, the AC voltage wave is rectified through the MOSFET to the drain as a DC photoresponse, ΔU . This is illustrated in Figure 2.8.

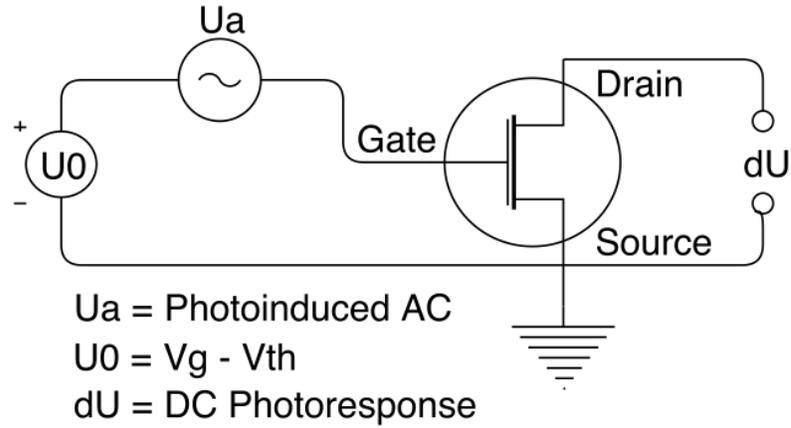


Figure 2.8.: Theoretical operation of MOSFET for THz detection. U_a is the photoinduced AC voltage, U_0 is the DC gate voltage bias from threshold, and dU is the rectified DC photoresponse voltage.

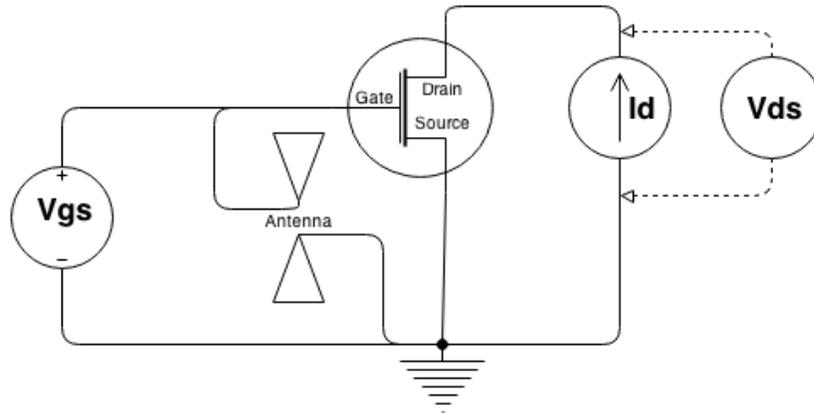


Figure 2.9.: Experimental application of MOSFET for THz detection. The antenna intends to enhance response, with the bias applied through V_{GS} and i_D . Response is measured through V_{DS} .

The MOSFET parameters must provide an asymmetry between the source and the drain in order to promote this rectification. This is achieved through physical construction of the MOSFET, circuit configurations, an antenna, or biasing schemes [15, 19, 39]. The asymmetry in the described test chip (Section 3) is provided by the antenna connec-

tions between the gate and source (Figure 2.9), and enhanced with an optimal drain bias current, i_D . This represents the approach taken for the experimental process described in Section 3.4 and corresponding results shown in Section 4. First though, the test chip and pixel design is described in Section 3.2.

3. Fabricated Test Chip Design & Evaluation

3.1. Research Collaborators

RIT, UR, and Exelis Geospatial Systems partnered to develop the prototype THz FPA and imaging system as previously described [24]. Current systems mainly consist of bulky technology, including large pulsed laser systems and primarily laboratory based setups. A silicon CMOS based technology was chosen with the goal of developing a practical imaging system. The THz FPA technology being tested is uncooled and employs the direct overdamped, plasmonic detection described in Chapter 2. The silicon CMOS MOSFETs are each coupled to an individual micro-antenna. The focus of this section is characterization of individual test transistors which will support future FPA pixel design. Testing of the technology is carried out at the Chester F. Carlson Center for Imaging Science, RIT.

3.2. Chip & Pixel Description

The chip used is designed and fabricated in a 0.35 μm silicon CMOS process. The foundry vendor utilized is the Taiwan Semiconductor Manufacturing Company Limited (TSMC) and the fabrication house is MOSIS. The chips are then sent to Malaysia for packaging. This fabrication process was chosen in order to match the frequency response of the detection MOSFET with the source frequencies of interest (see Section 2). Although

limited source options are currently available to the research group, the 188 GHz Gunn diode matches well with expected response frequencies. Figure 3.1 shows this response vs. frequency relationship for a 130 nm process.

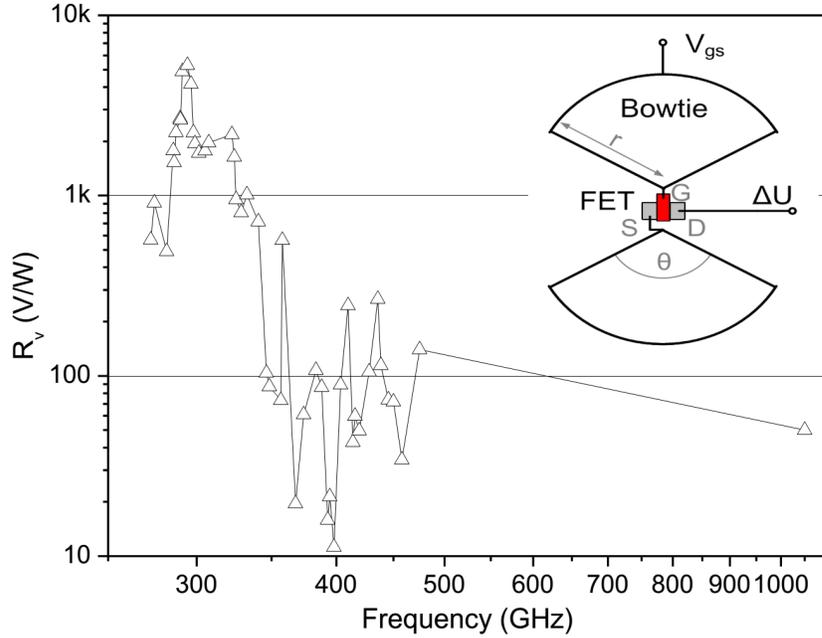


Figure 3.1.: Responsivity of THz MOSFET vs. frequency. The triangles show the measured points for a MOSFET with a gate length of $L = 130$ nm and coupled with a bowtie style antenna. Notice the peak response near 300 GHz. For a larger MOSFET such as the 350 nm under test, the response will be lower [16]. The inset shows a diagram of the MOSFET and antenna for reference. Adapted from [17].

On the chip are four test imager arrays and five test transistors. These ‘test’ transistors are connected directly to outputs for characterization without clocking electronics. Present work is focused on characterizing the response from these five test transistors. Figure 3.2 shows a micrograph of the test chip with the test transistors located on the bottom edge.

3. Fabricated Test Chip Design & Evaluation

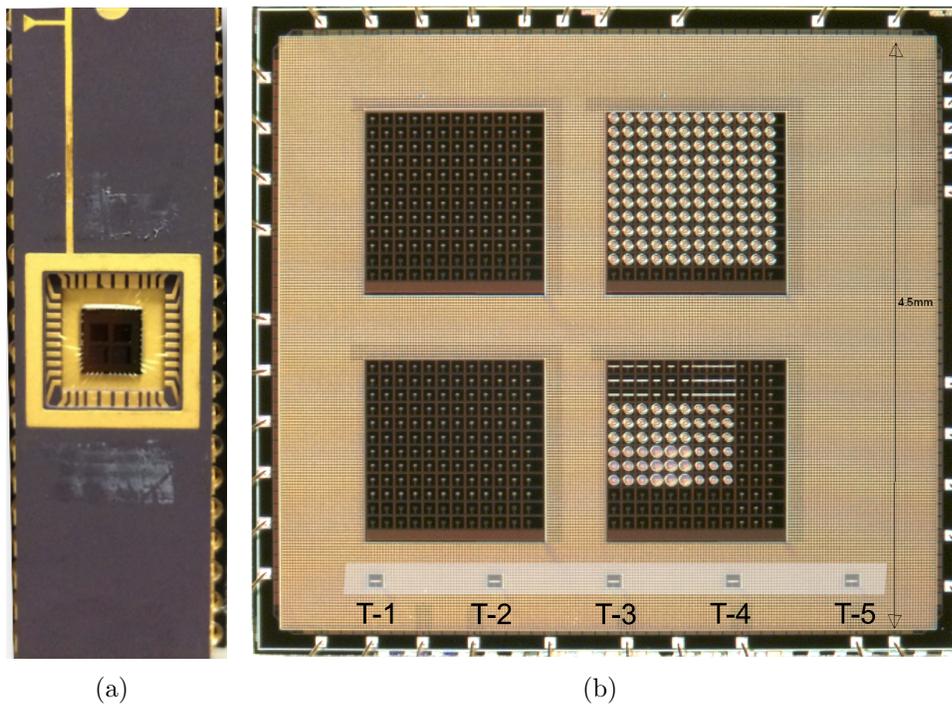


Figure 3.2.: Test Chip 1 (a) 40-pin DIP package for test chip. Individual pins allow selection of test transistors without clocking electronics. (b) Micrograph of the test chip. The five test transistors on the bottom edge are being characterized in this work.

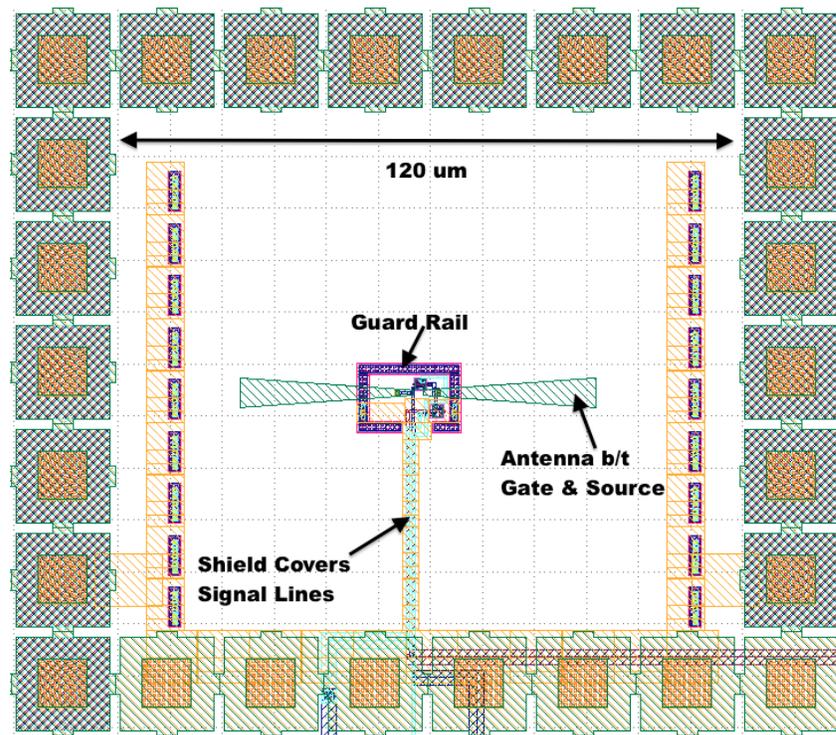


Figure 3.3.: GDS file snapshot of T-1 pixel design. Bowtie antenna has dimensions of $70\ \mu\text{m} \times 6\ \mu\text{m}$ with a $2\ \mu\text{m}$ gap.

3. Fabricated Test Chip Design & Evaluation

Figure 3.3 shows a snapshot of the pixel design. The pixel area is $120\ \mu\text{m}$ square. The antenna is connected to the gate on one side and the source on the other. The guard rail surrounding the detection MOSFET is connected to ground and isolates the detection region. The shield layer is grounded as well, and covers the switching MOSFET for the gate connection. The connections out to the pads are covered by the shield to isolate coupling to the pixel region. Two versions of the chip were designed to study antenna variations. The test transistors are identical with the variations of the antennas being either bowtie or spirals as shown in Figures 3.4 and 3.5. The five test transistors on each chip all have the same channel dimensions of $0.35\ \mu\text{m} \times 2\ \mu\text{m}$.

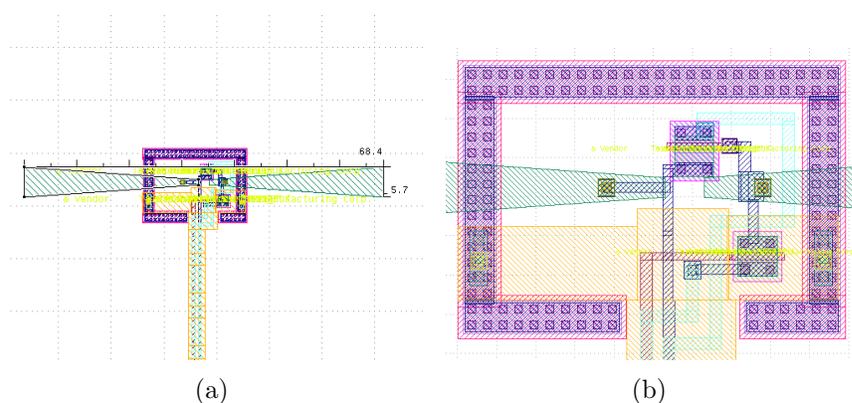


Figure 3.4.: GDS snapshot of a test transistor with bowtie antenna. (a) Full pixel and antenna $5.7\ \mu\text{m} \times 68.4\ \mu\text{m}$, $2\ \mu\text{m}$ Gap). (b) Blowup view of the design with the guard rail around the edge. The orange (lower) shaded area represents the metal shield layer, with the antenna and detection MOSFET located in the top half of the image.

3. Fabricated Test Chip Design & Evaluation

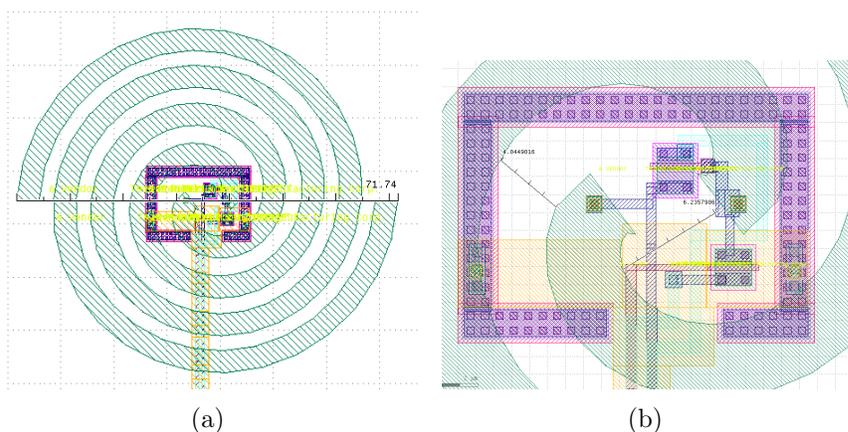


Figure 3.5.: GDS snapshot of test transistor with spiral antenna. (a) Full pixel and antenna with a $72\ \mu\text{m}$ diameter. (b) Blowup view of the pixel. These pixels are identical to the bowtie antenna pixels, the only difference being the antenna.

The only difference between the five test transistors is the size and shape of the MOSFET source region. The source region was varied for the purpose of investigating the effects of adding a small source-degeneration resistance on the overall responsivity. The added resistance was varied in $10\ \Omega$ increments from $0\ \Omega$ (T-1) to $40\ \Omega$ (T-5). This resistance is implemented through a source extension with length L_s as shown in Figure 3.6.

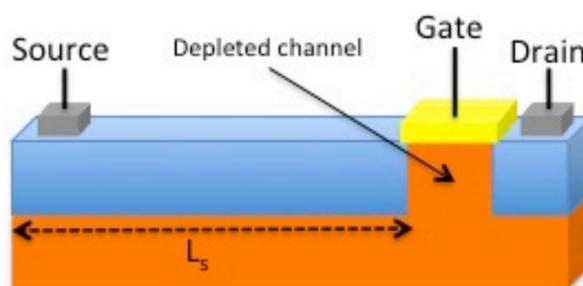


Figure 3.6.: MOSFET with extended source region. L_s represents the length of extension of the source from the gate.

The source extension is varied in $0.5\ \mu\text{m}$ increments from $0\ \mu\text{m}$ to $2\ \mu\text{m}$. This creates

an effect analogous to impedance matching for the incoming signal (further explanation in Section 4). The difference in the source drain extension as implemented on the chip is shown in Figure 3.7.

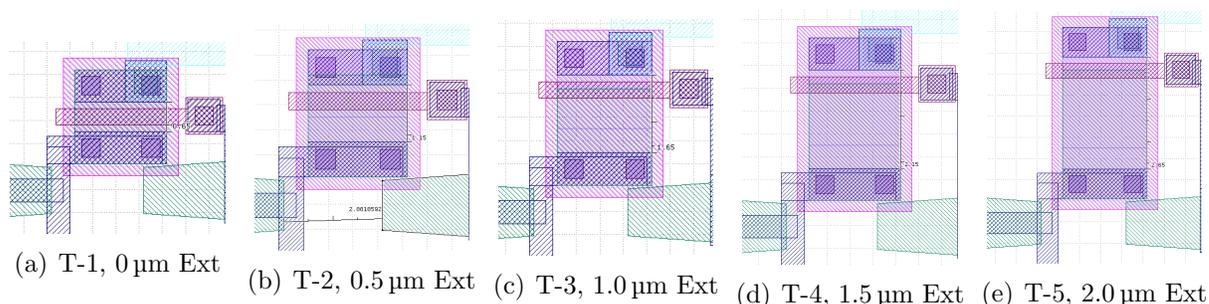


Figure 3.7.: The five bowtie test MOSFETs. The only difference in the five test transistors (T-1:5) is the source region extension which varies from 0 μm - 2 μm . The T-1 transistor has the shortest extension while the T-5 transistor has the longest extension. This notation corresponds to the discussion in the results Section 4.

3.3. Enclosure and Fanout Board Configuration

Bias and measurement signals for the chip are fed through a fanout board as shown in Figure 3.8. A 3.3 V regulator provides power for the chip and switching MOSFETs, while direct connections to the gate, drain, and source provide biasing and measurements.

The fanout board is mounted to the inside of the enclosure via standoff posts and nylon hardware to isolate the board from the enclosure. The enclosure is made out of aluminum and the removable cover has a conductive seal which creates a full Faraday cage. A second removable cover was fabricated to mount over the detector aperture and has a high resistivity silicon window attached. The silicon window prevents any visible light from entering the enclosure during test. These two covers are shown in Figure 3.9.

3. Fabricated Test Chip Design & Evaluation

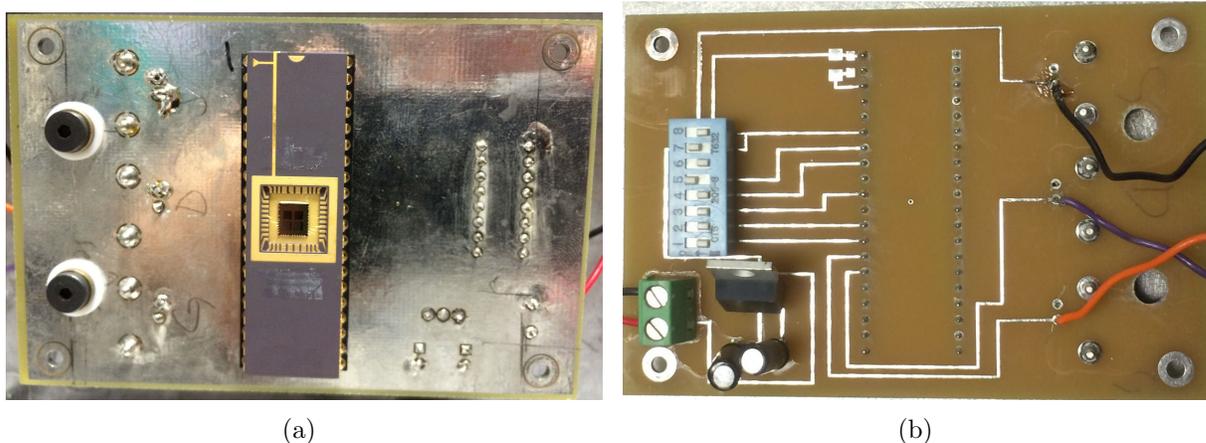


Figure 3.8.: The test chip is accessed through a fanout board. (a) Front view of the fanout board including the chip mounted in socket. (b) 9V battery connections, feed through capacitors, and a 3.3V regulator to provide power for the chip. Gate, Drain, and Source leads are fed directly to feedthrough capacitors on the enclosure.

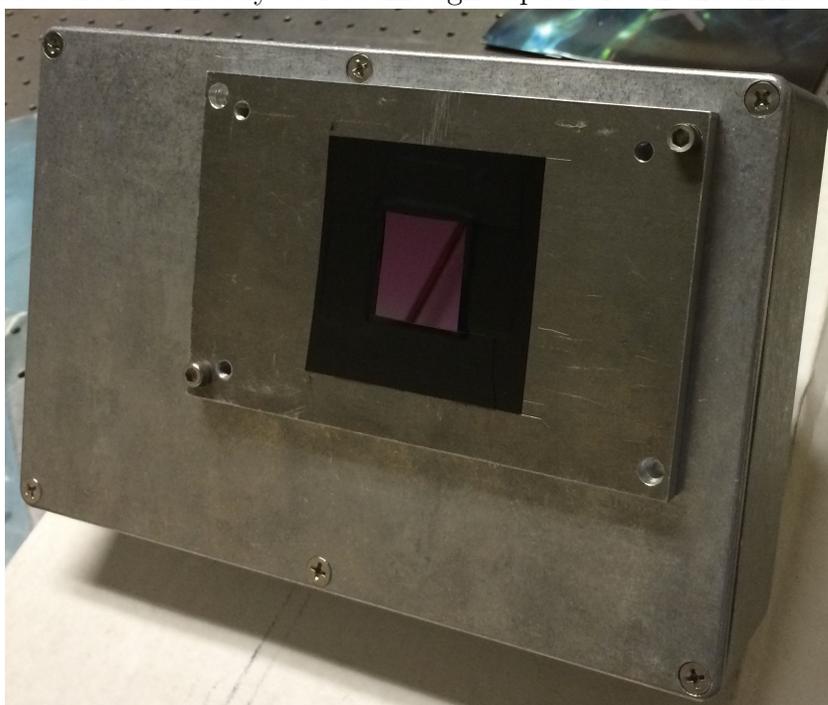


Figure 3.9.: Enclosure Front View. A removable silicon windows and front cover provide easy access to the fanout board and chip inside. A conductive seal creates a Faraday cage around the detector.

3. Fabricated Test Chip Design & Evaluation

Because the chip substrate is made of silicon (the same material used in visible CCDs, cell-phone cameras, etc), it also responds to visible light. It is important to assure that any response being measured is from the THz radiation only and not other wavebands. Four feed-through capacitors mounted through the rear of the enclosure provide connections for incoming signals from two TwinAx low noise shielded twisted pair cables. One cable connects to the fanout board gate and the other connects to the drain. The source connection is grounded which is used as the reference ground for the gate and drain signals. An external power switch allows the internal 9 V battery to be disconnected without removing the cover. A 1/4-20 threaded hole in the center of the rear enclosure panel allows for various mounting options including a rotation stage aligned the detector center axis. Also mounted to the rear of the enclosure is a banana jack. This is connected to the enclosure itself for use as a ground (independent of the reference ground used for the signal cables) in noise rejection configurations. This is most commonly connected to the measurement equipment ground to prevent RF coupling.

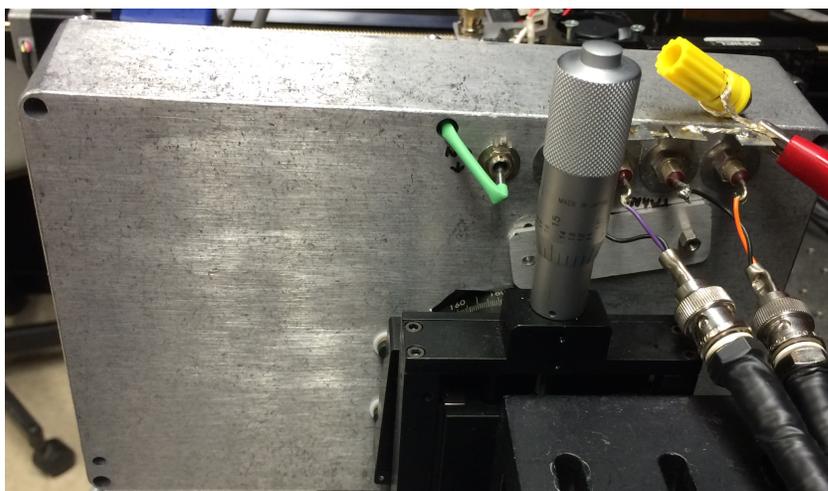


Figure 3.10.: Enclosure Back View. The rotation stage is centered at the detector center-axis for rotational purposes. The switch controls 9 V power to the board.

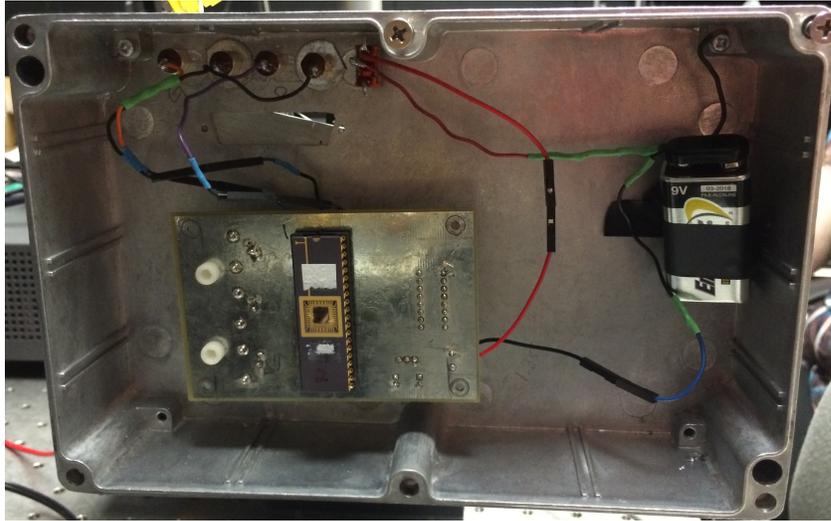


Figure 3.11.: Enclosure Internal View. Standoffs and nylon washers mount the board approximately 1" from the back of the enclosure.

3.4. Experiment Description

The transistors are biased using a Keithley 2602 Source Measurement Unit (SMU) which connects to the test enclosure via low noise shielded twisted pair cables. The enclosure is mounted on XYZ and rotation stages for alignment purposes. The SMU is controlled via a MATLAB serial interface for applying bias sweeps and relaying data. A high speed shutter in front of the detector enclosure is controlled through digital I/O by the SMU. In the simplest form, the detector is biased by the SMU, and measurements are taken with the shutter open and closed to determine response. Several biasing schemes are used to test the MOSFETs. Table 3.1 shows a matrix of possible biasing schemes performed on each MOSFET. The radiation source is a ≈ 50 mW, 188 GHz Gunn diode from Virginia Diodes. An example of the test setup is shown in Figure 3.12.

3. Fabricated Test Chip Design & Evaluation

Table 3.1.: SMU biasing schemes. Source/measure denote whether the SMU is providing signal or measuring signal. Voltages are either fixed or swept from 0 - 3.3 V. A measurement occurs for each increment of a given sweep. For each measurement, the SMU can perform a specified number (typically 100) of sub-measurements at an integration rate (NPLC) for a given static bias condition. Test 3 and 4 produced the strongest detection signal and are described in the previous theory section.

	V_{GS}	V_{DS}	i_D
Test 1	Source Sweep	Source Fixed	Measure
Test 2	Source Fixed	Source Sweep	Measure
Test 3	Source Sweep	Measure	Source Fixed
Test 4	Source Fixed	Measure	Source Sweep

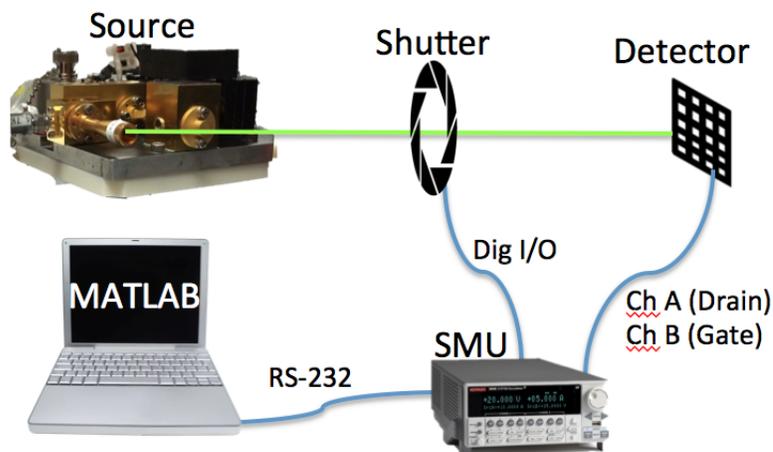


Figure 3.12.: Experimental test setup for detector characterization and beam profiling. The enclosure is translated via XYZ and rotation stages. The SMU is commanded via MATLAB serial interface for bias conditions, shutter position, and data transfer.

Typical bias conditions as in Table 3.1 Test 3 include providing a DC voltage to

the gate-source, V_{GS} , near the threshold region, and a drain bias current, i_D . The photoresponse voltage ΔU is then measured as a difference between the DC drain-source voltage, V_{DS} with the shutter open and closed, as shown in Figure 2.9.

3.5. Terahertz Signal Data Acquisition

This section explains a typical data acquisition process for a photoinduced response. First, the transconductance of the MOSFET is taken to ensure proper operation and repeatability (seen in Figure 4.2). For a given sweep, the SMU can source a range of specified MOSFET parameters between ± 3.3 V based on the chip circuitry. A gate voltage, V_{GS} is applied, along with a drain bias current, i_D . The response is measured as the drain source voltage, V_{DS} .

For each test, the following sequence is used to obtain the desired measurements:

1. Transconductance test prior to measurements
2. Enable the radiation source
3. Close the shutter
4. Enable SMU sources
5. Take X number of measurements: V_{GS} , V_{DS} , i_D
6. Open the shutter
7. Take X number of measurements: V_{GS} , V_{DS} , i_D
8. Close the shutter, turn off sources
9. Increment to next source voltage and repeat

All of these tests use an NPLC (Number Power Line Cycles) integration rate setting of 0.1. Since we are on 60 Hz power, this works out to a measurement frequency of $60/0.1 =$

600 Hz or intervals of 1.7ms. Once the MOSFET is biased, there is a 3 second delay for the circuit to settle, and then measurement begins. For each step in the V_{GS} voltage sweep, 100 samples are taken in succession with the shutter closed first, and then the shutter open. An example of the raw voltage measurements with respect to time are shown in Figure 3.13. Each pair of measurements corresponds to a gate voltage. Each measurement set has 100 samples taken at 600 Hz. The data shown in Figures 3.13 - 3.17 are from a response sweep of the T-5 bowtie MOSFET with $i_D = 0.15 \mu\text{A}$.

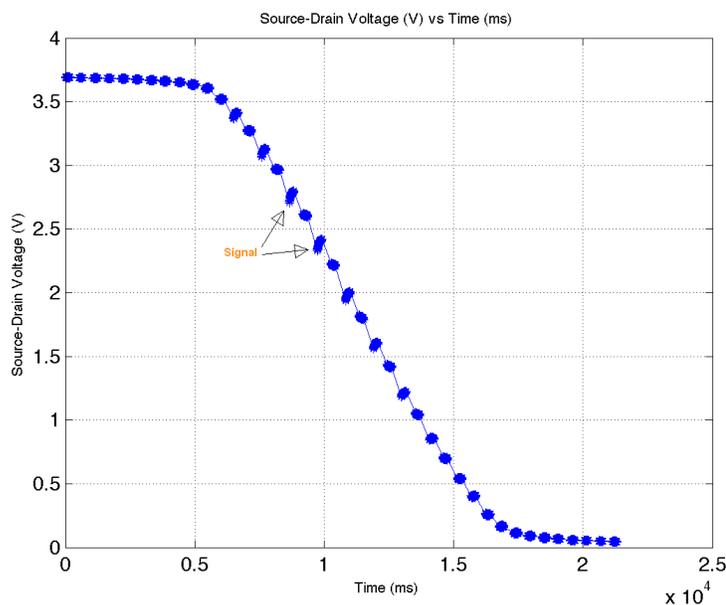


Figure 3.13.: T-5 bowtie example data. Raw voltage measurements. V_{DS} Measurements vs. Time

The averages of the 100 samples for the shutter closed and open are shown in Figures 3.14 and 3.15.

3. Fabricated Test Chip Design & Evaluation

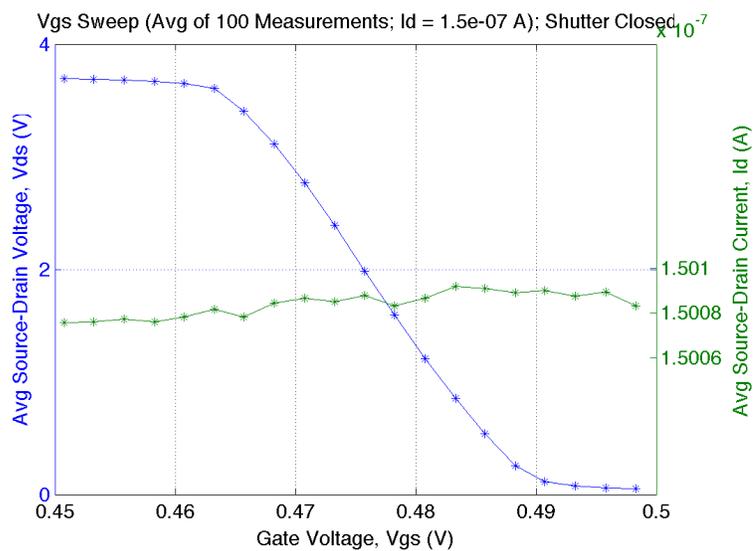


Figure 3.14.: T-5 bowtie example data. Shutter closed: V_{DS} and i_D .

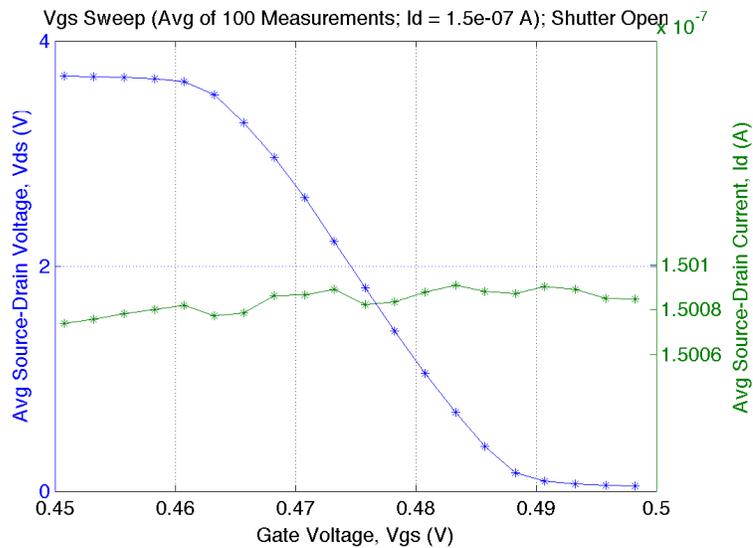


Figure 3.15.: T-5 bowtie example data. Shutter Open: V_{DS} and i_D .

The difference of the shutter open and closed measurements are plotted with respect to gate voltage to see the response as seen in Figure 3.16. Notice how the distribution changes near the response threshold. The difference is a simple subtraction of the closed and open measurements. Specifically, the i -th element of the shutter open vector for that measurement set is subtracted from the corresponding i -th element of the shutter closed vector. So the 100 samples for closed and open are lined up and subtracted with the assumption that the averaging will reduce the noise. Ideally, a correlated double sampling (CDS) approach would be taken to reduce the noise further, but is too complicated to implement with this test setup. Therefore the noise in this data is uncorrelated and includes all of the system noise. The average of the difference data from Figure 3.16 is shown in Figure 3.17.

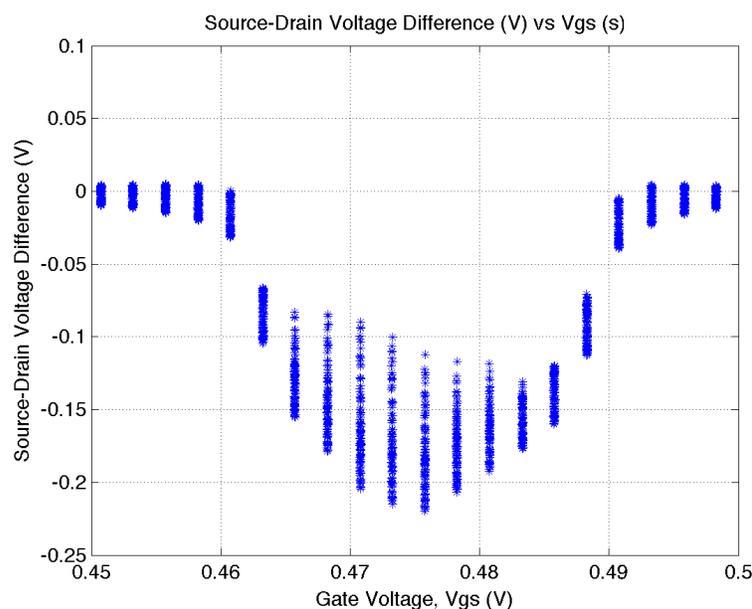


Figure 3.16.: T-5 bowtie example data. The signal ΔU vs V_{GS} (100 measurements per step).

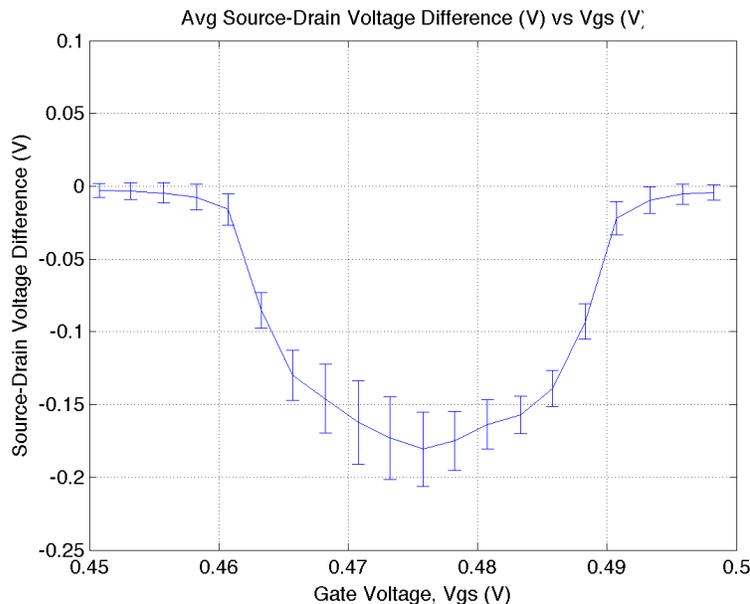


Figure 3.17.: T-5 bowtie example data. The average signal ΔU and noise as σ

As seen in Figure 3.16, the gate voltage range has been adjusted to catch the signal response curve. These ranges are different for each set of bias values for each MOSFET. This sweep takes approximately 2 minutes to run from sourcing to data reduction. Keeping enough samples for proper noise analysis and signal shape resolution is important, but large amounts of data transferred over a serial connection makes this troublesome at some point. Samples near this acquisition rate, delay setting, and interval are close to ideal for most of the data presented. For example though, in Figure 3.16, the sampling rate is actually resolving the signal voltage ramp near $0.47 V_{GS}$. The signal level has not settled yet in the detection circuit before sampling begins. This is adding to the noise of the signal measurement, and can be easily mitigated by adding a small delay between measurement sets.

4. Results & Discussion

Results described in this section are taken with the five test MOSFETs with the bowtie antennas (T-1 through T-5), unless otherwise described as a spiral antenna configuration. The spiral antenna MOSFETs did not produce nearly as significant of a response. This is possibly due to the smaller gap between the antenna arms in the spiral configuration. That is, the THz radiation ‘sees’ this as a contiguous metal plate instead of an antenna.

4.1. MOSFET Characterization

4.1.1. Transconductance

Before each test event, the proper operation and connection of the transistors is verified via a transconductance curve. For a more in-depth discussion of transconductance refer to Section 2. Figure 4.1 shows the transconductance curves for the bowtie antenna MOSFETs T-1 and T-5.

4. Results & Discussion

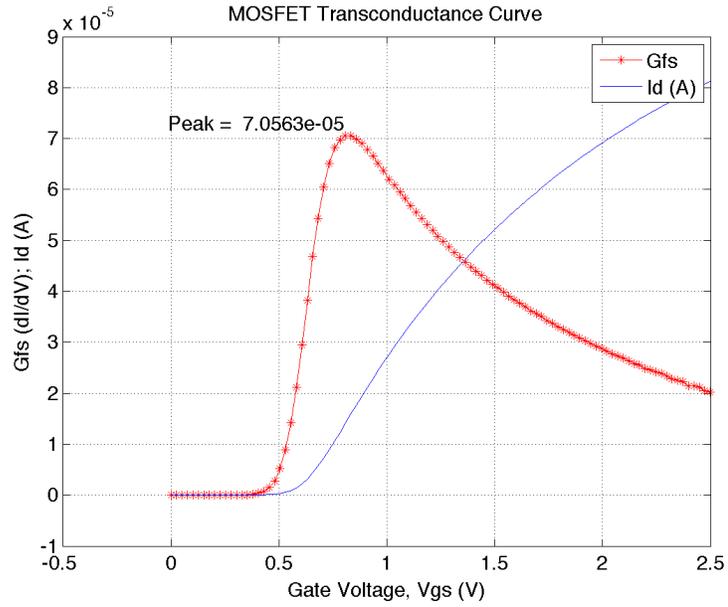


Figure 4.1.: T-1 Transconductance Curve.

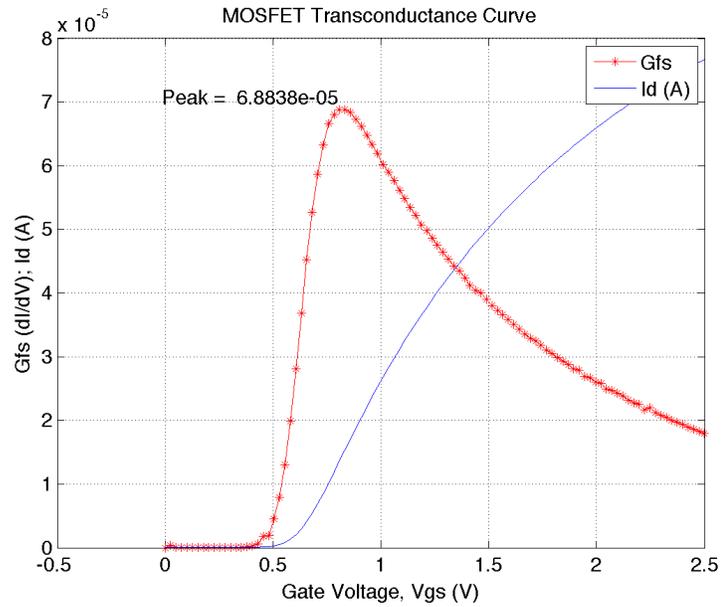


Figure 4.2.: T-5 Transconductance Curve.

4.1.2. Channel Conductance & Resistance

Another characteristic that is monitored during testing is the channel resistance of the MOSFETs with as it corresponds to V_{DS} and V_{GS} . In Figures 4.3 and 4.4, the T-5 transistor resistance is characterized. This is determined through measurement by $\Omega_{channel} = \Delta V_{DS} / \Delta i_D$, or also by taking the reciprocal of the conductance (i.e. Figure 4.5). Note that under incident THz radiation the resistance of the channel is changing due to the changing V_{DS} . The threshold of this device is near 0.5 V, which results in a resistance of about 10 M Ω . This will become more important as an array imaging system is realized if the decision is made to design a similar current source on chip for biasing.

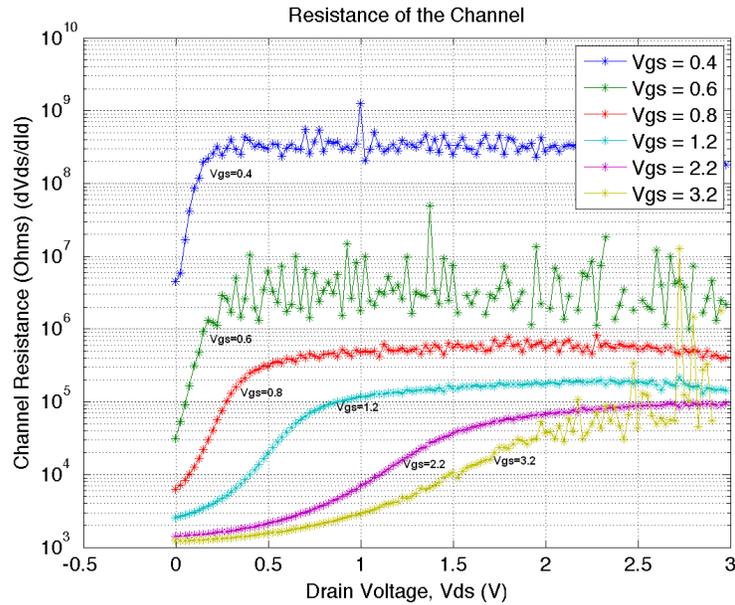


Figure 4.3.: T-5 Channel Resistance. The resistance of the channel as a function of drain voltage. Each series represents a different gate voltage bias.

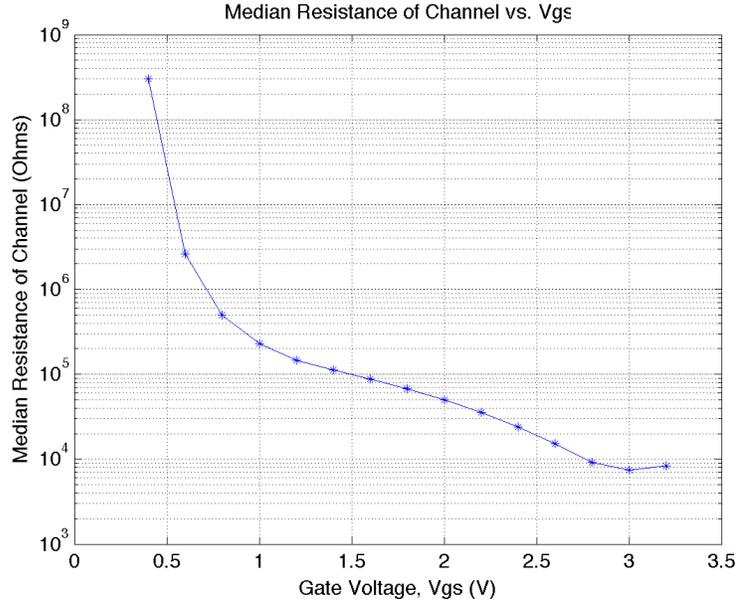


Figure 4.4.: T-5 Channel Resistance. The median resistance as a function of gate voltage.

4.1.3. Thermal Noise

The method used for collecting data consists of various sources of noise since measurements are taken at DC versus using a lock-in technique. A significant component at room temperature to consider is the thermal noise of the MOSFET. The thermal noise of the MOSFET can be estimated as a homogenous resistor when $V_{DS} \approx 0$. By picking a small V_{DS} , we can ensure the MOSFET is in the linear region where this approximation will hold. As V_{DS} increases, the resistivity of the channel changes as a function of distance between the source and the drain. This is due to the asymmetry of the channel due to the bias voltage (refer to Figure 2.2). This thermal noise can be approximated as the Johnson noise of a equivalent resistor:

$$i_{D,Therm} = \sqrt{\frac{4k_B T \Delta f}{r_0}} = \sqrt{4k_B T g_0 \Delta f} \quad (A) \quad (4.1)$$

4. Results & Discussion

where $k_B = 1.38 * 10^{-23}$ is Boltzmann's constant in JK^{-1} , T is the temperature in K, Δf is the bandwidth of the measurement in Hz, r_0 is the equivalent resistance of the source-drain channel in Ω , and $g_0 = \frac{1}{r_0}$ is the conductance of the source-drain channel in S.

A biasing point is selected in order for the MOSFET to be in the linear region where our noise analysis will occur. Figure 2.4 shows this linear region. A gate voltage, $V_{GS} = 0.5 \text{ V}$ is selected as it is near where we expect to see THz returns. A drain bias voltage, $V_{DS} = 25 \text{ mV}$ is selected since it is close to zero to uphold our resistor comparison, and is in the linear region of the FET. By plotting a line to fit the I-V curve, we can reduce the conductance of the source-drain channel, g_0 from the slope. Figure 4.5 shows this conductance as a function of V_{DS} , along with its resistance counterpart, r_0 , in Figure 4.6. Noted on the resistance graph is the point selected ($25 \text{ mV } V_{DS}$) for this data comparison. The equivalent resistance at these bias parameters is $\approx 366 \text{ k}\Omega$.

4. Results & Discussion

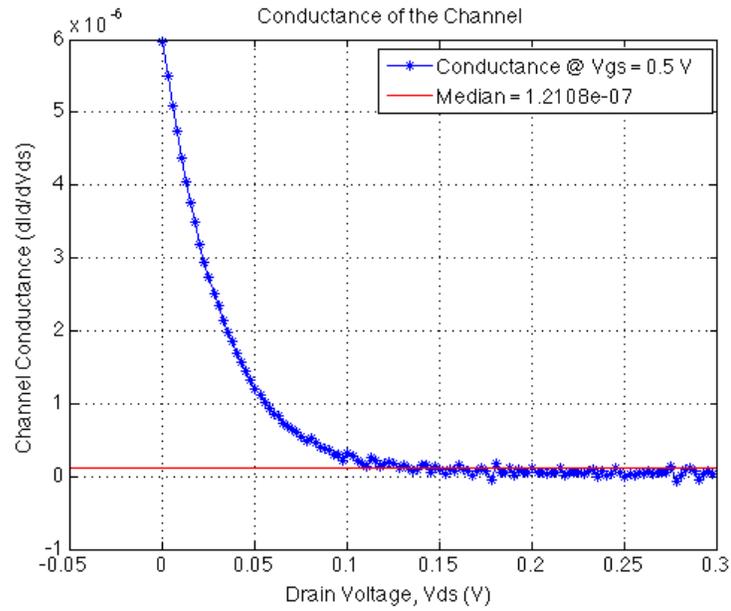


Figure 4.5.: T-5 Conductance, g_0 is calculated from the slope of the I-V curve from Figure 2.4.

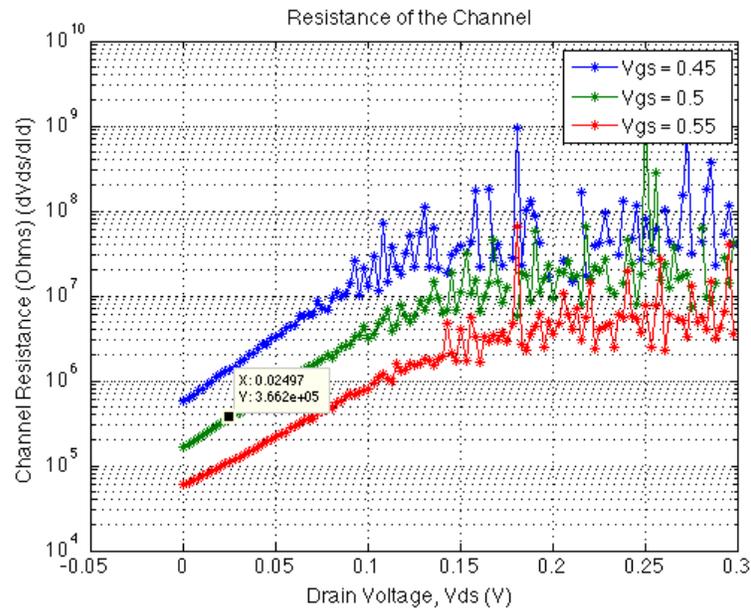


Figure 4.6.: T-5 Resistance, r_0 vs V_{DS} . Point of interest is noted.

4. Results & Discussion

Using these values, the thermal noise of the channel as related to Johnson noise is calculated through Equation (1) as follows:

$$\begin{aligned} i_{D,Therm} &= \sqrt{4k_B T g_0 \Delta f} \\ &= \sqrt{(4)(1.38E-23)(300)(2.731E-6)\left(\frac{.01}{60}\right)} \\ &= 17 \text{ pA} \end{aligned}$$

where the $\Delta f = \frac{.01}{60} = 1.7 * 10^{-4}$ Hz comes from the NPLC integration setting on the SMU in terms of power line cycles, in other words, the integration aperture for each measurement is a function of the number of power line cycles, in this case, NPLC = 0.01 of 60 Hz.

Next, in order to verify this calculation, a resistor of similar value to the MOSFET channel, 225 k Ω , is wired in place of the drain connections inside of the enclosure. This is shown in Figure 4.7.

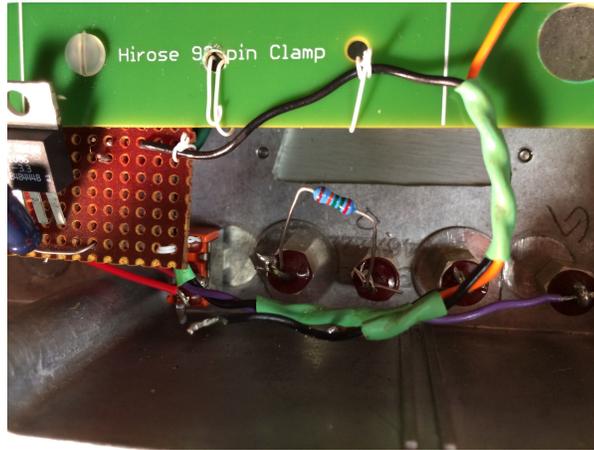


Figure 4.7.: Image of resistor wired on source-drain terminals inside enclosure.

By comparing the MOSFET with an actual resistor, uncertainties in the measurement

4. Results & Discussion

system and connections can be eliminated. This also acts as a sanity check for comparing calculated with measured values.

A current measurement is performed at the specified bias point of $V_{GS} = 0.5\text{ V}$, $V_{DS} = 25\text{ mV}$, with the MOSFET (Figure 4.8) and with the resistor in its place (Figure 4.9). The measurements performed are exactly the same, and occur in the same conditions, with the enclosure sealed and the silicon window in place. Ten measurement sets are taken in each case.

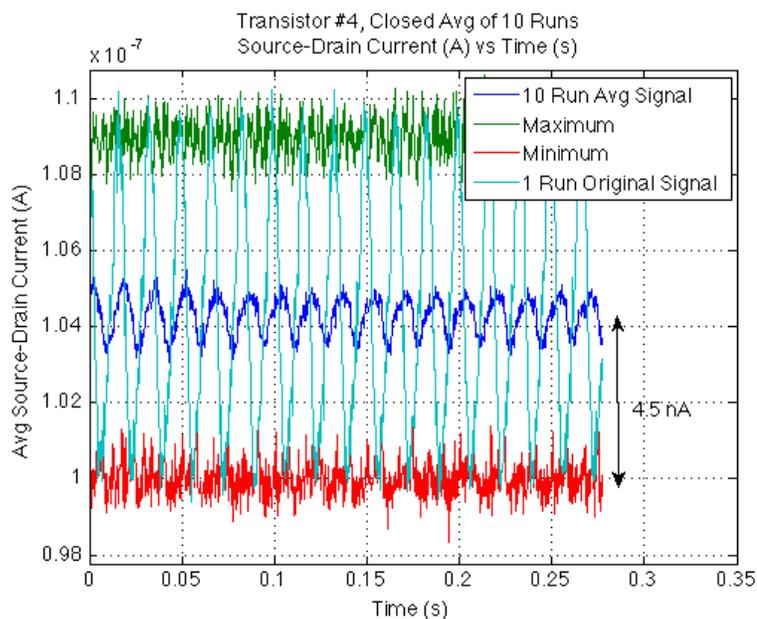


Figure 4.8.: MOSFET measurements for thermal noise. Current vs time.

4. Results & Discussion

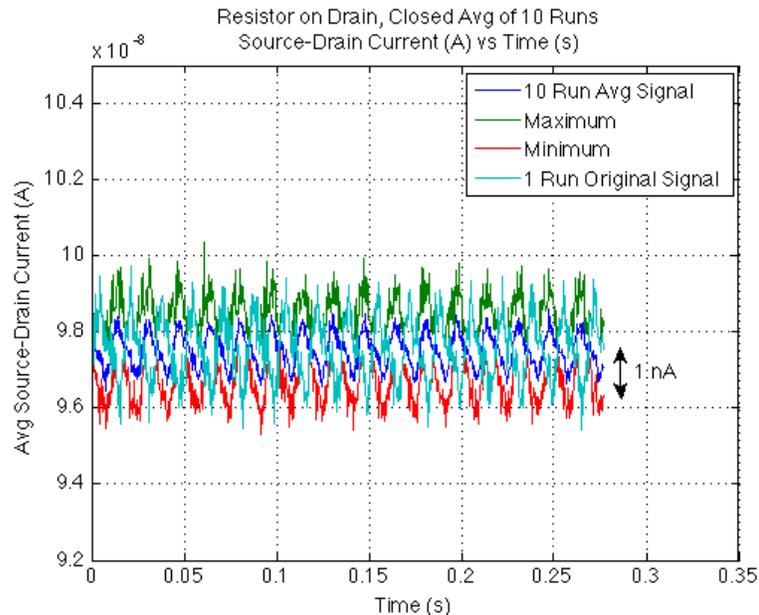


Figure 4.9.: 225 k Ω resistor measurements for thermal noise. Current vs time.

The cyan is a single measurement series, where the royal blue is an average of the ten runs together. Approximately a \sqrt{N} improvement is seen due to the signal averaging. The maximum and minimum are shown from all of the 10 runs at each measurement point.

Since the 60 Hz noise is so dominant in the signal (Figures 4.8 and 4.10), we remove it from the signal via fourier math (i.e. frequency analysis), and calculate the RMS noise of the signal as shown in Figure 4.11.

4. Results & Discussion

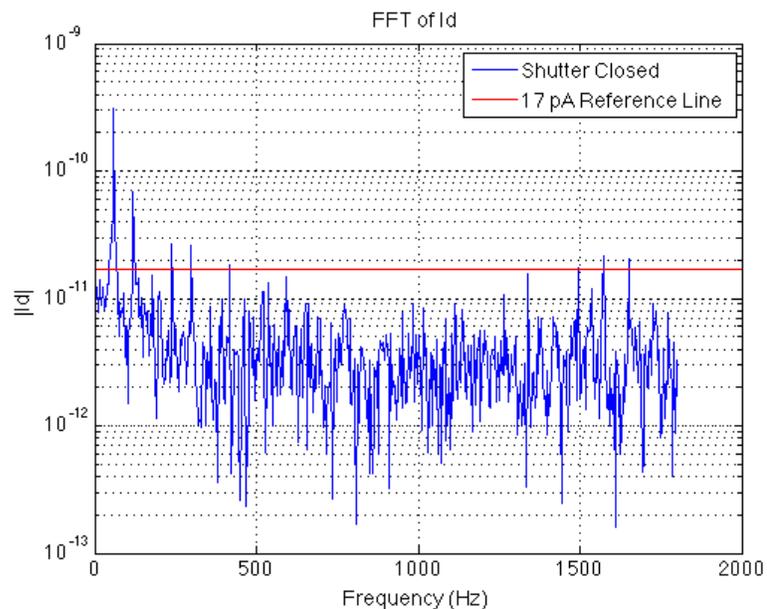


Figure 4.10.: Power spectrum of MOSFET signal from Figure 4.8.

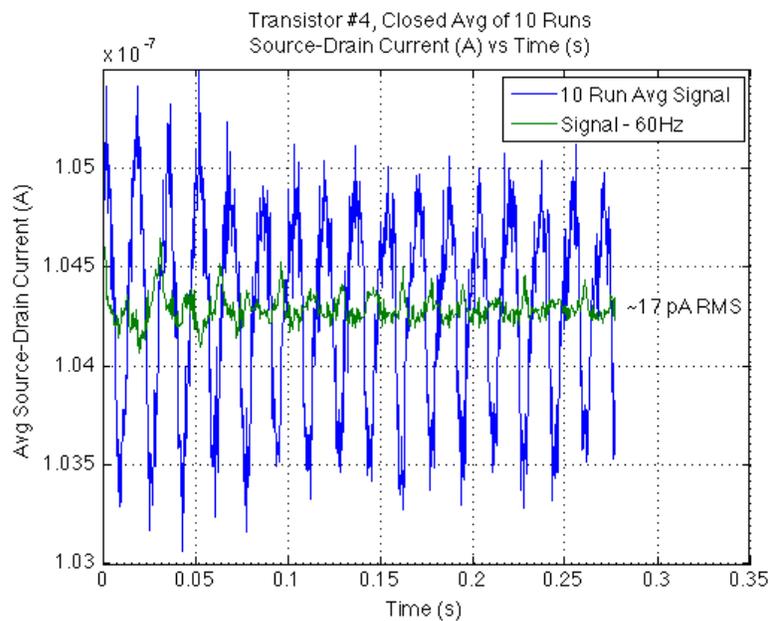


Figure 4.11.: Signal from Figure 4.8 w/o 60 Hz. Measurements agree with with calculated 17 pA Johnson noise.

The RMS noise measured is 17 pA, which matches the original calculation of Johnson noise. This is affirming since the most of the noise in the signal is a result of thermal noise and not other large factors other than the 60 Hz power noise. The 60 Hz power noise is extremely difficult to eliminate since all of the measurement equipment runs at that frequency. Efforts to reduce this include using a uninterrupted power supply (UPS) along with many different grounding schemes and measures. These procedures are discussed further in Appendix A.1.1.

4.1.4. Random Telegraph Signal Noise

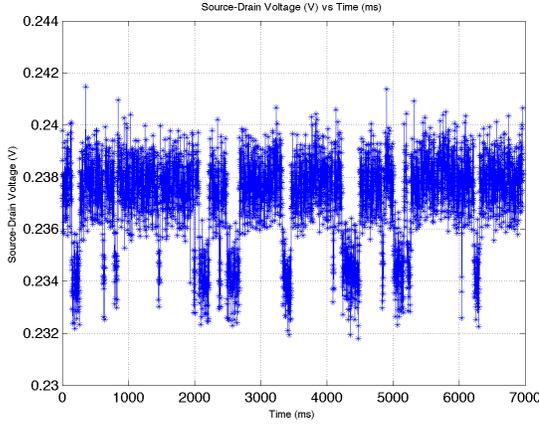
Random telegraph signal (RTS) noise, commonly referred to as popcorn noise or burst noise, is characterized by random fluctuations in time of voltage (or current) to two or more discrete levels [46, 47]. The voltage versus time waveform has the appearance of square waves with random lengths of time. In MOSFETs, this is typically caused by metal contaminants which precipitate during the foundry processing into the channel region beneath the gate of the MOSFET[47, 48].

Certain chips that were tested exhibited these random DC offset shifts due to manufacturing defects in the silicon substrate. Some chips/pixels exhibit a more severe case of RTS noise than others, and some had none at all. RTS noise adds a tremendous amount of low frequency noise into signal measurements, and most of the time pixels that exhibit this are deemed inoperable or bad. The T-1 through T-5 bowtie MOSFETs characterized are on a chip that did not exhibit a significant amount of RTS noise, however it is important to note when producing an imaging array since some pixels will have it, and it is difficult to mitigate.

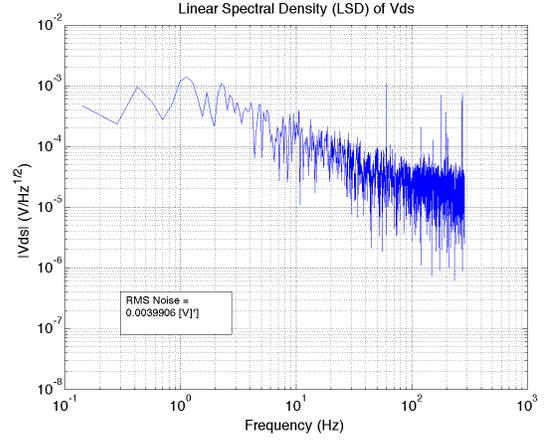
Figure 4.12 shows examples of RTS noise. The data are taken while ‘sitting’ on a biased pixel and reading the drain voltage. No THz energy or other radiation is present.

4. Results & Discussion

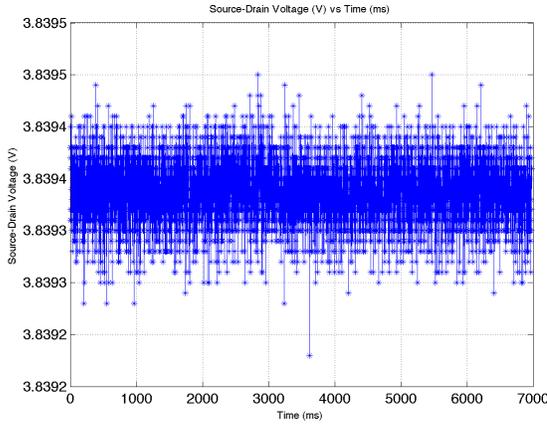
Figures 4.12(a) and 4.12(b) show a voltage time signal and its corresponding Linear Spectral Density (LSD) in frequency space, while Figures 4.12(c) and 4.12(d) show a normal pixel.



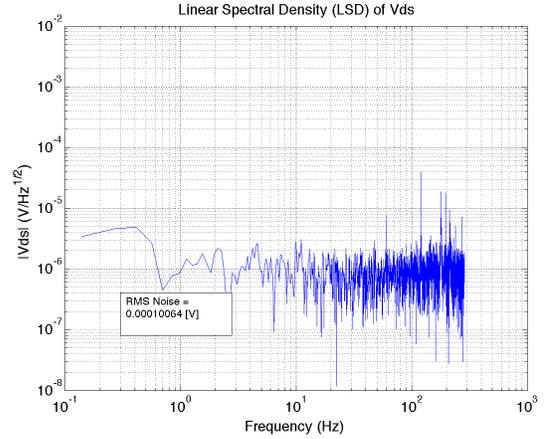
(a) RTS Pixel: V_{DS} vs. Time



(b) RTS Pixel: LSD $\frac{V}{\sqrt{Hz}}$



(c) Normal Pixel: V_{DS} vs. Time



(d) Normal Pixel: LSD $\frac{V}{\sqrt{Hz}}$

Figure 4.12.: RTS Noise Example. (a) and (c) show the voltage time signal and (b) and (d) show the linear spectral density for a normal vs a bad RTS pixel. Notice in (a) the random DC offset shift around two different voltage means. This is an example of RTS noise, whereas in (c), these shifts are not present. An increase of ≈ 4 mV RMS of noise is added to the signal due to RTS.

4.2. Gunn Diode Source Characterization

The Gunn diode was originally manufactured by Virginia Diodes [49] back in the 1990s for a university research project. It has a 95 GHz source with a doubling multiplier to 188 GHz. The source is powered using a single 10V source which connects via a BNC. The original power specification for the 188 GHz configuration is 56 mW. Figure 4.13 shows an image of the Gunn diode.

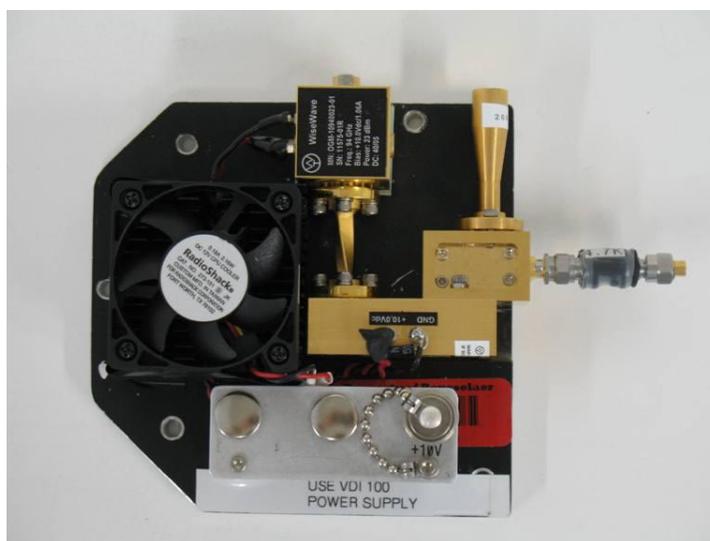


Figure 4.13.: VDI 188 GHz Gunn diode.

4.2.1. Power Measurements

The power of the 188 GHz Gunn diode was measured using two different detectors and compared with original specifications to arrive at an estimated total power of 55 mW. A Pacific Millimeter GaAs Diode (Model: GD) with a conical horn antenna, and a Gentec pyroelectric detector (Model: QS-5 THz-BL) were used for measurements and a Velmex Bi-Slide motorized stage was used for changing the distance between the source and detector. Neither of these detectors are calibrated for absolute response at 188 GHz

(relative response), but using the calibration curves and procedures recommended by the manufacturers, an integrated source power of 55 mW was derived. This is in agreement with the Virginia Diodes specification of 56 mW.

4.2.1.1. GaAs Diode Measurements

The Gunn diode was mounted on the Velmex stages and aligned with the GaAs diode detector on an optics table. A MATLAB script commands the stages to move away from the detector at specified increments, and the voltage is read into MATLAB via a TDS-200 digital oscilloscope. The result of these measurements is shown in Figure 4.14.

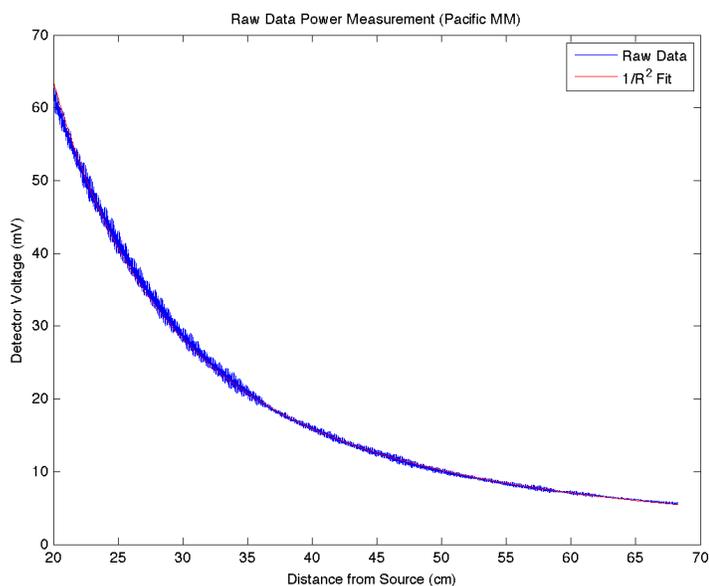


Figure 4.14.: Gunn diode power measurements using GaAs diode detector. The $1/R^2$ fit line follows the measurements precisely. The minor oscillations in the measured voltage are due to phasing reflections between the transmit and receive antennas.

The $1/R^2$ fit line follows the measurements precisely as expected. This allows us to use

4. Results & Discussion

the Friis transmission equation (see Equation 4.8) [50]. This was derived in 1945 by Bell Labs worker Harald Friis in order to calculate the ratio of power between transmit and receive antennas under ideal conditions. Assumptions include that the antennas are in the far field, are unobstructed, a single wavelength is present, and they are correctly aligned for polarization. Since we have a measured receive power, the equation can calculate the transmit power based on known system specifications. The horn antennas used for the transmit and receive sides have different sized circular apertures and therefore different gains. The aperture efficiency of the antennas are assumed to be the same, and have been reduced from the specifications. The following specifications are identified:

<u>Description</u>	<u>Symbol</u>	<u>Value</u>
Aperture efficiency	e_A	0.465
Diameter Transmit Antenna	d_t	8.33 mm
Diameter Receive Antenna	d_r	11 mm
Effective Area of Receive Antenna	$A_{r,eff}$	0.422 cm ²
Wavelength of Radiation	λ	1.5 mm
GaAs Detector Sensitivity @ -20 dBm	$S_{detector}$	400 mV/mW
GaAs Detector Measured Voltage	$V_{measured}$	- V
Power at Transmit Side	P_t	- mW
Power at Receive Side	P_r	- mW
Distance Between Antennas	r	- cm

The gain of the antennas can be calculated through:

$$G_t = e_A \left[\frac{\pi d_t}{\lambda} \right]^2 = 141.534 \quad (4.2)$$

$$= 21.5 \text{ dB} \quad (4.3)$$

$$G_r = e_A \left[\frac{\pi d_r}{\lambda} \right]^2 = 246.806 \quad (4.4)$$

$$= 23.9 \text{ dB} \quad (4.5)$$

The receive power calculated using the measured voltage from the GaAs detector for a given distance is found through the following:

$$P_r(r) = \frac{3200}{750} \left[\frac{V_{measured}(r)}{S_{detector}} \right] \quad (4.6)$$

The 3200/750 factor comes from the chart, Figure 4.15. The chart is for a 20 GHz detector, but the relationship translates to our 200 GHz detector as well. This multiplication factor considers temperature and signal level to adjust sensitivity. The manufacturer gives the sensitivity specification, $S_{detector}$ at an incident power level of -20 dBm so this must be translated to the appropriate curve on the chart.

4. Results & Discussion

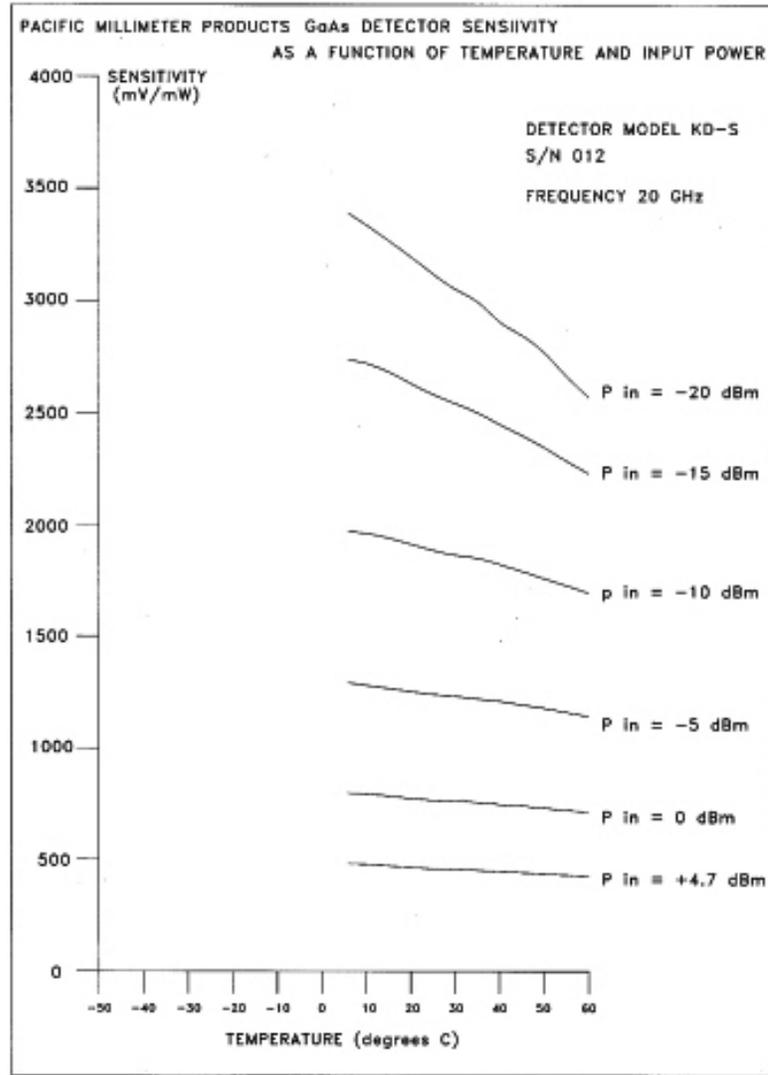


Figure 4.15.: Response Curves for 20 GHz Detector. The 200 GHz detector is approximately 8 times less sensitive than the 20 GHz detector the chart is spec'd for. This factor of 8 was provided by Pacific MM support.

The effective area of the receive antenna is calculated through:

$$A_{r,eff} = \frac{\lambda^2 G_r}{4\pi} = 0.442 \text{ cm}^2 \quad (4.7)$$

4. Results & Discussion

Lastly the total power applied to the transmitting antenna using the Friis equations is:

$$P_t(r) = \frac{P_r(r)}{G_t G_r \left[\frac{\lambda}{4\pi r} \right]^2} \quad (4.8)$$

$$P_t(45cm) = 55.385 \text{ mW} \quad (4.9)$$

Performing this calculation for all of the measurement points over the distance range results in Figure 4.16.

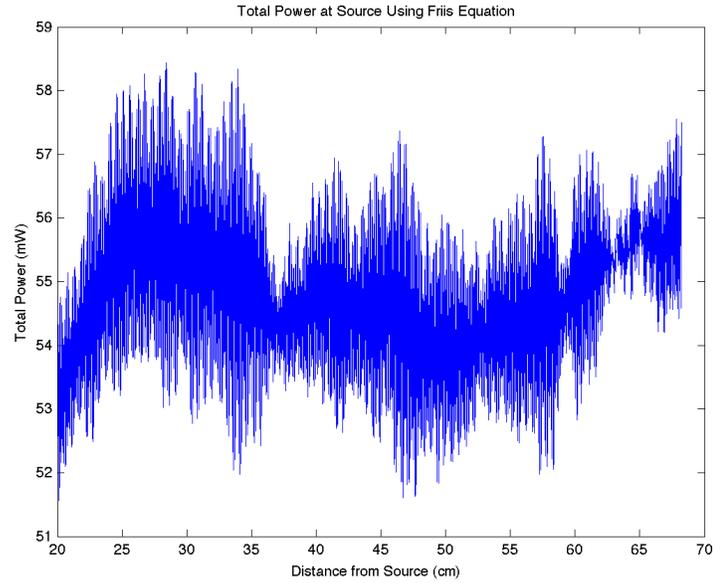


Figure 4.16.: Power calculated at transmit side (source) for each measurement along r . The average is ≈ 55 mW total integrated power. Sampling increments are 1 mm. The noise around the average is due to variations in the measured voltage output from the GaAs diode.

4.2.1.2. Pyroelectric Measurements

The same procedure for acquiring data with the GaAs diode detector is used for the pyroelectric detector. A Gentec EO QS5-BL detector and corresponding Gentec testing setup are used to collect these power measurements. This detector is very sensitive to all forms of thermal radiation and is much more difficult to acquire measurements. Very slight changes in thermal background, such as a person in the room conducting measurements, or temperature instability of the supporting electronics will produce a change in signal. For these reasons we manually collect several data points and extrapolate a $1/R^2$ fit since this relationship was confirmed through the previous method. The detector voltage and fit line for these measurements are shown in Figure 4.17.

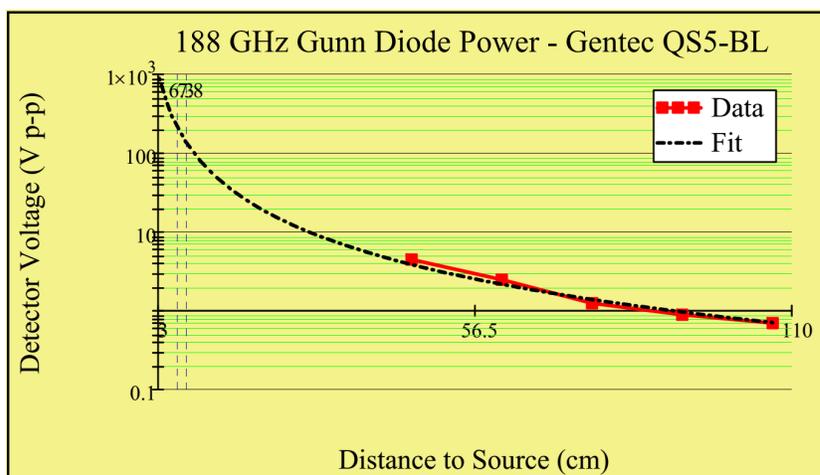


Figure 4.17.: Gunn diode power measurements using the pyroelectric detector. The data and $1/R^2$ fit line are shown.

For this method we use a different approach since the receiving detector does not have an antenna, but is a simple detector with a fixed area. First, the following parameters are defined:

4. Results & Discussion

<u>Description</u>	<u>Symbol</u>	<u>Value</u>
Wavelength of Radiation	λ	1.5 mm
Pyroelectric Responsivity	R_d	93.500 V/W
Area of Pyroelectric Detector	A_d	0.196 cm ²
Diameter of Source Aperture	d_s	8.33 mm
Pyroelectric Absorption Factor	α	0.25
Pyroelectric Measured Voltage	$V_{measured}$	– V
Total Power on the Detector	P_d	– mW
Total Power at the Source	P_s	– mW
Distance from Source	r	– cm

The power on the pyroelectric detector is calculated through:

$$Pd(r) = \frac{V_{measured}}{\alpha R_d} \quad (4.10)$$

where α is the absorption factor of the pyroelectric detector. This factor is advertised on the manufacturer's specifications sheet as an estimated value of 10% at the 188 GHz wavelength. Using this value for α did not produce reasonable results, and as this is not an absolute calibrated detector (partly because absolute calibrated equipment does not exist for many of the THz wavebands), the manufacturer was contacted to verify this factor. Although undocumented, the manufacturer confirmed that many other customers working in this wavelength region were seeing absorption at the same 25% level and that an $\alpha = 0.25$ is the appropriate value to use for power calculations.

The source is estimated as having a Gaussian profile following the three sigma rule. This states that $> 99.7\%$ of the power will lie within 3σ of the mean in a normal distribution. This agrees with the radiation profile of the source antenna as well which

is shown in Figure 4.18.

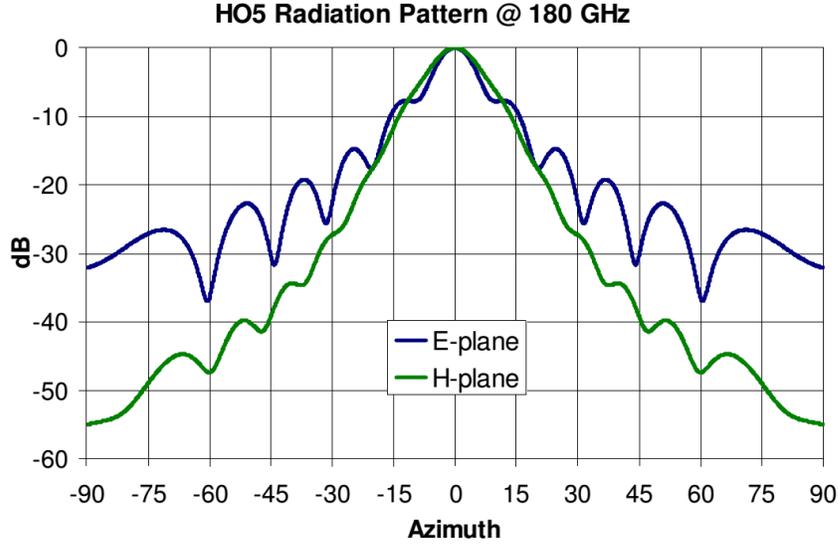


Figure 4.18.: Radiation pattern of the Gunn diode source antenna. Provided by the manufacturer Custom Microwave, Inc. [18].

Using 12 degrees as the angle of 3σ for the radiation profile, the standard deviation for a matched 2-D Gaussian profile at a distance $r = 45$ cm is $\sigma = 3.2$ cm. A second verification of this estimation can be calculated by estimating the transmit horn antenna as a circular aperture for the source, and therefore the first zero for diffraction is found by $1.22\lambda/d_s$. If this is estimated as the 3σ point, the standard deviation is easily found to be $\sigma = 3.3$ cm. Both of these methods for estimating the source energy of a Gaussian profile agree.

The volume of a Gaussian related to the distance from the source, r and the standard deviation, σ is then used to calculate the total power:

$$P_s(r) = 2\pi \frac{P_d(r)}{A_d} \sigma^2 \quad (4.11)$$

$$P_s(45 \text{ cm}) = 55.865 \text{ mW} \quad (4.12)$$

4. Results & Discussion

This calculation agrees well with the GaAs diode measurements as well, and therefore it is concluded with confidence that the source power is ≈ 55 mW. Because of the inaccuracies in both methods, the sensitivity relationship for the GaAs diode detector and the absorption factor for the pyroelectric detector, an uncertainty of $\pm 10\%$ is assigned to this value and corresponding responsivity and NEP calculations in the results section.

4.3. MOSFET Response

4.3.1. Gas Laser Results

Worth noting are the experiments conducted with a 100 mW class 1.63 THz gas laser at the UR. These did not produce any measurable results, likely in part due to instrumentation noise and/or low responsivity. This is expected however, due to the fact that the gate length, L , has a significant effect on response for a given radiation frequency. Tauk et. al. [19] studied this variation of gate length vs frequency at 0.7 THz and documented a significant drop in signal, as shown in Figure 4.19.

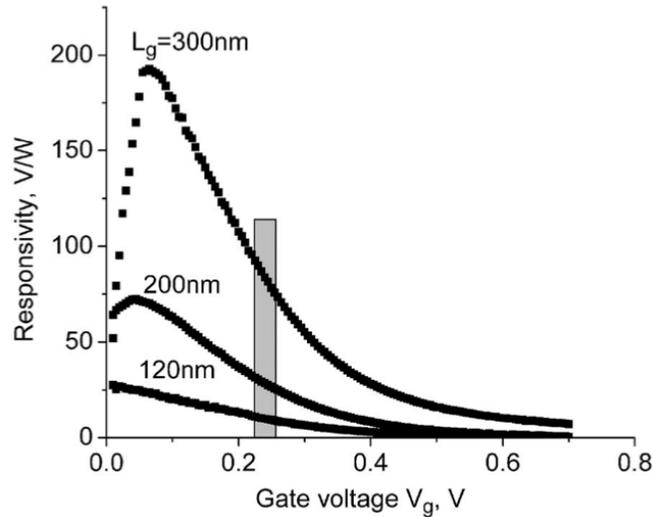


Figure 4.19.: Detection signal as a function of gate length at 0.7 THz Reprinted from [19]

Coupled with the fact that our chip has no gain stage, if signal would be present, it is unlikely that we would be able to measure it. For these tests, the detector was aligned with a parabolic gold coated mirror to focus the laser radiation. This setup is shown in Figure 4.20. Lock-in measurement techniques are also employed to boost SNR.

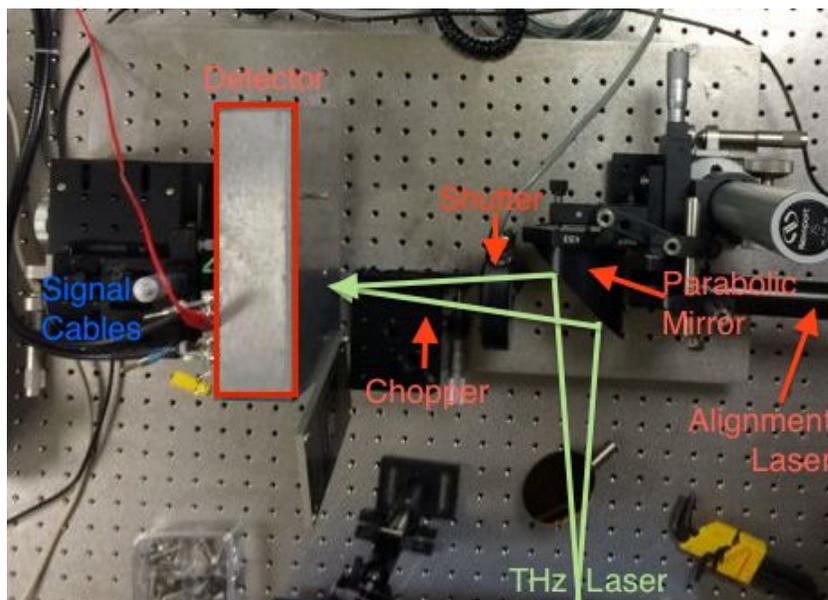


Figure 4.20.: Experiment Setup in the UR THz Laser Lab. A mechanical chopper wheel and lock-in measurement technique are used to boost SNR. A HeNe alignment laser is used to confirm algorithm operation since the silicon chip is sensitive to visible light.

4.3.2. Gunn Diode Results

4.3.2.1. Bowtie Antenna MOSFETs

The T-1 through T-5 MOSFETs are each characterized for THz response through test by finding the peak biasing regimes and then zeroing in on maximum signal. Figures 4.21-4.25 show the measurements for each MOSFET to find the peak response. These results were gathered using the procedures outlined in Section 3.4 with the shutter and enclosure. The Gunn diode is at a fixed distance of 6.3 cm for these tests. Response is ‘peaked’ through manual alignment of the MOSFET and source. As response characteristics are realized, the response region is concentrated on. Several ranges of bias currents and voltages are shown to help provide a perspective of how the response shifts with bias changes. Along with the response signal, the inverse of signal-to-noise (SNR) or (NSR)

4. Results & Discussion

is shown to determine how this correlates with peak response. The NSR calculated as the standard deviation divided by the signal mean will produce a sharp peak where the best ratio lies.

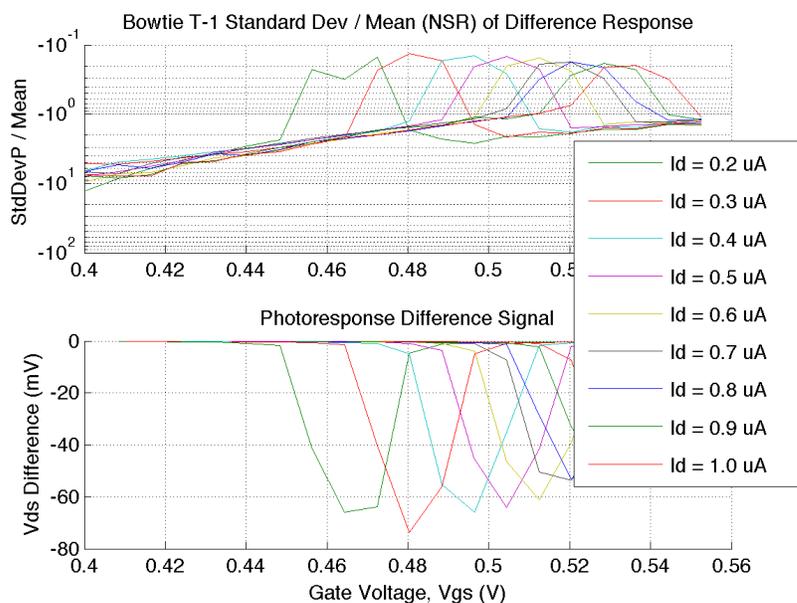


Figure 4.21.: Bowtie T-1 Photoresponse & NSR.

4. Results & Discussion

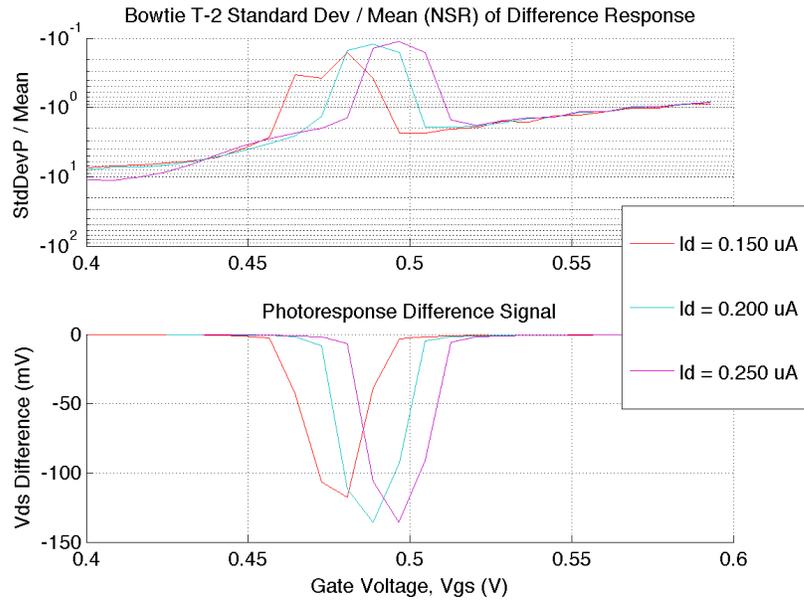


Figure 4.22.: Bowtie T-2 Photoresponse & NSR.

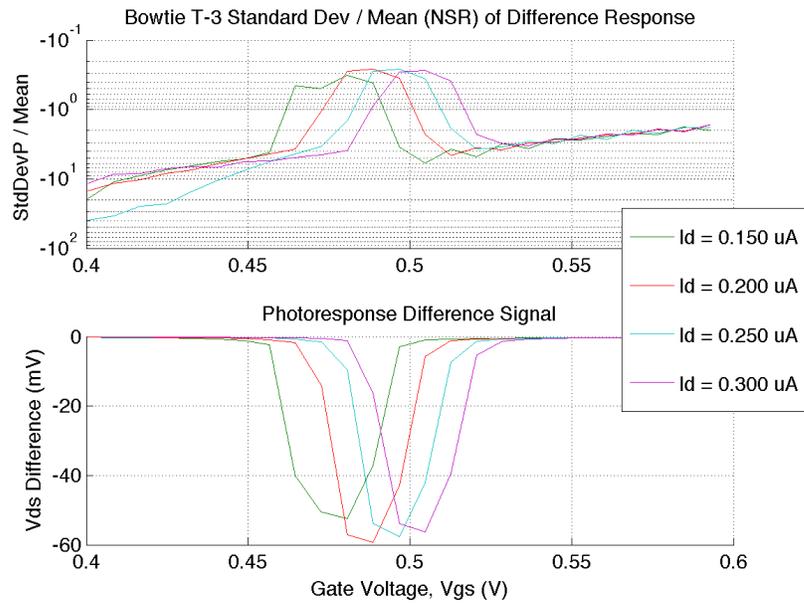


Figure 4.23.: Bowtie T-3 Photoresponse & NSR.

4. Results & Discussion

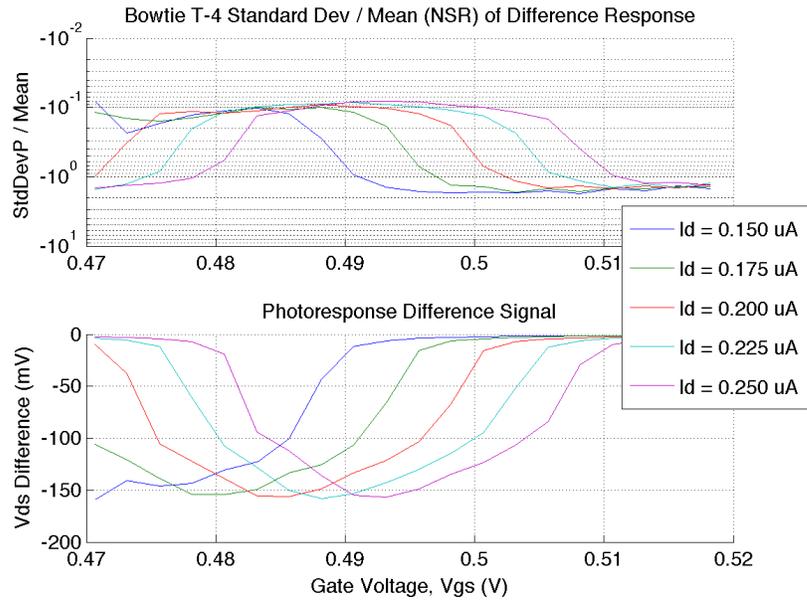


Figure 4.24.: Bowtie T-4 Photoresponse & NSR.

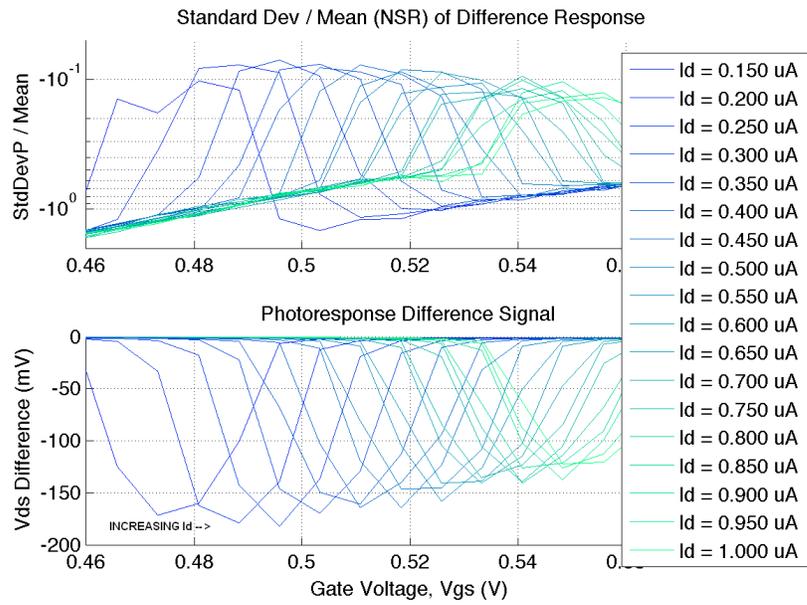


Figure 4.25.: Bowtie T-5 Photoresponse & NSR.

4. Results & Discussion

Notice how the peak response gate voltage shifts with respect to increasing i_D . Also seen is that the peak of the NSR curve coincides with the peak of the response voltage curve for a given i_D . The T-5 transistor which has the most source region extension has the best response out of the five MOSFETs tested. Figure 4.26 shows the peak response of all five transistors tested.

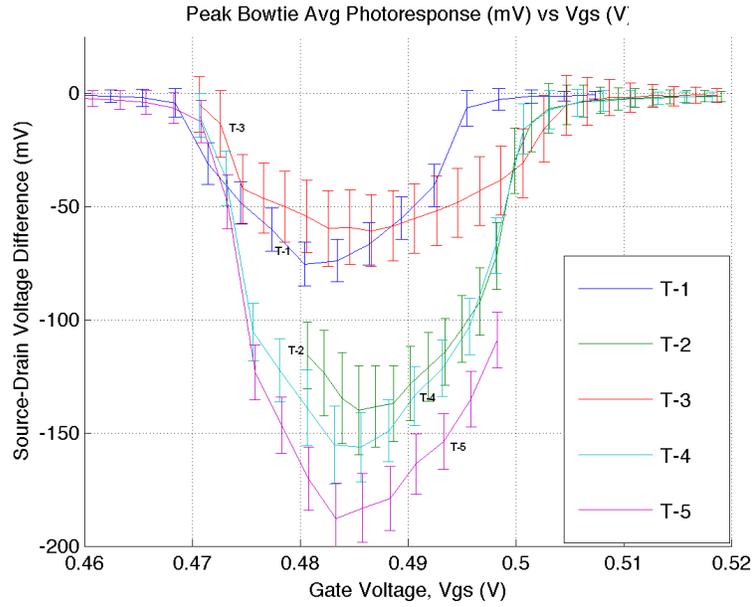


Figure 4.26.: Photoresponse ΔU of all five MOSFETs at peak drain bias. Errorbars represent the standard deviation of signal measurement.

The measurement results indicate that the source region extension helps increase the responsivity without much noise penalty. T-3 is the exception, which could be explained by a impedance mismatch, substrate defects, or testing error. Enhancements in the test setup to allow less invasive switching of the transistor under test is needed to fully characterize the T-3 exception. It is believed that this responsivity improvement due to source region extension comes from the reduction of the effective voltage U_0 seen by the gate-channel interface, resulting in additional signal rectification. However, further

detailed analysis and modeling are required to fully understand this mechanism. These data are promising though, as efforts to improve pixel design as ongoing.

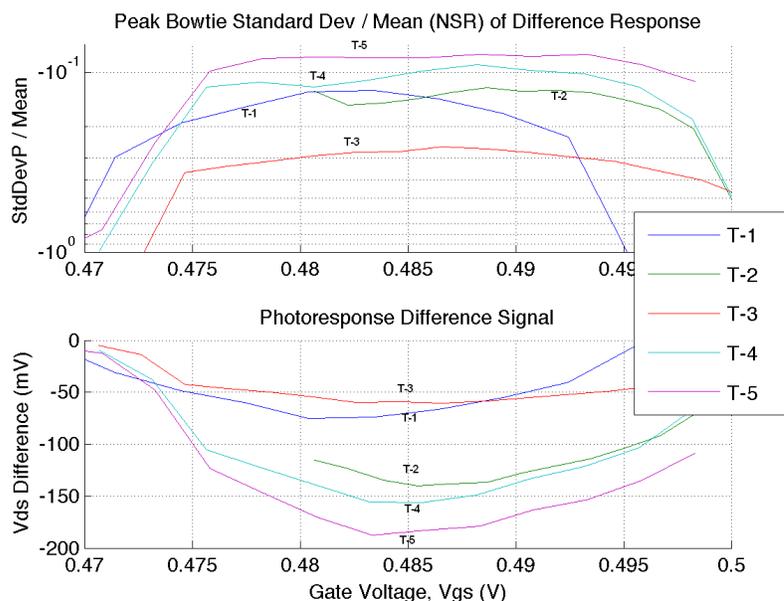


Figure 4.27.: Blowup of Figure 4.26 near peak response with NSR.

4.3.2.2. Spiral Antenna MOSFETs

The spiral antenna coupled MOSFETs did not produce any measurable results. There are several possible reasons for the lack of response. One is that the gap between the arms of $5\ \mu\text{m}$ is significantly smaller than the wavelength of the radiation at 1.5 mm. Thus, to the radiation this antenna design looks like a solid sheet of metal preventing any signal coupling into the MOSFET. If the antenna electric field is the primary method of coupling, this would also explain no signal as antenna arms are much further apart, reducing the electric field strength near the source region. Finally, these antennas are not designed for the 188 GHz radiation so response to begin with is somewhat unexpected.

The next chip design for this project team will be able to determine the method of signal coupling to help further explain this lack of response.

4.3.3. Response & Orientation

Another variable studied is the orientation of the pixel with respect to the polarization of the incoming radiation. All the previous results present the case where the antenna, gate channel, and radiation polarization are all aligned in the same horizontal direction as shown in Figure 4.28.

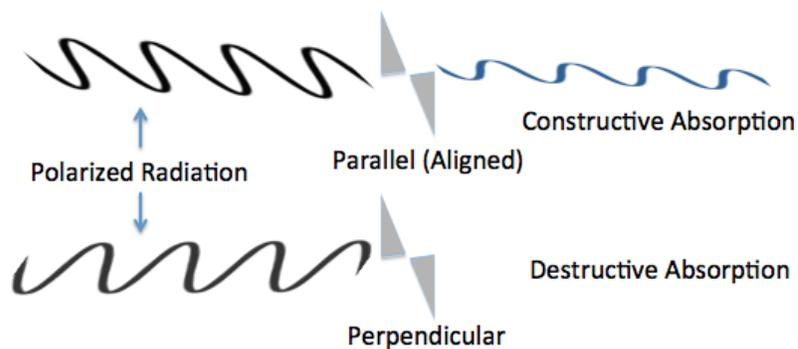


Figure 4.28.: Diagram of polarized radiation. (Top) In the case where the electric field of the antenna is aligned with the polarization of the source, the radiation will be coupled. (Bottom) If the electric field of the antenna is perpendicular to the polarization of the source, than little to no radiation is coupled.

It was not expected that the orientation would have such a drastic effect on the response, since the antenna on the pixels are not optimized for 188 GHz radiation. Nevertheless, Figure 4.29 shows that the orientation has a significant effect on the radiation coupling into the MOSFET channel.

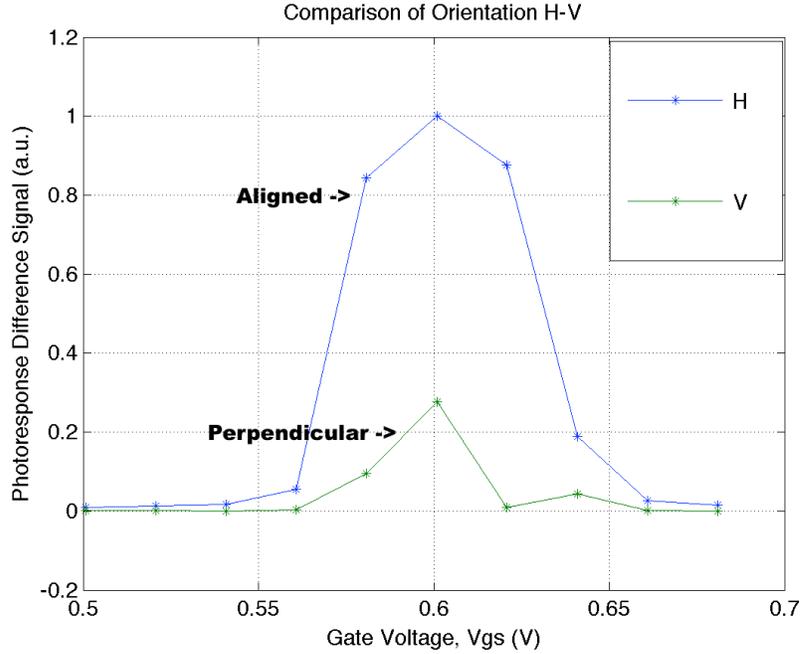


Figure 4.29.: Bowtie T-1 Response. Horizontal vs Vertical Orientation. The polarization of the source is co-aligned with the horizontal case. The bowtie antenna and the MOSFET gate (perpendicular to channel current flow) are in the same orientation.

4.3.4. Responsivity & NEP

In order to calculate the responsivity, the power of the source per unit area as well as the definition of the pixel area must be determined. The power of the 188 GHz Gunn diode was measured using two different types detectors and compared with original specifications to arrive at an estimated total power of 55 mW. This is used for the responsivity calculations and further details are explained in Section 4.2. The primary uncertainty is the definition of the pixel's effective aperture (area). The most precise method is through measurement and deconvolution as explained in [20]. This method calculates the effective aperture through a quasi-optical system estimation, as shown in Figure 4.30.

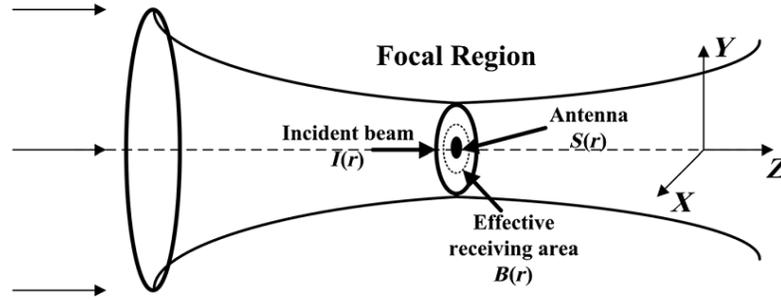


Figure 4.30.: Quasi-optical system interpretation. The focal region describes the plane at which incident radiation interacts with the antenna. Reprinted from [20].

Through this estimation the spatial response of the antenna coupled detector $B(r)$ is [20]:

$$S(r) = \int_{-\infty}^{\infty} B(r)I(r)\delta r \quad (4.13)$$

where $S(r)$ represents the detected signal and $I(r)$ is the 2-D field distribution of the incident Gaussian beam. The measured response signal from this detector can also be considered as the convolution between $B(r)$ and $I(r)$ [20]:

$$S(x, y) = B(x, y) * I(x, y) \quad (4.14)$$

$$= \iint_{-\infty}^{\infty} B(x', y')I(x - x', y - y')\delta x'\delta y' \quad (4.15)$$

If $S(x, y)$ and $I(x, y)$ are known, the effective aperture can be found through deconvolution. $I(x, y)$ can be found through measurements using a knife-edge technique and proper optical setup, and $S(x, y)$ is the output signal from the detector system over a 2-D scan of the radiation source. The spatial response $B(x, y)$ can then be found using

a deconvolution technique such as the Richardson-Lucy algorithm [20]:

$$B^{i+1}(x, y) = B^i(x, y) \frac{\iint_{-\infty}^{\infty} \frac{S(x', y')}{S^i(x', y')} I(x - x', y - y') \delta x' \delta y'}{\iint_{-\infty}^{\infty} I(x', y') \delta x' \delta y'} \quad (4.16)$$

where B^i is the detector response and i is the current step of the calculation. The resulting x-y projection of $B(x, y)$ is the effective aperture of the pixel. Figure 4.31 shows an example result for an NbN microbolometer pixel. For this example the effective aperture is 7.4 times larger than the physical size of the detector.

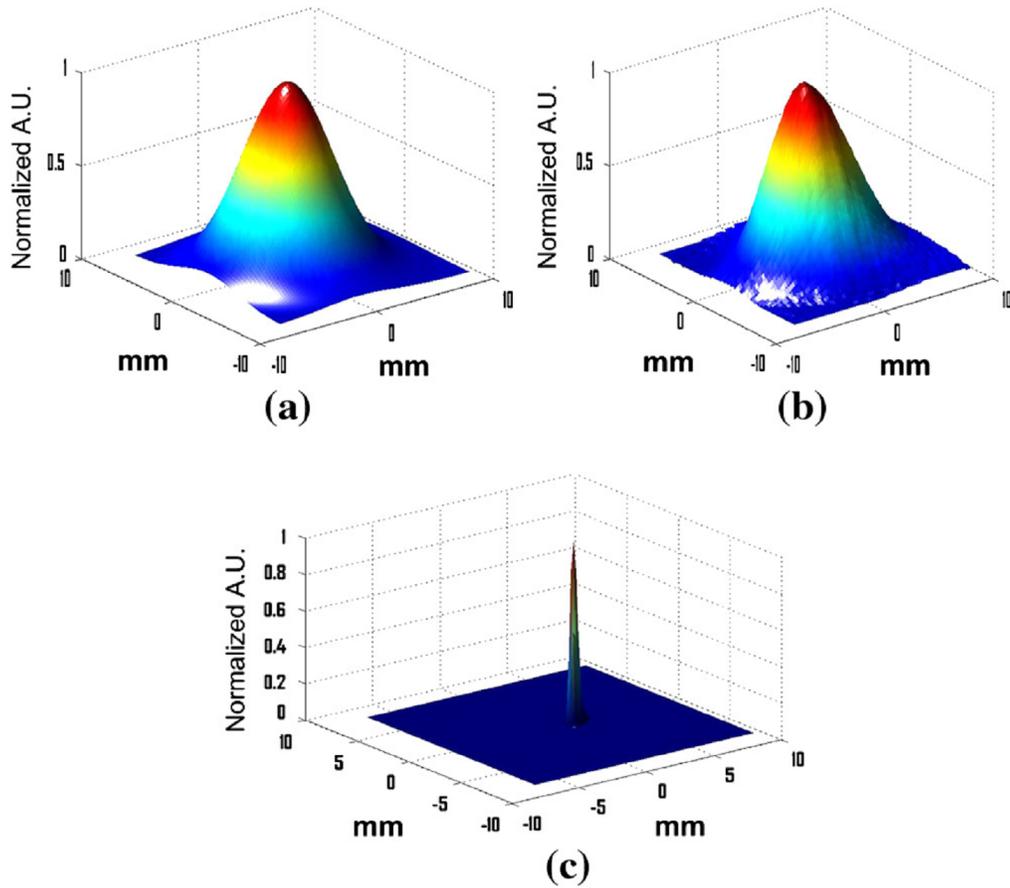


Figure 4.31.: Example deconvolution result for NbN microbolometer pixel. (a) The intensity distribution of the incident beam, $I(x, y)$. (b) The 2-D measured scan $S(x, y)$ using the NbN pixel. (c) Deconvoluted spatial response of the pixel $B(x, y)$. The x-y projection of this response is the effective aperture (area) of the pixel which is 0.7 mm^2 , 7.4 times larger than the physical size of the detector (not including the dipole planar antenna). Reprinted from [20].

A much simpler albeit inaccurate method for relative responsivity numbers is to use a predetermined physical area for the pixel aperture. This was chosen in part for direct comparison with previous published results [17]. Here the area of the pixel, A_{pix} , is defined as the square active region surrounding the antenna and pixel (100 μm x 100 μm), as in Figure 4.32.

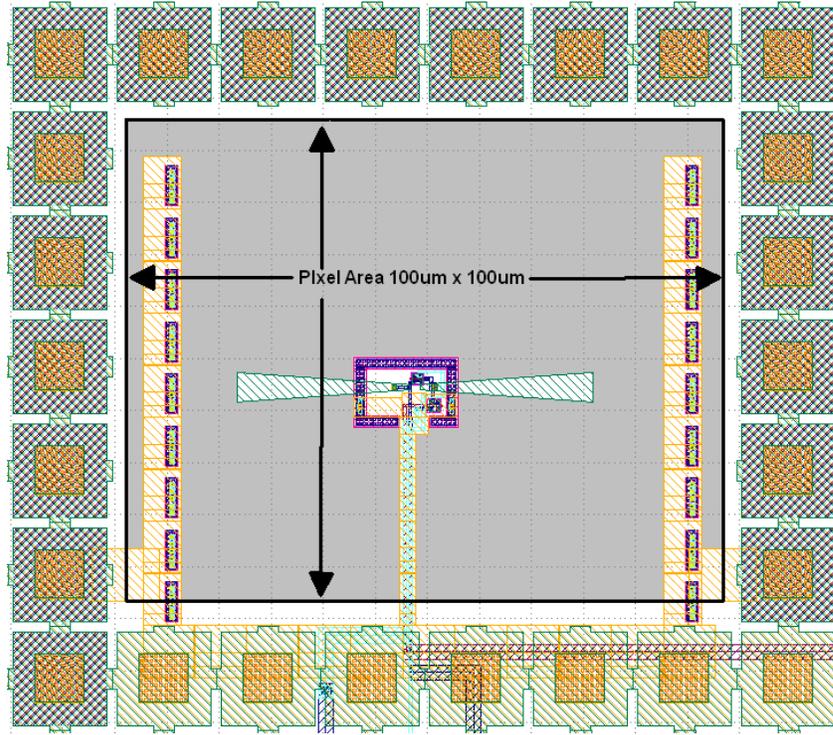


Figure 4.32.: Pixel Area Definition. For comparison with other's results, a 100 μm x 100 μm area is defined.

At a measurement distance of 6.3 mm, the power per area of the source, $P_{density} = 44.5$ mW cm^{-2} . The responsivity of the test MOSFET is then calculated as a function of the measured signal voltage, V_{pix} through:

$$R_v = \frac{V_{pix}}{P_{density} * A_{pix}} \left[\frac{V}{W} \right] \quad (4.17)$$

4. Results & Discussion

The noise equivalent power (NEP) based on a Johnson noise estimation is determined through:

$$NEP = \frac{\sqrt{4k_b T R_{ds}}}{R_v} \left[\frac{pW}{\sqrt{Hz}} \right] \quad (4.18)$$

where $k_B = 1.38 * 10^{-23} \left[\frac{J}{K} \right]$ is the Boltzmann constant, $T = 300 \text{ [K]}$ is the temperature, and $R_{ds} = 10 \text{ [M}\Omega\text{]}$ is the resistance of the drain-source at the detection bias point. The resulting responsivity and NEP for the test MOSFETs is shown in Figures 4.33 and 4.34.

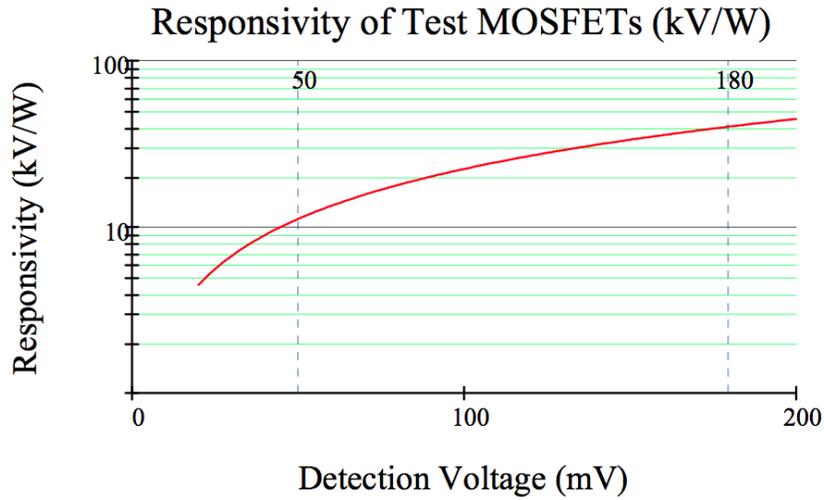


Figure 4.33.: The responsivity as a function of detection voltage. The markers represent the range of detection voltages from 50 mV (T-1) to 180 mV (T-5).

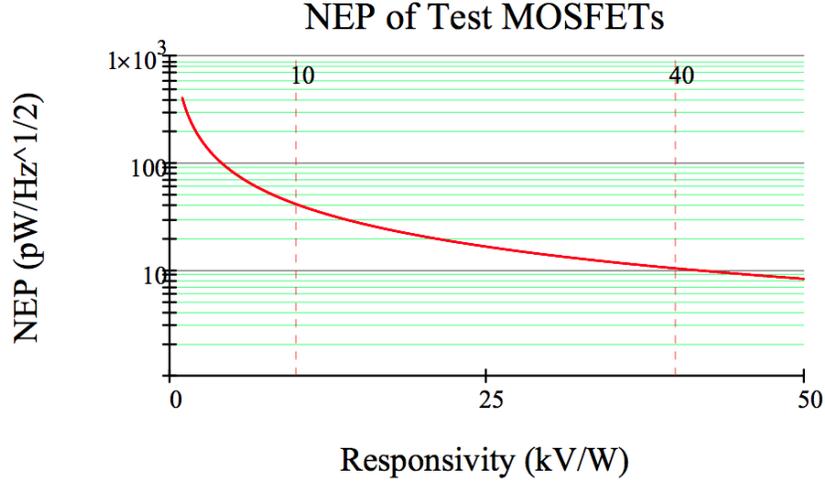


Figure 4.34.: NEP as a function of the responsivity. The markers represent the range of responsivity of the test MOSFETs from $\text{pW } \sqrt{\text{Hz}}^{-1}$ (T-1) to $\text{pW } \sqrt{\text{Hz}}^{-1}$ (T-5).

The T-5 transistor which has the highest response voltage of 180 mV out of the five test transistors translates to a responsivity of 40 kV W^{-1} and NEP of $10 \text{ pW } \sqrt{\text{Hz}}^{-1}$. Because of uncertainty in the absolute calibration of the source power these numbers are estimated to be within $\pm 10\%$ (see Section 4.2.1.2) of the calculated value.

The response of the T-1 transistor agrees well with other research [15, 17, 40, 51] with a responsivity of 10 kV/W . Specifically, Schuster et. al. [17] achieve a maximum responsivity of 5 kV/W out of a similar transistor with no gain stage. The active area was larger by a factor of four at $210 \text{ um} \times 210 \text{ um}$ and a dissimilar antenna describes the factor of two difference.

5. Conclusion

In order to realize the use of MOSFETs as a THz detector for FPAs, this thesis has presented the background, testing efforts, and evaluated results for five test MOSFETs. A thorough discussion of the radiation properties and state-of-the-art applications is given, along with a review of the theory behind plasmonic response in silicon MOSFET technologies. A test chip designed for direct broadband response is described and tested at 0.2 THz and an increase in response due to the implementation of an extended source region was found.

A significant amount of effort was put in to the development of the testing setup. The implementation of serial communication between the Keithley SMU and MATLAB, along with the resulting programming scheme provided a convenient data acquisition system. Adding the shutter via digital control through the SMU, and a similar coding effort with the motorized stages was important as well. All of these components allowed for many data points to be gathered for a given test, reducing errors and allowing for further statistical analysis of the results.

Results produced agree with theory and results published in literature [15, 17, 40, 51]. It was shown that the responsivity of the MOSFETs in the non-resonant regime was strongly affected by the biasing parameters, particularly the drain current, i_D . Adding a drain bias, when compared with zero bias, greatly increases the MOSFET response in the near-threshold region. An ideal bias is found empirically, as a change in bias current shifts the maximum voltage response with respect to gate voltage. The increase in response

falls off after the ideal drain bias, as asymmetry in the channel region is maximized and additional current flow starts to interfere with response. The best response of the five MOSFETs tested was T-5, which provided a maximum 180 mV of signal at a drain bias current, $i_D = 200$ nA. This MOSFET had the highest source region extension (2 μm) which improved response. It seems that this responsivity improvement due to an increase of L_s comes from the reduction of the effective voltage U_0 seen by the gate-channel interface. This additional signal rectification and results in a responsivity of 40 kV/W and NEP of $10 \text{ pW }^{-1}\sqrt{\text{Hz}}$.

5.1. Future Work

Efforts to characterize this increased response further will continue under this CEIS research partnership. The results presented were conducted with a single source frequency and type. Ideally a range of source frequencies would be available in order to fully understand the mechanism(s) of radiation coupling. Lack of an absolute calibrated source present some uncertainty in our measurements ($\pm 10\%$). Moving to a newer tunable type source and/or a calibrated detector will reduce this uncertainty. Determining whether the response of this novel design is linear or exponential with respect to source power is also an important confirmation. It is possible that response effects differing from the current theory are present, and need to be further understood. A novel theory of thermionic response in the source region, modified by the source extension region is being discussed.

A subsequent chip design is currently underway with many new test MOSFET designs and a 7 x 7 array based on the T-5 results to help explain these questions (see Figure 5.1).

5. Conclusion

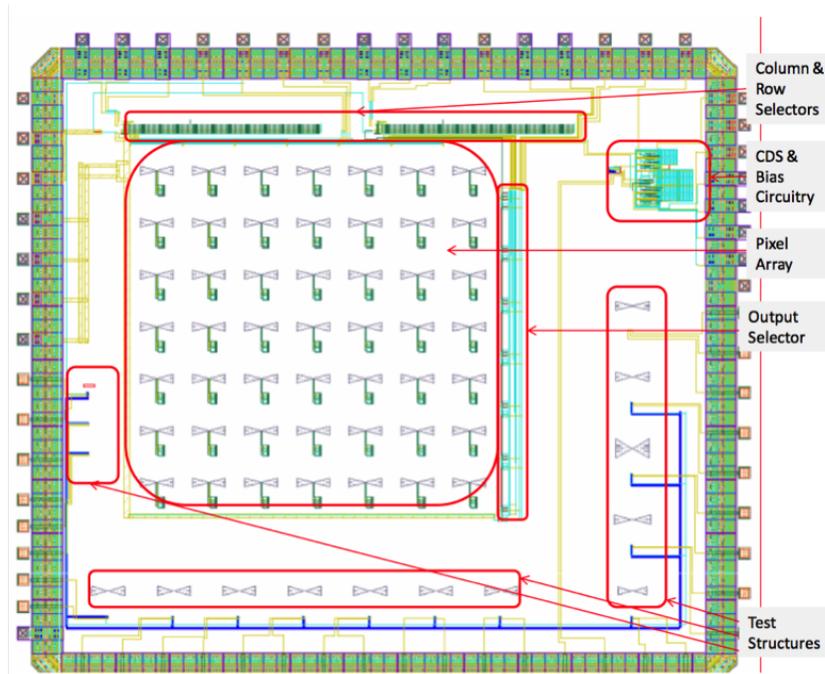


Figure 5.1.: Next-generation test chip design. Includes a 7x7 pixel array based on the T-5 test MOSFET which includes on-chip CDS, current biasing, and gain. 15 new test MOSFETs are also included to help parametrize future pixel design.

The current test setup does not have provisions for implementing a cooling mechanism, which would allow for testing the response as a function of temperature as well. A custom enclosure is being designed along with the new chip with provisions for a thermo-electric cooler (TEC) which will allow for temperature control within one degree Celsius. Results from this new chip will be able to determine the coupling effects of the current antenna, a new optimized antenna based on updated modeling, and also some additional source region modifications. Implementation of an on chip amplifier, current biasing circuit, and correlated double sampling (CDS) within the array will greatly increase SNR and provide an imaging capable package.

A. Test Equipment Description

A.1. Keithley 2602A Source Measurement Unit

The Keithley 2602A is a series 2600A system source meter instrument provides two channels designed for precision, DC, pulse and low frequency AC source measure I-V testing [52]. Specifications, protocols, commanding language and wiring provisions can be found in the following list of useful documents provided by Keithley:

<u>Document Title</u>	<u>Document Number</u>
Series 2600 Specifications	No. 2594-0605
Series 2600A Reference Manual	No. 2600AS-901-01 Rev. E
Series 2600 User's Manual	No. 2600S-900-01 Rev. A
Model 2600-TRIAx SMU Connector	No. PA-916 Rev. A
Series 2600A Semiconductor Device Test Applications Guide	No. 2911

A.1.1. Noise Considerations

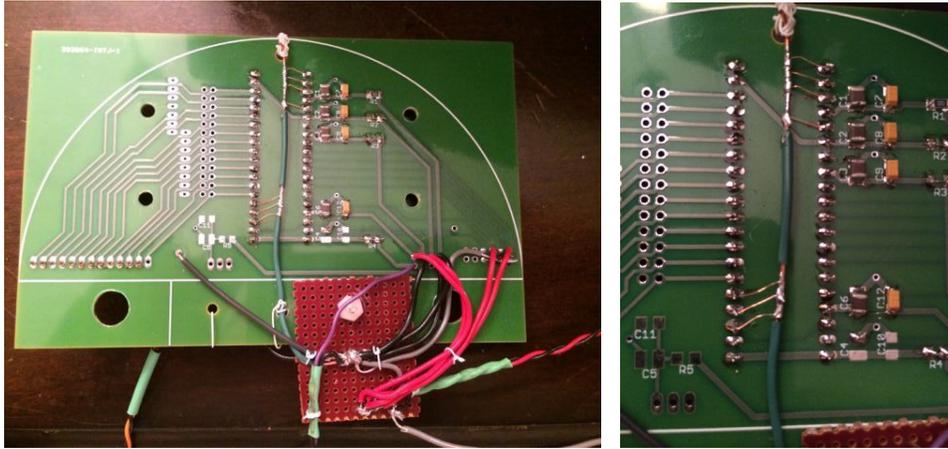
The use of this type of instrument inevitably leads to the use of external wiring, which introduces all sorts of issues with noise. Many of the noise concerns come from grounding issues, but RF coupling into the wires or device enclosure, or coupling from other electronics can also present many challenges. Some of the items that were realized to effect the noise of measurements include:

- Wall power noise (used battery backup)
- Ground loops from different equipment power sources
- Proximity of other test equipment to SMU
- Chopper wheel noise from digital I/O into SMU
- Coupling into measurement cables
- Issues with circuit settling on first measurement in a series
- Board/Enclosure wiring configuration

The testing setup and enclosure were completely rewired with twinax cable along with a new grounding scheme to attempt to minimize noise. The re-wiring of one of the fanout boards is shown in Figure A.1. Efforts to reduce noise and ground loops are the primary reasoning for the rewire. The power and ground connections were redone, and a ground conduit was added behind the chip which connects the ground plane on the chip to prevent any ground loops.

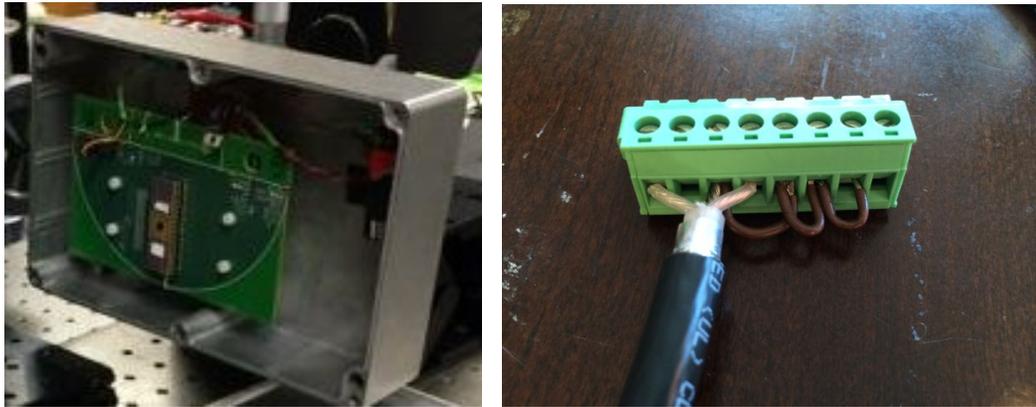
The board was then placed in the enclosure and cabled to the SMU using twinax cabling. Twinax is a twisted pair of conduit with a single shield around the pair. For the gate cable, the gate signal and signal reference (source) were connected to the twisted pair, and the shield was connected to the guard on the SMU. For the drain cable, the drain output signal and the signal reference (source, same as other) were connected to the

A. Test Equipment Description



- (a) The power and ground connections were isolated to single star points to minimize signal loops.
- (b) A single ground conduit was run behind the chip and connected to each ground on the chip to ensure a single ground plane for the circuit.

Figure A.1.: Re-wiring of fanout board.



- (a) Board mounted in the test enclosure.
- (b) SMU connector. The jumpers connect all of the guard signals together and to the shield on the twinax.

Figure A.2.: THz test enclosure and connections.

twisted pair, and the shield was connected to the guard on the SMU. On the enclosure

A. Test Equipment Description

side, all four twisted pair signals were isolated from the enclosure using feed through capacitors. This creates a balanced signal between the two twinax connections and the enclosure. We also left provisions to connect the guards to the enclosure ground. The enclosure ground is isolated from the reference ground (source) via these feed through capacitors.

As a comparison, Figure A.3 shows a similar test with the system in the UR lab before the rewire.

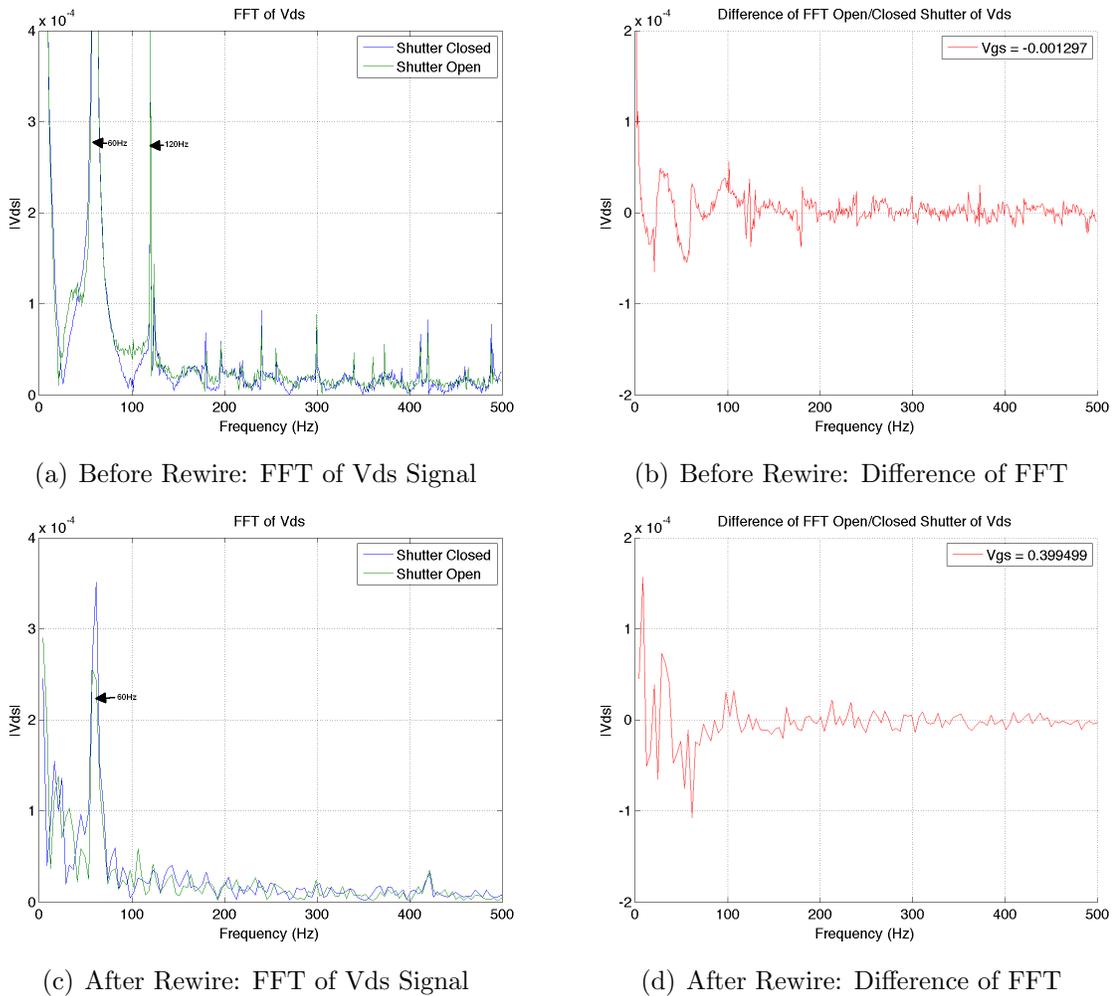


Figure A.3.: Noise comparison before and after rewire.

A. Test Equipment Description

Several grounding schemes using the enclosure ground, guard wires, and equipment grounds are tested. The configuration with the lowest noise was with the guard wires left connected on the SMU side only, the enclosure ground tied to the chassis ground on the SMU via the banana jack cable, and the test equipment running on a battery UPS. Other considerations to reduce the noise including a grounding strap that was affixed from the optics table to the earth ground of the circuit breaker box in the lab, and the computer used with the SMU running on battery to prevent ground noise entering through the serial connection.

B. Wavelength Frequency Conversion Table

B. Wavelength Frequency Conversion Table

Wavelength(um)	Wavelength(mm)	Wavelength (m)	Band	freq (Hz=1/s)	freq(GHz)	freq(THz)
0.1	0.0001	1.00E-07		3.00E+15	3.00E+24	3.00E+27
0.5	0.0005	5.00E-07	VIS	6.00E+14	6.00E+23	6.00E+26
1	0.001	1.00E-06		3.00E+14	3.00E+23	3.00E+26
1.5	0.0015	1.50E-06	SWIR	2.00E+14	2.00E+23	2.00E+26
2	0.002	2.00E-06		1.50E+14	1.50E+23	1.50E+26
2.5	0.0025	2.50E-06		1.20E+14	1.20E+23	1.20E+26
3	0.003	3.00E-06		1.00E+14	1.00E+23	1.00E+26
3.5	0.0035	3.50E-06	MWIR	8.57E+13	8.57E+22	8.57E+25
4	0.004	4.00E-06		7.50E+13	7.50E+22	7.50E+25
4.5	0.0045	4.50E-06		6.67E+13	6.67E+22	6.67E+25
5	0.005	5.00E-06		6.00E+13	6.00E+22	6.00E+25
5.5	0.0055	5.50E-06		5.45E+13	5.45E+22	5.45E+25
6	0.006	6.00E-06		5.00E+13	5.00E+22	5.00E+25
6.5	0.0065	6.50E-06		4.62E+13	4.62E+22	4.62E+25
7	0.007	7.00E-06		4.29E+13	4.29E+22	4.29E+25
7.5	0.0075	7.50E-06		4.00E+13	4.00E+22	4.00E+25
8	0.008	8.00E-06	LWIR	3.75E+13	3.75E+22	3.75E+25
8.5	0.0085	8.50E-06		3.53E+13	3.53E+22	3.53E+25
9	0.009	9.00E-06		3.33E+13	3.33E+22	3.33E+25
9.5	0.0095	9.50E-06		3.16E+13	3.16E+22	3.16E+25
10	0.01	1.00E-05		3.00E+13	3.00E+22	3.00E+25
15	0.015	1.50E-05		2.00E+13	2.00E+22	2.00E+25
20	0.02	2.00E-05		1.50E+13	1.50E+22	1.50E+25
25	0.025	2.50E-05		1.20E+13	12000.00	12.00
30	0.03	3.00E-05		1.00E+13	10000.00	10.00
35	0.035	3.50E-05		8.57E+12	8571.43	8.57
40	0.04	4.00E-05		7.50E+12	7500.00	7.50
45	0.045	4.50E-05		6.67E+12	6666.67	6.67
50	0.05	5.00E-05		6.00E+12	6000.00	6.00
55	0.055	5.50E-05		5.45E+12	5454.55	5.45
60	0.06	6.00E-05		5.00E+12	5000.00	5.00
65	0.065	6.50E-05		4.62E+12	4615.38	4.62
70	0.07	7.00E-05		4.29E+12	4285.71	4.29
75	0.075	7.50E-05		4.00E+12	4000.00	4.00
80	0.08	8.00E-05		3.75E+12	3750.00	3.75
85	0.085	8.50E-05		3.53E+12	3529.41	3.53
90	0.09	9.00E-05		3.33E+12	3333.33	3.33
95	0.095	9.50E-05		3.16E+12	3157.89	3.16
100	0.1	1.00E-04		3.00E+12	3000.00	3.00
150	0.15	1.50E-04		2.00E+12	2000.00	2.00
200	0.2	2.00E-04		1.50E+12	1500.00	1.50
250	0.25	2.50E-04		1.20E+12	1200.00	1.20
300	0.3	3.00E-04	THz	1.00E+12	1000.00	1.00
350	0.35	3.50E-04		8.57E+11	857.14	0.86
400	0.4	4.00E-04		7.50E+11	750.00	0.75
450	0.45	4.50E-04		6.67E+11	666.67	0.67
500	0.5	5.00E-04		6.00E+11	600.00	0.60
550	0.55	5.50E-04		5.45E+11	545.45	0.55
600	0.6	6.00E-04		5.00E+11	500.00	0.50
650	0.65	6.50E-04		4.62E+11	461.54	0.46
700	0.7	7.00E-04		4.29E+11	428.57	0.43
750	0.75	7.50E-04		4.00E+11	400.00	0.40
800	0.8	8.00E-04		3.75E+11	375.00	0.38
850	0.85	8.50E-04		3.53E+11	352.94	0.35
900	0.9	9.00E-04		3.33E+11	333.33	0.33
950	0.95	9.50E-04		3.16E+11	315.79	0.32
1000	1	1.00E-03		3.00E+11	300.00	0.30
1500	1.5	1.50E-03		2.00E+11	200.00	0.20
2000	2	2.00E-03		1.50E+11	150.00	0.15
2500	2.5	2.50E-03		1.20E+11	120.00	0.12
3000	3	3.00E-03		1.00E+11	100.00	0.10

B. Wavelength Frequency Conversion Table

3.50E+03	3.5	3.50E-03		8.57E+10	8.57E+01	8.57E-02
4.00E+03	4	4.00E-03		7.50E+10	7.50E+01	7.50E-02
4.50E+03	4.5	4.50E-03		6.67E+10	6.67E+01	6.67E-02
5.00E+03	5	5.00E-03		6.00E+10	6.00E+01	6.00E-02
5.50E+03	5.5	5.50E-03		5.45E+10	5.45E+01	5.45E-02
6.00E+03	6	6.00E-03		5.00E+10	5.00E+01	5.00E-02
6.50E+03	6.5	6.50E-03	mm Wave	4.62E+10	4.62E+01	4.62E-02
7.00E+03	7	7.00E-03	EHF	4.29E+10	4.29E+01	4.29E-02
7.50E+03	7.5	7.50E-03		4.00E+10	4.00E+01	4.00E-02
8.00E+03	8	8.00E-03		3.75E+10	3.75E+01	3.75E-02
8.50E+03	8.5	8.50E-03		3.53E+10	3.53E+01	3.53E-02
9.00E+03	9	9.00E-03		3.33E+10	3.33E+01	3.33E-02
9.50E+03	9.5	9.50E-03		3.16E+10	3.16E+01	3.16E-02
1.00E+04	10	1.00E-02		3.00E+10	3.00E+01	3.00E-02
1.50E+04	15	1.50E-02		2.00E+10	2.00E+01	2.00E-02
2.00E+04	20	2.00E-02		1.50E+10	1.50E+01	1.50E-02
2.50E+04	25	2.50E-02		1.20E+10	1.20E+01	1.20E-02
3.00E+04	30	3.00E-02		1.00E+10	1.00E+01	1.00E-02
3.50E+04	35	3.50E-02		8.57E+09	8.57E+00	8.57E-03
4.00E+04	40	4.00E-02		7.50E+09	7.50E+00	7.50E-03
4.50E+04	45	4.50E-02		6.67E+09	6.67E+00	6.67E-03
5.00E+04	50	5.00E-02		6.00E+09	6.00E+00	6.00E-03
5.50E+04	55	5.50E-02	SHF	5.45E+09	5.45E+00	5.45E-03
6.00E+04	60	6.00E-02		5.00E+09	5.00E+00	5.00E-03
6.50E+04	65	6.50E-02		4.62E+09	4.62E+00	4.62E-03
7.00E+04	70	7.00E-02		4.29E+09	4.29E+00	4.29E-03
7.50E+04	75	7.50E-02		4.00E+09	4.00E+00	4.00E-03
8.00E+04	80	8.00E-02		3.75E+09	3.75E+00	3.75E-03
8.50E+04	85	8.50E-02		3.53E+09	3.53E+00	3.53E-03
9.00E+04	90	9.00E-02		3.33E+09	3.33E+00	3.33E-03
9.50E+04	95	9.50E-02		3.16E+09	3.16E+00	3.16E-03
1.00E+05	100	1.00E-01		3.00E+09	3.00E+00	3.00E-03
1.50E+05	150	1.50E-01		2.00E+09	2.00E+00	2.00E-03
2.00E+05	200	2.00E-01		1.50E+09	1.50E+00	1.50E-03
2.50E+05	250	2.50E-01		1.20E+09	1.20E+00	1.20E-03
3.00E+05	300	3.00E-01		1.00E+09	1.00E+00	1.00E-03
3.50E+05	350	3.50E-01		8.57E+08	8.57E-01	8.57E-04
4.00E+05	400	4.00E-01		7.50E+08	7.50E-01	7.50E-04
4.50E+05	450	4.50E-01		6.67E+08	6.67E-01	6.67E-04
5.00E+05	500	5.00E-01		6.00E+08	6.00E-01	6.00E-04
5.50E+05	550	5.50E-01		5.45E+08	5.45E-01	5.45E-04
6.00E+05	600	6.00E-01	UHF	5.00E+08	5.00E-01	5.00E-04
6.50E+05	650	6.50E-01		4.62E+08	4.62E-01	4.62E-04
7.00E+05	700	7.00E-01		4.29E+08	4.29E-01	4.29E-04
7.50E+05	750	7.50E-01		4.00E+08	4.00E-01	4.00E-04
8.00E+05	800	8.00E-01		3.75E+08	3.75E-01	3.75E-04
8.50E+05	850	8.50E-01		3.53E+08	3.53E-01	3.53E-04
9.00E+05	900	9.00E-01		3.33E+08	3.33E-01	3.33E-04
9.50E+05	950	9.50E-01		3.16E+08	3.16E-01	3.16E-04
1.00E+06	1000	1.00E+00		3.00E+08	3.00E-01	3.00E-04
1.50E+06	1.50E+03	1.50E+00		2.00E+08	2.00E-01	2.00E-04
2.00E+06	2.00E+03	2.00E+00		1.50E+08	1.50E-01	1.50E-04
2.50E+06	2.50E+03	2.50E+00		1.20E+08	1.20E-01	1.20E-04
3.00E+06	3.00E+03	3.00E+00		1.00E+08	1.00E-01	1.00E-04
3.50E+06	3.50E+03	3.50E+00		8.57E+07	8.57E-02	8.57E-05
4.00E+06	4.00E+03	4.00E+00		7.50E+07	7.50E-02	7.50E-05
4.50E+06	4.50E+03	4.50E+00		6.67E+07	6.67E-02	6.67E-05
5.00E+06	5.00E+03	5.00E+00		6.00E+07	6.00E-02	6.00E-05
5.50E+06	5.50E+03	5.50E+00		5.45E+07	5.45E-02	5.45E-05
6.00E+06	6.00E+03	6.00E+00	VHF	5.00E+07	5.00E-02	5.00E-05
6.50E+06	6.50E+03	6.50E+00		4.62E+07	4.62E-02	4.62E-05
7.00E+06	7.00E+03	7.00E+00		4.29E+07	4.29E-02	4.29E-05
7.50E+06	7.50E+03	7.50E+00		4.00E+07	4.00E-02	4.00E-05

B. Wavelength Frequency Conversion Table

8.00E+06	8.00E+03	8.00E+00		3.75E+07	3.75E-02	3.75E-05
8.50E+06	8.50E+03	8.50E+00		3.53E+07	3.53E-02	3.53E-05
9.00E+06	9.00E+03	9.00E+00		3.33E+07	3.33E-02	3.33E-05
9.50E+06	9.50E+03	9.50E+00		3.16E+07	3.16E-02	3.16E-05
1.00E+07	1.00E+04	1.00E+01		3.00E+07	3.00E-02	3.00E-05
1.50E+07	1.50E+04	1.50E+01		2.00E+07	2.00E-02	2.00E-05
2.00E+07	2.00E+04	2.00E+01		1.50E+07	1.50E-02	1.50E-05
2.50E+07	2.50E+04	2.50E+01		1.20E+07	1.20E-02	1.20E-05
3.00E+07	3.00E+04	3.00E+01		1.00E+07	1.00E-02	1.00E-05
3.50E+07	3.50E+04	3.50E+01		8.57E+06	8.57E-03	8.57E-06
4.00E+07	4.00E+04	4.00E+01		7.50E+06	7.50E-03	7.50E-06
4.50E+07	4.50E+04	4.50E+01		6.67E+06	6.67E-03	6.67E-06
5.00E+07	5.00E+04	5.00E+01		6.00E+06	6.00E-03	6.00E-06
5.50E+07	5.50E+04	5.50E+01		5.45E+06	5.45E-03	5.45E-06
6.00E+07	6.00E+04	6.00E+01	HF	5.00E+06	5.00E-03	5.00E-06
6.50E+07	6.50E+04	6.50E+01		4.62E+06	4.62E-03	4.62E-06
7.00E+07	7.00E+04	7.00E+01		4.29E+06	4.29E-03	4.29E-06
7.50E+07	7.50E+04	7.50E+01		4.00E+06	4.00E-03	4.00E-06
8.00E+07	8.00E+04	8.00E+01		3.75E+06	3.75E-03	3.75E-06
8.50E+07	8.50E+04	8.50E+01		3.53E+06	3.53E-03	3.53E-06
9.00E+07	9.00E+04	9.00E+01		3.33E+06	3.33E-03	3.33E-06
9.50E+07	9.50E+04	9.50E+01		3.16E+06	3.16E-03	3.16E-06
1.00E+08	1.00E+05	1.00E+02		3.00E+06	3.00E-03	3.00E-06

C. MATLAB Test Scripts

The following section includes the MATLAB scripts used in testing of the first two generations of test transistors.

C.1. Programming the Keithley 2602

The source measurement unit(SMU) can communicate via several connections but for this work a serial connection was used. Parameters and pinouts can be found in the Reference Manual. I preferred to use a Macbook Pro running Matlab 2013b, a USB-to-Serial adapter, and write my own scripts. This allowed all of the data to be commanded and read directly into MATLAB from the SMU which makes data analysis very easy. The programming language the SMU uses is script based, so commands are sent in ASCII strings over the serial connection, and the SMU then runs the programs on the unit itself. There are two approaches to take by doing this. You can either send each command individually, or send a whole program, or section of the program to be run on the SMU one at a time. There are pros and cons to each. Because of the speed limitation of the serial communication, I found it best to send one command at a time, unless a loop was required to perform a voltage-measurement sweep. This allows the MATLAB send and receive as coded. One issue with this is that the serial input buffer will fill up quickly if lots of data points are being read back to the computer. This has to be managed properly in the MATLAB code or timeouts and data overwrites will occur. Another issue is managing whether the SMU or MATLAB is managing variable values, and making sure they are passed back and forth properly. An example script would then look something like:

1. Setup MATLAB variables
2. Command SMU to setup each parameter individually (modes,limits, etc)
3. Command SMU to perform the testing loop all at once
4. Read back variables individually from SMU to MATLAB and ensure vector lengths are correct
5. Reset SMU for next experiment

C.2. TSP Transconductance Code

This is the example code tsp file provided by Keithley [52] which sweeps the gate voltage and measures the drain current to provide a transconductance curve. It is written in the Keithley programming language only. This can be run by using the Test Script Builder software.

Code

--Transconductance():

This program sources a voltage bias on a drain-source of a FET (Vds), sources a voltage on the gate (Vgs1), and measures the drain-source current (Id1). Then, another source value (Vgs2) is sourced and the Ids2 is measured.

The Transconductance (gfs) is then calculated by taking the change in Ids divided by the change in Vgs.

The drain-source voltage (Vds), Transconductance (gfs), gate-source voltage (Vgs), and drain-source current (Id) are returned.

Required equipment:

- (1) Dual channel Keithley 2600 Series System Sourcemeter(c)
- (1) SD210 N-channel FET

Running this script creates functions which can be used to create a Transconductance Test of FET's. The default values are for an N-channel SD210 FET.

The functions created are:

1. Transconductance(vgsstart, vgsstop, vgssteps, vdsbias)
--Default values vgsstart = 0V, vgsstop = 5V, vgssteps = 100,
vdsbias = 10V
2. Check_Comp()

See detailed information listed in individual functions

To Run:

- 1) From Test Script Builder
 - Right click in the program window, select "Run as TSP"
 - At the TSP> prompt in the Instrument Control Panel, type Transconductance()
- 2) From an external program
 - Send the entire program text as a string using standard GPIB Write

calls.

Rev1: JAC 6.18.2007

]]--

----- Keithley TSP Function -----

--function Transconductance(vgsstart, vgsstop, vgssteps, vdsbias)
--Configure SMUA to source a user defined voltage on the drain-source (Vds)
--while SMUB performs a fixed voltage bias (Vgs)on the gate-source

vgsstart=0
vgsstop=2.5
vgssteps=100 --and the Ids is measured.
vdsbias=0.1 --SMUB then steps to the next base current and the
Ic is measured.
--Returns measured Vds, Vgs, Id, gfs values are
returned.

--Global variables

l_icmpl = 1E-3 --Source compliance

--Shared local variables

l_nplc = 1 --Integration rate of measurement

--Local sweep variables

l_vgsstart = vgsstart --Vgs start voltage
l_vgsstop = vgsstop --Vgs sweep stop voltage
l_vgssteps = vgssteps --Number of steps in sweep
l_vdsbias = vdsbias --Drain-source voltage

--Default values and level check

if (l_vgsstart == nil) then --Use default value
l_vgsstart = 0
end --if

if (l_vgsstart > 1) then --Coerce value

l_vgsstart = 0.1
end --if

if (l_vgsstop == nil) then --Use default value

l_vgsstop = 1
end --if

if (l_vgsstop > 5) then --Coerce value

l_vgsstop = 5
end --if

```
if (l_vgssteps == nil) then --Use default value
l_vgssteps = 50
end --if

if (l_vgssteps > 100) then --Coerce value
l_vgssteps = 100
end --if

l_vgsstep = (l_vgsstop - l_vgsstart)/ (l_vgssteps - 1) --Vbe step size
l_vgssource_val = l_vgsstart --Source value during sweep
l_i = 1 --Iteration variable

if (l_vds_bias == nil) then --Use default value
l_vds_bias = 0.1
end --if

if (l_vds_bias > 1) then --Coerce value
l_vds_bias = 1
end --if

--Data tables
l_vgs = {} --Create data table for gate-source voltage
    l_id = {} --Create data table for drain-source current
    l_gfs = {} --Create data table for transconductance (gfs)

smua.reset() --Reset SMU
smub.reset() --Reset SMU

errorqueue.clear() --Clear the error queue

--Configure Collector/Emitter (SMUA) source and measure settings
smua.source.func = smua.OUTPUT_DCVOLTS
    smua.source.autorangev = smua.AUTORANGE_ON --Enable source autorange
    smua.source.levelv = 0
    smua.source.limiti = l_icmpl
    smua.measure.autorangei = smua.AUTORANGE_ON --Enable measure autorange
    smua.measure.autozero = smua.AUTOZERO_AUTO
    smua.measure.nplc = l_nplc --Measurement integration rate
    smua.source.output = smua.OUTPUT_ON --Enable Output

--Configure Base (SMUB) source and measure settings
smub.source.func = smub.OUTPUT_DCVOLTS
    smub.source.autorangev = smub.AUTORANGE_ON --Enable source autorange
    smub.source.levelv = 0
    smub.source.limiti = l_icmpl
```

C. MATLAB Test Scripts

```
smub.measure.autorangev = smub.AUTORANGE_ON --Enable measure autorange
smub.measure.autozero = smub.AUTOZERO_AUTO
smub.measure.nplc = l_nplc --Measurement integration rate
smub.source.output = smub.OUTPUT_ON --Enable Output
smua.source.levelv = l_vds_bias

--Execute sweep
for l_i = 1,l_vgssteps do
  if (l_i == 1) then --Intialize start source value
    l_vgssource_val = l_vgsstart
  end --if

--delay(1)

l_vgs[l_i] = smub.measure.v() --Measure Vgs
l_id[l_i] = smua.measure.i() --Measure Id
l_vgssource_val = l_vgssource_val + l_vgsstep --Calculate new source value

if (l_i == l_vgssteps) then --Reinitialize voltage value after last iteration
  l_vgssource_val = l_vgsstart
end --if

smub.source.levelv = l_vgssource_val --Increment source

  end --for

smua.source.output = smua.OUTPUT_OFF --Disable output
smub.source.output = smub.OUTPUT_OFF --Disable output
smua.source.levelv = 0 --Return source to bias level
smub.source.levelv = 0 --Return source to bias level

--Print_Data(l_vds_bias, l_vgssteps, l_vgs, l_id)

--end--function Transconductance()

--function Print_Data(vdsbias, vgssteps,vgs, id)
--Calculate Gfs value and print data to output queue

--Local Variables
--l_vds_bias = vdsbias --Vds bias value
--l_vgs_steps = vgssteps --Number of steps in Vgs sweep
--l_vgs = vgs --Gate-source Voltage data
--l_id = id --Drain-source current data
l_gfs = {} --Table for Transconductance calculations
l_i = 1 --Iteration variable
```

```
--Calculate gfs values and populate table

for l_i = 1,vgssteps do
if (l_i ~= 1) then --If not the first iteration, calculate gfs
l_gfs[l_i] = (l_id[l_i] - l_id[l_i - 1])/(l_vgs[l_i] - l_vgs[l_i - 1])
--gfs = dId/dVgs
end--if
end --for

l_i = 1 --Reinitialize Vgs iteration variable
print("")
print("Vds", l_vds_bias)
print("Vgs (V)","Id (A)","gfs (s)")
for l_i = 2, vgssteps do
print(l_vgs[l_i],l_id[l_i], l_gfs[l_i])
end --for
--end --function Print_Data()
--end Transconductance()

--[[
```

C.3. Serial Port Enable Code

This script will setup the USB-to-Serial adapter as a serial port 'obj1' in MATLAB. Commands are then written to the port as strings. Scanning the port reads data on the input buffer back into MATLAB as variables.

enableserial.m

```
1 %% Load serial port object and run beep sequence
2 clearvars -except testnum;
3
4 % Find a serial port object.
5 obj1 = instrfind('Type', 'serial', 'Port', '/dev/tty.usbserial', 'Tag', ...
6     '');
7
8 % Create the serial port object if it does not exist
9 % otherwise use the object that was found.
10 if isempty(obj1)
11     obj1 = serial('/dev/tty.usbserial');
12 else
13     fclose(obj1);
14     obj1 = obj1(1)
15 end
16
17 % Set Buffer Sizes
18 set(obj1, 'BaudRate', 115200, 'Timeout', 120);
19 set(obj1, 'InputBufferSize', 5000000);
20 set(obj1, 'OutputBufferSize', 12000);
21
22 % Connect to instrument object, obj1.
23 fopen(obj1);
24
25 % Communicating with instrument object, obj1.
26 fprintf(obj1, '*RST'); %command to reset the keithley
27 fprintf(obj1, 'smua.reset()'); % Reset SMU
28 fprintf(obj1, 'smub.reset()'); % Reset SMU
29 fprintf(obj1, 'errorqueue.clear()'); % Clear the error queue
30 fprintf(obj1, '*IDN?'); %Request identification from keithley
31 data1 = fscanf(obj1); %read identification info
32
33 %Beeper tones for startup
34 fprintf(obj1, 'beeper.enable = beeper.ON'); %enable beeper
```

```
34 fprintf(obj1, 'beeper.beep(0.12, 500)'); %send beeper tone ...  
    (duration,frequency)  
35 fprintf(obj1, 'beeper.beep(0.12, 800)'); %send beeper tone ...  
    (duration,frequency)  
36 fprintf(obj1, 'beeper.beep(0.12, 1000)'); %send beeper tone ...  
    (duration,frequency)  
37 fprintf(obj1, 'beeper.beep(0.12, 2000)'); %send beeper tone ...  
    (duration,frequency)
```

C.4. Transconductance Script

This script will perform a transconductance sweep on a MOSFET. It is modified from the provided Keithley TSP script from Section C.2. Channel A is connected to the Drain and Channel B is connected to the Gate. These connections are the same for all scripts.

transconductance.m

```
1 %% Transconductance script
2 % This program sources a voltage bias on a drain-source of a FET (Vds),
3 % sources a voltage on the gate (Vgs1), and measures the drain-source ...
4 % current (Id1).
5 %
6 % Then, another source value (Vgs2) is sourced and the Ids2 is measured.
7 %
8 % The Transconductance (gfs) is then calculated by taking the change in Ids
9 % divided by the change in Vgs.
10 %
11 % The drain-source voltage (Vds), Transconductance (gfs), gate-source ...
12 % voltage (Vgs),
13 % and drain-source current (Id) are returned.
14 close all;
15 clearvars -except obj1 testnum;
16 testnum=testnum+1;
17
18 cd '/Users/gfertig/Dropbox/Thesis/THz Project/Testing/20140418 Test Data'
19 foldername = [num2str(testnum), '-OldBoard-Chip1-T5-Trans'];
20 fprintf(obj1, '*RST'); %command to reset the keithley
21 fprintf(obj1, 'smua.reset()'); % Reset SMU
22 fprintf(obj1, 'smub.reset()'); % Reset SMU
23 fprintf(obj1, 'errorqueue.clear()'); % Clear the error queue
24
25 fprintf(obj1, 'display.clear()');
26 fprintf(obj1, 'display.settext("$BTest in Progress$B)");
27 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
28     (duration,frequency)
29 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
30     (duration,frequency)
31
32 fprintf(obj1, ['digio.writebit(2,0) ']); %Set digio bit 2 to High for ...
33     shutter open
34 fprintf(obj1, 'vgsstart=0');
35 fprintf(obj1, 'vgsstop=2.5');
36 fprintf(obj1, 'vgssteps=100');
```

C. MATLAB Test Scripts

```
31 vgssteps=100;
32 fprintf(obj1, 'vdsbias=0.1');
33
34 fprintf(obj1, 'l.icmpl = 1E-3'); % Source compliance
35 fprintf(obj1, 'l.nplc = 1'); % Integration rate of measurement
36
37 % Local sweep variables
38 fprintf(obj1, 'l.vgsstart = vgsstart'); % Vgs start voltage
39 fprintf(obj1, 'l.vgsstop = vgsstop'); % Vgs sweep stop voltage
40 fprintf(obj1, 'l.vgssteps = vgssteps'); % Number of steps in sweep
41 fprintf(obj1, 'l.vdsbias = vdsbias'); % Drain-source voltage
42
43 % Default values and level check
44 fprintf(obj1, 'if (l.vgsstart == nil) then l.vgsstart = 0 end'); % Use ...
    default value
45 fprintf(obj1, 'if (l.vgsstart > 1) then l.vgsstart = 0.1 end'); % ...
    Coerce value
46 fprintf(obj1, 'if (l.vgsstop == nil) then l.vgsstop = 1 end'); % Use ...
    default value
47 fprintf(obj1, 'if (l.vgsstop > 5) then l.vgsstop = 5 end'); % Coerce value
48 fprintf(obj1, 'if (l.vgssteps == nil) then l.vgssteps = 50 end'); % Use ...
    default value
49 fprintf(obj1, 'if (l.vgssteps > 100) then l.vgssteps = 100 end'); % ...
    Coerce value
50 fprintf(obj1, 'l.vgsstep = (l.vgsstop - l.vgsstart)/ (l.vgssteps - ...
    1)'); % Vbe step size
51 fprintf(obj1, 'l.vgssource_val = l.vgsstart'); % Source value during sweep
52 fprintf(obj1, 'if (l.vdsbias == nil) then l.vdsbias = 0.1 end'); % Use ...
    default value
53 fprintf(obj1, ['if (l.vdsbias > 1) then ',...
54             'l.vdsbias = 1 ',...
55             'end']); % Coerce value
56
57 % Data tables
58 fprintf(obj1, 'l.vgs = {}'); % Create data table for gate-source voltage
59 fprintf(obj1, 'l.id = {}'); % Create data table for drain-source current
60 fprintf(obj1, 'l.gfs = {}'); % Create data table for transconductance (gfs)
61
62 % fprintf(obj1, 'smua.reset()'); % Reset SMU
63 % fprintf(obj1, 'smub.reset()'); % Reset SMU
64 % fprintf(obj1, 'errorqueue.clear()'); % Clear the error queue
65
66 % Configure Collector/Emitter (SMUA) source and measure settings
67 fprintf(obj1, 'smua.source.func = smua.OUTPUT_DCVOLTS');
68 fprintf(obj1, 'smua.source.autorangev = smua.AUTORANGE_ON'); % Enable ...
    source autorange
```

C. MATLAB Test Scripts

```
69 fprintf(obj1, 'smua.source.levelv = 0');
70 fprintf(obj1, 'smua.source.limiti = l_icmpl');
71 fprintf(obj1, 'smua.measure.autorangei = smua.AUTORANGE_ON'); % Enable ...
    measure autorange
72 %fprintf(obj1, 'smua.measure.rangei = smua.AUTORANGE_ON'); % Enable ...
    measure autorange
73 fprintf(obj1, 'smua.measure.autozero = smua.AUTOZERO_AUTO');
74 fprintf(obj1, 'smua.measure.nplc = l_nplc'); % Measurement integration rate
75 fprintf(obj1, 'smua.source.output = smua.OUTPUT_ON'); % Enable Output
76
77 % Configure Base (SMUB) source and measure settings
78 fprintf(obj1, 'smub.source.func = smub.OUTPUT_DCVOLTS');
79 fprintf(obj1, 'smub.source.autorangev = smub.AUTORANGE_ON'); % Enable ...
    source autorange
80 fprintf(obj1, 'smub.source.levelv = 0');
81 fprintf(obj1, 'smub.source.limiti = l_icmpl');
82 fprintf(obj1, 'smub.measure.autorangev = smub.AUTORANGE_ON'); % Enable ...
    measure autorange
83 fprintf(obj1, 'smub.measure.autozero = smub.AUTOZERO_AUTO');
84 fprintf(obj1, 'smub.measure.nplc = l_nplc'); % Measurement integration rate
85 fprintf(obj1, 'smub.source.output = smub.OUTPUT_ON'); % Enable Output
86 fprintf(obj1, 'smua.source.levelv = l_vdsbias');
87
88 % Execute sweep
89 fprintf(obj1, 'display.clear()');
90 fprintf(obj1, 'display.screen = 2');
91 fprintf(obj1, ['for l_i = 1,l_vgssteps do ',...
92             'if (l_i == 1) then ',... % Intialize start source ...
    value
93             'l_vgssource_val = l_vgsstart ',...
94             'end ',...
95             'l_vgs[l_i] = smub.measure.v() ',... % Measure Vgs
96             'l_id[l_i] = smua.measure.i() ',... % Measure Id
97             'l_vgssource_val = l_vgssource_val + l_vgsstep ',... % ...
    Calculate new source value
98             'if (l_i == l_vgssteps) then ',... % Reinitialize ...
    voltage value after last iteration
99             'l_vgssource_val = l_vgsstart ',...
100            'end ',...
101            'smub.source.levelv = l_vgssource_val ',... % Increment ...
    source
102            'end']);
103
104 fprintf(obj1, 'smua.source.output = smua.OUTPUT_OFF'); % Disable output
105 fprintf(obj1, 'smub.source.output = smub.OUTPUT_OFF'); % Disable output
106 fprintf(obj1, 'smua.source.levelv = 0'); % Return source to bias level
```

```

107 fprintf(obj1, 'smub.source.levelv = 0'); % Return source to bias level
108
109 fprintf(obj1, 'display.clear()');
110 fprintf(obj1, 'display.settext("$BSending Data$B")');
111
112 fprintf(obj1, 'l_gfs = {}'); % Table for Transconductance calculations
113 fprintf(obj1, 'l_i = 1'); % Iteration variable
114
115 % Calculate gfs values and populate table
116 fprintf(obj1, ['for l_i = 1,vgssteps do ',...
117             'if (l_i ≠ 1) then ',... % If not the first ...
118             'iteration, calculate gfs
119             'l_gfs[l_i] = (l_id[l_i] - l_id[l_i - ...
120             '1)]/(l_vgs[l_i] - l_vgs[l_i - 1]) ',...
121             'end ',...
122             'end']);
123
124 % Loop for data read
125 fprintf(obj1, 'print(l_vdsbias)');
126 vdsbias = str2num(fscanf(obj1)); %Read data from serial port and assign ...
127     to var
128
129 for i=1:vgssteps %Loop to print all tables to matlab variables
130     fprintf(obj1, ['print(l_vgs[' ,num2str(i),']) \n'] );
131     temp1 = fscanf(obj1);
132     temp1 = strtrim(temp1);
133     if strcmp(temp1,'nil') == 1
134         vgs(i) = 0;
135     else
136         vgs(i)= str2num(temp1);
137     end
138
139     fprintf(obj1, ['print(l_id[' ,num2str(i),']) \n'] );
140     temp2 = fscanf(obj1);
141     temp2 = strtrim(temp2);
142     if strcmp(temp2,'nil') == 1
143         id(i) = 0;
144     else
145         id(i)= str2num(temp2);
146     end
147
148     fprintf(obj1, ['print(l_gfs[' ,num2str(i),']) \n'] );
149     temp3 = fscanf(obj1);
150     temp3 = strtrim(temp3);
151     if strcmp(temp3,'nil') == 1
152         gfs(i) = 0;

```

```

150     else
151         gfs(i)= str2num(temp3);
152     end
153 end
154
155 %% Print to Keithley screen and beep complete sequence
156 fprintf(obj1, 'display.clear()');
157 fprintf(obj1, 'display.settext("Test Complete")');
158 fprintf(obj1, 'beeper.beep(0.12, 500)'); %send beeper tone ...
    (duration,frequency)
159 fprintf(obj1, 'beeper.beep(0.12, 800)'); %send beeper tone ...
    (duration,frequency)
160 fprintf(obj1, 'beeper.beep(0.12, 1000)'); %send beeper tone ...
    (duration,frequency)
161 fprintf(obj1, 'beeper.beep(0.12, 2000)'); %send beeper tone ...
    (duration,frequency)
162
163 %% Plot Transconductance Curves
164 figure();
165 h = plot (vgs,gfs, '-*r');
166 hold on;
167 plot (vgs,id, '-b');
168 x = get (h, 'XData'); % get the plotted data
169 y = get (h, 'YData');
170 imin = find(min(y) == y);% find the index of the min and max
171 imax = find(max(y) == y);
172 % text(x(imin),y(imin),[' Minimum = ',num2str(y(imin))],...
173 %     'VerticalAlignment','middle',...
174 %     'HorizontalAlignment','left',...
175 %     'FontSize',14)
176 text(x(imax),y(imax),['Peak = ',num2str(y(imax))],...
177     'VerticalAlignment','bottom',...
178     'HorizontalAlignment','right',...
179     'FontSize',14)
180 set(gca, 'fontsize',14);
181 set(gcf, 'color', 'w');
182 grid on;
183 legend('Gfs','Id (A)');
184 title('MOSFET Transconductance Curve');
185 xlabel('Gate Voltage, Vgs (V)');
186 ylabel('Gfs (dI/dV); Id (A)');
187 mkdir([foldername]);
188 print('-dpng', [foldername, '\transconductance.png']);
189
190 %% Reset Keithley & Save Data
191 save([foldername, '/transconductance.mat']);

```

```
192 fprintf(obj1, '*RST'); %command to reset the Keithley
```

C.5. Drain Sweep Scripts

This script will sweep the gate voltage over a specified range, and at each gate voltage, perform a sweep of the drain voltage over a specified range while measuring the current. This then in turn provides the family of curves of the MOSFET. The drainsweep-condplot script also calculates and plots the conductance and resistance based on measured values.

drainsweep.m

```
1  %% Sweeps Vgs, and conducts a Source-Drain Sweep for each fixed Vgs
2
3  clearvars -except obj1 testnum;
4  testnum=testnum+1;
5
6  cd '/Users/gfertig/Dropbox/Thesis/THz Project/Testing/20140207 Test Data'
7  foldername = [num2str(testnum), '-OldBoard-T3-DrainSweep'];
8  fprintf(obj1, '*RST'); %command to reset the Keithley
9  fprintf(obj1, 'smua.reset()'); % Reset SMU
10 fprintf(obj1, 'smub.reset()'); % Reset SMU
11 fprintf(obj1, 'errorqueue.clear()'); % Clear the error queue
12
13 fprintf(obj1, 'display.clear()');
14 fprintf(obj1, 'display.settext("$BTest in Progress$B")');
15 fprintf(obj1, 'beeper.beep(0.1, 500)'); %send beeper tone ...
    (duration,frequency)
16 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
17
18 for j=1:3
19     vgs_bias = 0.4+0.05*j;
20     fprintf(obj1, ['vgs_bias=', num2str(vgs_bias)]);
21     fprintf(obj1, 'vdsstart=0');
22     fprintf(obj1, 'vdsstop=0.3');
23     fprintf(obj1, 'vdssteps=121');
24     vdssteps = 121;
25
26     % Global variables
27     fprintf(obj1, 'licmpl = 1E-3'); % Source compliance
28
29     % Shared local variables
30     fprintf(obj1, 'lnplc = 1'); % Integration rate of measurement
31
32     % Default values and level check
```

```

33 fprintf(obj1, 'if (vdsstart == nil) then vdsstart = 0 end'); % Use ...
    default value
34 fprintf(obj1, 'if (vdsstart > 1) then vdsstart = 0.1 end'); % ...
    Coerce value
35 fprintf(obj1, 'if (vdsstop == nil) then vdsstop = 3 end'); % Use ...
    default value
36 fprintf(obj1, 'if (vdsstop > 3.3) then vdsstop = 3.3 end'); % ...
    Coerce value
37 fprintf(obj1, 'if (vdssteps == nil) then vdssteps = 50 end'); % Use ...
    default value
38 %fprintf(obj1, 'if (vdssteps > 100) then vdssteps = 100 end'); % ...
    Coerce value
39 fprintf(obj1, 'vdsstep = (vdsstop - vdsstart) / (vdssteps - 1)'); ...
    % step size
40 fprintf(obj1, 'vdssource_val = vdsstart'); % Source value during sweep
41 fprintf(obj1, 'l_i = 1'); % Iteration variable
42
43 fprintf(obj1, 'if (vgs.bias == nil) then vgs.bias = 0.5 end'); % ...
    Use default value
44 fprintf(obj1, 'if (vgs.bias > 3.3) then vgs.bias = 3.3 end'); % ...
    Coerce value
45
46 % Data tables
47 fprintf(obj1, 'vds = {}'); % Create data table for drain-source ...
    voltage
48 fprintf(obj1, 'id = {}'); % Create data table for drain-source current
49 fprintf(obj1, 'vgs = {}'); % Create data table for gate
50
51 % fprintf(obj1, 'smua.reset()'); % Reset SMU drain
52 % fprintf(obj1, 'smub.reset()'); % Reset SMU gate
53 % fprintf(obj1, 'errorqueue.clear()'); % Clear the error queue
54
55 % Configure (SMUA) source and measure settings
56 fprintf(obj1, 'smua.source.func = smua.OUTPUT_DCVOLTS');
57 fprintf(obj1, 'smua.source.autorangev = smua.AUTORANGE_ON'); % ...
    Enable source autorange
58 fprintf(obj1, 'smua.source.levelv = 0');
59 fprintf(obj1, 'smua.source.limiti = l_icmpl');
60 %fprintf(obj1, 'smua.measure.autorangei = smua.AUTORANGE_ON'); % ...
    Enable measure autorange
61 fprintf(obj1, 'smua.measure.rangei = 1E-6'); % Enable measure ...
    autorange
62 fprintf(obj1, 'smua.measure.autozero = smua.AUTOZERO_AUTO');
63 fprintf(obj1, 'smua.measure.nplc = l_nplc'); % Measurement ...
    integration rate
64 fprintf(obj1, 'smua.source.output = smua.OUTPUT_ON'); % Enable Output

```

```

65
66 % Configure SMUb gate and measure settings
67 fprintf(obj1, 'smub.source.func = smub.OUTPUT_DCVOLTS');
68 fprintf(obj1, 'smub.source.autorangev = smub.AUTORANGE_ON'); % ...
    Enable source autorange
69 fprintf(obj1, 'smub.source.levelv = 0');
70 fprintf(obj1, 'smub.source.limiti = l_icmpl');
71 fprintf(obj1, 'smub.measure.autorangev = smub.AUTORANGE_ON'); % ...
    Enable measure autorange
72 fprintf(obj1, 'smub.measure.autozero = smub.AUTOZERO_AUTO');
73 fprintf(obj1, 'smub.measure.nplc = l_nplc'); % Measurement ...
    integration rate
74 fprintf(obj1, 'smub.source.output = smub.OUTPUT_ON'); % Enable Output
75 fprintf(obj1, 'smub.source.levelv = vgs_bias');
76
77 % Execute sweep
78 fprintf(obj1, 'display.clear()');
79 fprintf(obj1, 'display.screen = 2');
80 fprintf(obj1, ['for l_i = 1,vdssteps do ',...
81     'if (l_i == 1) then ',...% Intialize start source value
82     'vdssource_val = vdsstart ',...
83     'end ' ,...
84     'vgs[l_i] = smub.measure.v() ',... % Measure Vgs
85     'id[l_i] = smua.measure.i() ',... % Measure Id
86     'vds[l_i] = smua.measure.v() ',...
87     'vdssource_val = vdssource_val + vdsstep ',... % Calculate new ...
        source value
88     'if (l_i == vdssteps) then ',... % Reinitialize voltage value ...
        after last iteration
89     'vdssource_val = vdsstart ',...
90     'end ',...
91     'smua.source.levelv = vdssource_val ',... % Increment source
92     'end ']);
93
94 fprintf(obj1, 'smua.source.output = smua.OUTPUT_OFF'); % Disable ...
    output
95 fprintf(obj1, 'smub.source.output = smub.OUTPUT_OFF'); % Disable ...
    output
96 fprintf(obj1, 'smua.source.levelv = 0'); % Return source to bias level
97 fprintf(obj1, 'smub.source.levelv = 0'); % Return source to bias level
98
99 fprintf(obj1, 'display.clear()');
100 fprintf(obj1, 'display.settext("$BSending Data$B")');
101
102 fprintf(obj1, 'l_i = 1'); % Reinitialize iteration variable
103

```

C. MATLAB Test Scripts

```
104 % Loop for data read
105 fprintf(obj1, 'print(vgs.bias)');
106 vgsbias(j) = str2num(fscanf(obj1)); %Read data from serial port and ...
    assign to var
107
108 for i=1:vdssteps %Loop to print all tables to matlab variables
109     fprintf(obj1, ['print(id[' ,num2str(i),']) \n' ] );
110     temp2 = fscanf(obj1);
111     temp2 = strtrim(temp2);
112     if strcmp(temp2,'nil') == 1
113         id(j,i) = 0;
114     else
115         id(j,i)= str2num(temp2);
116     end
117
118     fprintf(obj1, ['print(vds[' ,num2str(i),']) \n' ] );
119     temp3 = fscanf(obj1);
120     temp3 = strtrim(temp3);
121     if strcmp(temp3,'nil') == 1
122         vds(j,i) = 0;
123     else
124         vds(j,i)= str2num(temp3);
125     end
126 end
127 end
128
129 %% Print to Keithley screen and beep complete sequence
130 fprintf(obj1, 'display.clear()');
131 fprintf(obj1, 'display.settext("Test Complete")');
132 fprintf(obj1, 'beeper.beep(0.12, 500)'); %send beeper tone ...
    (duration,frequency)
133 fprintf(obj1, 'beeper.beep(0.12, 800)'); %send beeper tone ...
    (duration,frequency)
134 fprintf(obj1, 'beeper.beep(0.12, 1000)'); %send beeper tone ...
    (duration,frequency)
135 fprintf(obj1, 'beeper.beep(0.12, 2000)'); %send beeper tone ...
    (duration,frequency)
136
137 %% Plot Curves
138 % figure();
139 % plot ...
    (vds(3,:),id(3,:),vds(4,:),id(4,:),vds(5,:),id(5,:),vds(6,:),id(6,:),
140 vds(7,:),id(7,:),vds(8,:),id(8,:));
141 % set(gca,'fontsize',14);
142 % set(gcf, 'color', 'w');
143 % grid on;
```

C. MATLAB Test Scripts

```
144 % legend([num2str(vgsbias(3)), ' Vgs'], [num2str(vgsbias(4)), ' ...
      Vgs'], [num2str(vgsbias(5)), ' Vgs'], [num2str(vgsbias(6)), ' ...
      Vgs'], [num2str(vgsbias(7)), ' Vgs'], [num2str(vgsbias(8)), ' Vgs']);
145 % set(legend, ...
146 %     'Position', [0.215939049121225 0.582099669520804 0.116189725101397 ...
      0.286929172299835]);
147 % title('MOSFET Family of Curves');
148 % xlabel('Drain Voltage, Vds (V)');
149 % ylabel('Drain Current, Id (A)');
150 % mkdir([foldername]);
151 % print('-dpng', [foldername, '\familyofcurves2.png']);
152
153 %% Plot Curves
154 figure();
155 plot (vds(1,:), id(1,:), vds(2,:), id(2,:), vds(3,:), id(3,:));
156 set(gca, 'fontsize', 14);
157 set(gcf, 'color', 'w');
158 grid on;
159 legend([num2str(vgsbias(1)), ' Vgs'], [num2str(vgsbias(2)), ' ...
      Vgs'], [num2str(vgsbias(3)), ' Vgs']);
160 set(legend, ...
161     'Position', [0.215939049121225 0.582099669520804 0.116189725101397 ...
      0.286929172299835]);
162 title('MOSFET Family of Curves');
163 xlabel('Drain Voltage, Vds (V)');
164 ylabel('Drain Current, Id (A)');
165 mkdir([foldername]);
166 print('-dpng', [foldername, '\familyofcurves.png']);
167
168 %% Reset Keithley & Save Data
169 save([foldername, '/drainsweep.mat']);
170 fprintf(obj1, '*RST'); %command to reset the keithley
```

drainsweep_condplot.m

```
1 %% Sweeps Vgs, and conducts a Source-Drain Sweep for each fixed Vgs
2
3 clearvars -except obj1 testnum;
4 testnum=testnum+1;
5
6 cd '/Users/gfertig/Dropbox/Thesis/THz Project/Testing/20140318 Test ...
   Data - RTS'
7 foldername = [num2str(testnum), '-Spin2-Chip1-T2-DrainSweep'];
8 fprintf(obj1, '*RST'); %command to reset the Keithley
9 fprintf(obj1, 'smua.reset()'); % Reset SMU
10 fprintf(obj1, 'smub.reset()'); % Reset SMU
11 fprintf(obj1, 'errorqueue.clear()'); % Clear the error queue
12
13 fprintf(obj1, 'display.clear()');
14 fprintf(obj1, 'display.setText("$BTest in Progress$B)');
15 fprintf(obj1, 'beeper.beep(0.1, 500)'); %send beeper tone ...
   (duration,frequency)
16 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
   (duration,frequency)
17 fprintf(obj1, ['digio.writebit(2,0) ']); %Set digio bit 2 to High for ...
   shutter open
18
19 for j=1:15
20     vgs_bias = 0.2+0.1*j;
21     fprintf(obj1, ['vgs_bias=', num2str(vgs_bias)]);
22     fprintf(obj1, 'vdsstart=0');
23     fprintf(obj1, 'vdsstop=3');
24     fprintf(obj1, 'vdssteps=121');
25     vdssteps = 121;
26
27     % Global variables
28     fprintf(obj1, 'l.icmpl = 100E-3'); % Source compliance
29
30     % Shared local variables
31     fprintf(obj1, 'l.nplc = 1'); % Integration rate of measurement
32
33     % Default values and level check
34     fprintf(obj1, 'if (vdsstart == nil) then vdsstart = 0 end'); % Use ...
   default value
35     fprintf(obj1, 'if (vdsstart > 1) then vdsstart = 0.1 end'); % ...
   Coerce value
36     fprintf(obj1, 'if (vdsstop == nil) then vdsstop = 3 end'); % Use ...
   default value
```

```

37 fprintf(obj1, 'if (vdsstop > 3.3) then vdsstop = 3.3 end'); % ...
    Coerce value
38 fprintf(obj1, 'if (vdssteps == nil) then vdssteps = 50 end'); % Use ...
    default value
39 %fprintf(obj1, 'if (vdssteps > 100) then vdssteps = 100 end'); % ...
    Coerce value
40 fprintf(obj1, 'vdsstep = (vdsstop - vdsstart) / (vdssteps - 1)'); ...
    % step size
41 fprintf(obj1, 'vdssource_val = vdsstart'); % Source value during sweep
42 fprintf(obj1, 'l_i = 1'); % Iteration variable
43
44 fprintf(obj1, 'if (vgs.bias == nil) then vgs.bias = 0.5 end'); % ...
    Use default value
45 fprintf(obj1, 'if (vgs.bias > 3.3) then vgs.bias = 3.3 end'); % ...
    Coerce value
46
47 % Data tables
48 fprintf(obj1, 'vds = {}'); % Create data table for drain-source ...
    voltage
49 fprintf(obj1, 'id = {}'); % Create data table for drain-source current
50 fprintf(obj1, 'vgs = {}'); % Create data table for gate
51
52 % fprintf(obj1, 'smua.reset()'); % Reset SMU drain
53 % fprintf(obj1, 'smub.reset()'); % Reset SMU gate
54 % fprintf(obj1, 'errorqueue.clear()'); % Clear the error queue
55
56 % Configure (SMUA) source and measure settings
57 fprintf(obj1, 'smua.source.func = smua.OUTPUT_DCVOLTS');
58 fprintf(obj1, 'smua.source.autorangev = smua.AUTORANGE_ON'); % ...
    Enable source autorange
59 fprintf(obj1, 'smua.source.levelv = 0');
60 fprintf(obj1, 'smua.source.limiti = l_icmpl');
61 fprintf(obj1, 'smua.measure.autorangei = smua.AUTORANGE_ON'); % ...
    Enable measure autorange
62 %fprintf(obj1, 'smua.measure.rangei = 1E-6'); % Enable measure ...
    autorange
63 fprintf(obj1, 'smua.measure.autozero = smua.AUTOZERO_AUTO');
64 fprintf(obj1, 'smua.measure.nplc = l_nplc'); % Measurement ...
    integration rate
65 fprintf(obj1, 'smua.source.output = smua.OUTPUT_ON'); % Enable Output
66
67 % Configure SMUb gate and measure settings
68 fprintf(obj1, 'smub.source.func = smub.OUTPUT_DCVOLTS');
69 fprintf(obj1, 'smub.source.autorangev = smub.AUTORANGE_ON'); % ...
    Enable source autorange
70 fprintf(obj1, 'smub.source.levelv = 0');

```

C. MATLAB Test Scripts

```
71 fprintf(obj1, 'smub.source.limiti = l_icmpl');
72 fprintf(obj1, 'smub.measure.autorangev = smub.AUTORANGE_ON'); % ...
    Enable measure autorange
73 fprintf(obj1, 'smub.measure.autozero = smub.AUTOZERO_AUTO');
74 fprintf(obj1, 'smub.measure.nplc = l_nplc'); % Measurement ...
    integration rate
75 fprintf(obj1, 'smub.source.output = smub.OUTPUT_ON'); % Enable Output
76 fprintf(obj1, 'smub.source.levelv = vgs_bias');
77
78 % Execute sweep
79 fprintf(obj1, 'display.clear()');
80 fprintf(obj1, 'display.screen = 2');
81 fprintf(obj1, ['for l_i = 1,vdssteps do ',...
82     'if (l_i == 1) then ',...% Intialize start source value
83     'vdssource_val = vdsstart ',...
84     'end ' ,...
85     'vgs[l_i] = smub.measure.v() ',... % Measure Vgs
86     'id[l_i] = smua.measure.i() ',... % Measure Id
87     'vds[l_i] = smua.measure.v() ',...
88     'vdssource_val = vdssource_val + vdsstep ',... % Calculate new ...
    source value
89     'if (l_i == vdssteps) then ',... % Reinitialize voltage value ...
    after last iteration
90     'vdssource_val = vdsstart ',...
91     'end ' ,...
92     'smua.source.levelv = vdssource_val ',... % Increment source
93     'end ']);
94
95 fprintf(obj1, 'smua.source.output = smua.OUTPUT_OFF'); % Disable ...
    output
96 fprintf(obj1, 'smub.source.output = smub.OUTPUT_OFF'); % Disable ...
    output
97 fprintf(obj1, 'smua.source.levelv = 0'); % Return source to bias level
98 fprintf(obj1, 'smub.source.levelv = 0'); % Return source to bias level
99
100 fprintf(obj1, 'display.clear()');
101 fprintf(obj1, 'display.settext("$BSending Data$B")');
102
103 fprintf(obj1, 'l_i = 1'); % Reinitialize iteration variable
104
105 % Loop for data read
106 fprintf(obj1, 'print(vgs_bias)');
107 vgsbias(j) = str2num(fscanf(obj1)); %Read data from serial port and ...
    assign to var
108
109 for i=1:vdssteps %Loop to print all tables to matlab variables
```

```

110     fprintf(obj1, ['print(id[' , num2str(i), ']) \n' ] );
111     temp2 = fscanf(obj1);
112     temp2 = strtrim(temp2);
113     if strcmp(temp2, 'nil') == 1
114         id(j,i) = 0;
115     else
116         id(j,i)= str2num(temp2);
117     end
118
119     fprintf(obj1, ['print(vds[' , num2str(i), ']) \n' ] );
120     temp3 = fscanf(obj1);
121     temp3 = strtrim(temp3);
122     if strcmp(temp3, 'nil') == 1
123         vds(j,i) = 0;
124     else
125         vds(j,i)= str2num(temp3);
126     end
127 end
128 end
129
130 %%
131 for i = 1:length(vgsbias)
132 cond(i,:) = diff(id(i,:))./diff(vds(i,:));
133 resist(i,:) = 1./cond(i,:);
134 medianresist(i) = median(resist(i,:));
135 end
136
137 %% Print to Keithley screen and beep complete sequence
138 fprintf(obj1, 'display.clear()');
139 fprintf(obj1, 'display.settext("Test Complete")');
140 fprintf(obj1, 'beeper.beep(0.12, 500)'); %send beeper tone ...
141     (duration,frequency)
142 fprintf(obj1, 'beeper.beep(0.12, 800)'); %send beeper tone ...
143     (duration,frequency)
144 fprintf(obj1, 'beeper.beep(0.12, 1000)'); %send beeper tone ...
145     (duration,frequency)
146 fprintf(obj1, 'beeper.beep(0.12, 2000)'); %send beeper tone ...
147     (duration,frequency)
148
149 %% Plot Curves
150 figure();
151 plot (vds(3,:),id(3,:),vds(4,:),id(4,:),vds(5,:),id(5,:),vds(6,:),id(6,:),
152 vds(7,:),id(7,:),vds(8,:),id(8,:));
153 set(gca, 'fontsize',14);
154 set(gcf, 'color', 'w');
155 grid on;

```

```

152 legend([num2str(vgsbias(3)), ' Vgs'], [num2str(vgsbias(4)), ' ...
        Vgs'], [num2str(vgsbias(5)), ' Vgs'], [num2str(vgsbias(6)), ' ...
        Vgs'], [num2str(vgsbias(7)), ' Vgs'], [num2str(vgsbias(8)), ' Vgs']);
153 set(legend, ...
154     'Position', [0.215939049121225 0.582099669520804 0.116189725101397 ...
        0.286929172299835]);
155 title('MOSFET Family of Curves');
156 xlabel('Drain Voltage, Vds (V)');
157 ylabel('Drain Current, Id (A)');
158 mkdir([foldername]);
159 print('-dpng', [foldername, '\familyofcurves2.png']);
160
161 %% Plot Curves
162 figure();
163 plot (vds(1,:), id(1,:), vds(2,:), id(2,:), vds(3,:), id(3,:));
164 set(gca, 'fontsize', 14);
165 set(gcf, 'color', 'w');
166 grid on;
167 legend([num2str(vgsbias(1)), ' Vgs'], [num2str(vgsbias(2)), ' ...
        Vgs'], [num2str(vgsbias(3)), ' Vgs']);
168 set(legend, ...
169     'Position', [0.215939049121225 0.582099669520804 0.116189725101397 ...
        0.286929172299835]);
170 title('MOSFET Family of Curves');
171 xlabel('Drain Voltage, Vds (V)');
172 ylabel('Drain Current, Id (A)');
173 mkdir([foldername]);
174 print('-dpng', [foldername, '\familyofcurves.png']);
175
176 %% Single Conductance
177 marker = 3;
178 figure();
179 plot (vds(2,1:length(vds)-1), cond(marker,:), '-*');
180 hold on;
181 hline = reffline(0, median(cond(marker, :)));
182 set(hline, 'Color', 'r')
183 set(gca, 'fontsize', 14);
184 set(gcf, 'color', 'w');
185 grid on;
186 %xlim([0 2]);
187 legend(['Conductance @ Vgs = ', num2str(vgsbias(marker)), ' V'], ['Median ...
        = ', num2str(median(cond(marker, :)))]);
188 title('Conductance of the Channel');
189 ylabel('Channel Conductance (dId/dVds)');
190 xlabel('Drain Voltage, Vds (V)');
191 print('-dpng', [foldername, '\conductance1.png']);

```

```

192
193 %% Single Resistance
194 figure();
195 plot(vds(2,1:length(vds)-1), resist(marker,:), '-*');
196 hold on;
197 hline = reffline(0,median(resist(marker,:)));
198 set(hline, 'Color', 'r')
199 set(gca, 'fontsize', 14, 'yscale', 'log');
200 set(gcf, 'color', 'w');
201 grid on;
202 %xlim([0 2]);
203 legend(['Resistance @ Vgs = ', num2str(vgsbias(marker)), ' V'], ['Median = ...
    ', num2str(median(resist(marker,:)))]);
204 title('Resistance of the Channel');
205 ylabel('Channel Resistance (Ohms) (dVds/dId)');
206 xlabel('Drain Voltage, Vds (V)');
207 print('-dpng', [foldername, '\resistance1.png']);
208
209 %% Median Resistance vs Vgs
210 figure();
211 plot(vgsbias, medianresist, '-*');
212 hold on;
213 %hline = reffline(0,median(resist(marker,:)));
214 %set(hline, 'Color', 'r')
215 set(gca, 'fontsize', 14, 'yscale', 'log');
216 set(gcf, 'color', 'w');
217 xlabel('Gate Voltage, Vgs (V)');
218 ylabel('Median Resistance of Channel (Ohms)');
219 title('Median Resistance of Channel vs. Vgs');
220 grid on;
221 print('-dpng', [foldername, '\medianresist.png']);
222
223 %% Multi Resistance
224 figure();
225 x=vds(2,1:length(vds)-1);
226 plot(x, resist(1,:), '-*', x, resist(2,:), '-*', x, resist(3,:), '-*', x, ...
    resist(5,:), '-*', x, resist(10,:), '-*', x, resist(15,:), '-*');
227 hold on;
228 set(gca, 'fontsize', 14, 'yscale', 'log');
229 set(gcf, 'color', 'w');
230 grid on;
231 legend(['Vgs = ', num2str(vgsbias(1))], ['Vgs = ...
    ', num2str(vgsbias(2))], ['Vgs = ', num2str(vgsbias(3))], ['Vgs = ...
    ', num2str(vgsbias(5))], ['Vgs = ', num2str(vgsbias(10))], ['Vgs = ...
    ', num2str(vgsbias(15))])'
232 title('Resistance of the Channel');

```

```
233 ylabel('Channel Resistance (Ohms) (dVds/dId)');
234 xlabel('Drain Voltage, Vds (V)');
235 print('-dpng', [foldername, '\multiresist.png']);
236
237 %% Reset Keithley & Save Data
238 save([foldername, '/drainsweep.mat']);
239 fprintf(obj1, '*RST'); %command to reset the keithley
```

C.6. Terahertz Detection Scripts

fixedId_sweepvgs_measurevds.m

```
1 %% Chopper Detect
2 % This script sweeps the gate voltage for a fixed drain current bias and
3 % measures the source drain voltage.
4
5 clearvars -except obj1 k testnum;
6 testnum=testnum+1;
7
8 % Timing Variables
9 cd '/Users/gfertig/Dropbox/Thesis/THz Project/Testing/20140227 Test Data'
10 foldername = ['URFri/', num2str(testnum), '-BT-T5-S-RTSNoise-NoShutter'];
11 chopperfreq = 30; %Hz - For plots
12 nplc = .1; % Integration rate of measurement 1,.1,.01,.001 (0.001 finest)
13 measnum = 3000; %Number of submeasurements per set (gate gate voltage ...
    instance)
14 measinterval = 0.000; %Keithley delay BETWEEN submeasurements
15 measdelay = 0.000; %Keithley delay BEFORE a set of measurements occurs
16 shutterdelay = 0.000; % Delay (s) for shutter full open
17
18 % Matlab Variables
19 idbias = 6E-6; % (A) Drain Bias Current
20 vgssteps = 1; %Number of steps for Vgs sweep %Max ...
    vgssteps*measnum=10000 for measnum=1000 is 10
21 vgsstart = 0.62; %Start Voltage for Vgs sweep
22 vgsstop = 0.62; %Stop Voltage for Vgs sweep
23 stepsize = (vgsstop - vgsstart) / (vgssteps); %Calculate sweep step size
24
25 vrangeg = 6; %gate
26 vranged = 6; %Voltage source range: 100mV, 1V, 6V, 40V
27 irange = 100E-6; % Current source range: 100nA, 1uA, 100uA, 1mA, 10mA, ...
    100mA, 1A, 3A
28 %irange = 100E-9; % Current source range: 100nA, 1uA, 100uA, 1mA, 10mA, ...
    100mA, 1A, 3A
29 vlimit = 40; %Voltage limit
30 ilimit = 10E-3; %Current limit
31
32 buffappend = 1; %'0' tells the buffers to overwrite for each set of ...
    measurements, '1' appends
33 filespecifier = ...
    ['CH', num2str(chopperfreq), '_STP', num2str(vgssteps), '_MEAS',
34 num2str(measnum), '_NPLC', num2str(nplc), '_Id', num2str(idbias)];
```

C. MATLAB Test Scripts

```
35 %% Reset Keithley
36 fprintf(obj1, '*RST'); %command to reset the keithley
37 fprintf(obj1, 'smua.reset()'); % Reset SMU
38 fprintf(obj1, 'smub.reset()'); % Reset SMU
39 fprintf(obj1, 'errorqueue.clear()'); % Clear the error queue
40
41 %Display and Beep Sequence
42 fprintf(obj1, 'display.clear()');
43 fprintf(obj1, 'display.settext("$BTest in Progress$B")');
44 fprintf(obj1, 'beeper.beep(0.1, 500)'); %send beeper tone ...
    (duration,frequency)
45 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
46
47 %Digio reset
48 fprintf(obj1, ['digio.writebit(2,1) ']); %Set digio bit 2 to High for ...
    shutter open
49 pause(shutterdelay);
50 fprintf(obj1, ['digio.writebit(2,0) ']); %Set digio bit 2 to Low for ...
    shutter close
51
52
53 %% Configure Sources
54 % Source A Configuration - Drain
55 fprintf(obj1, ['smua.source.func = smua.OUTPUT_DCAMPS']); %Select ...
    current source function
56 fprintf(obj1, ['smua.source.rangei = ',num2str(irange)]); % Set source ...
    function range (lowest for Voltmeter)
57 fprintf(obj1, ['smua.source.limiti = ',num2str(ilimit)]); %Set current ...
    limit
58 fprintf(obj1, ['smua.source.limitv = ',num2str(vlimit)]); %Set voltage ...
    limit
59 fprintf(obj1, ['smua.source.leveli = 0']); %Set source current to 0 to ...
    start
60
61 % Source B Configuration - Gate
62 fprintf(obj1, ['smub.source.func = smub.OUTPUT_DCVOLTS']);
63 fprintf(obj1, ['smub.source.rangev = ',num2str(vrangep)]); % Set ...
    source range
64 fprintf(obj1, ['smub.source.limitv = ',num2str(vlimit)]); %Set voltage ...
    limit
65 fprintf(obj1, ['smub.source.limiti = ',num2str(ilimit)]); %Setcurrent limit
66 fprintf(obj1, ['smub.source.levelv = 0']); %Set source voltage to 0 to ...
    start
67
68 %% Configure Measurements & Buffers
```

C. MATLAB Test Scripts

```
69
70 % SMUA Measurement Settings - Drain (Measuring Voltage)
71 fprintf(obj1, 'smua.measure.autozero = smua.AUTOZERO_ONCE');
72 fprintf(obj1, ['smua.measure.rangev = ', num2str(vranged)]); % Set ...
    Voltmeter range
73 fprintf(obj1, ['smua.measure.nplc = ', num2str(nplc)]); % Measurement ...
    integration rate
74 fprintf(obj1, ['smua.measure.count = ', num2str(measnum)]); % Number of ...
    measurements to collect.
75 fprintf(obj1, ['smua.measure.delay = ', num2str(measdelay)]); % Set the ...
    delay before the first measurement
76 fprintf(obj1, ['smua.measure.interval = ', num2str(measinterval)]); % ...
    Set the delay between measurements
77
78 %SMUA Buffer 1 - Stores drain voltage
79 fprintf(obj1, 'smua.nvbuffer1.clear() '); % Clears the buffer
80 fprintf(obj1, 'smua.nvbuffer1.collecttimestamps = 1 '); % Enables ...
    timestamp collect
81 fprintf(obj1, ['smua.nvbuffer1.appendmode = ', num2str(buffappend)]); ...
    %Overwrites previous measurements in buffer
82 fprintf(obj1, 'smua.nvbuffer1.timestampresolution = 0.000001 '); %Sets ...
    timestamp resolution to the lus (finest)
83
84 %SMUA Buffer 2 - Stores drain current
85 fprintf(obj1, 'smua.nvbuffer2.clear() '); % Clears the buffer
86 fprintf(obj1, 'smua.nvbuffer2.collecttimestamps = 1 '); % Enables ...
    timestamp collect
87 fprintf(obj1, ['smua.nvbuffer2.appendmode = ', num2str(buffappend)]); ...
    %Overwrites previous measurements in buffer
88 fprintf(obj1, 'smua.nvbuffer2.timestampresolution = 0.000001 '); %Sets ...
    timestamp resolution to the lus (finest)
89
90 %SMUB Measurement Settings - Gate (Measuring Voltage)
91 fprintf(obj1, 'smub.measure.autozero = smub.AUTOZERO_ONCE');
92 fprintf(obj1, ['smub.measure.nplc = ', num2str(nplc)]); % Measurement ...
    integration rate
93 fprintf(obj1, ['smub.measure.rangev = ', num2str(vranged)]); % Set ...
    measure range
94 fprintf(obj1, ['smub.measure.count = ', num2str(measnum)]); % Number of ...
    measurements to collect.
95 fprintf(obj1, ['smub.measure.delay = ', num2str(measdelay)]); % Set the ...
    delay before the first measurement
96 fprintf(obj1, ['smub.measure.interval = ', num2str(measinterval)]); % ...
    Set the delay between measurements
97
98 %SMUB Buffer 1
```

C. MATLAB Test Scripts

```
99 fprintf(obj1, 'smub.nvbuffer1.clear() '); % Clears the buffer
100 fprintf(obj1, 'smub.nvbuffer1.collecttimestamps = 1 '); % Enables ...
    timestamp collect
101 fprintf(obj1, ['smub.nvbuffer1.appendmode = ', num2str(buffappend)]); ...
    %Overwrites previous measurements in buffer
102 fprintf(obj1, 'smub.nvbuffer1.timestampresolution = 0.000001 '); %Sets ...
    timestamp resolution to the 1us (finest)
103
104
105 %% Execute sweep
106 fprintf(obj1, 'display.clear()');
107 fprintf(obj1, 'display.screen = 2');
108
109 vgsloop = vgsstart;
110 fprintf(obj1, ['vgsloop = ', num2str(vgsstart)]);
111 fprintf(obj1, ['stepsize = ', num2str(stepsize)]);
112 fprintf(obj1, ['loopend = 0']);
113 fprintf(obj1, ['loopcheck = ', num2str(vgsstop-stepsize)]);
114
115 fprintf(obj1, ['smua.source.output = smua.OUTPUT_ON ']); ...
    %Measure w/shutter closed
116 fprintf(obj1, ['smub.source.output = smub.OUTPUT_ON ']); ,...
    %Enable Output
117 fprintf(obj1, ['smub.source.levelv = vgsloop ']); %Set gate voltage
118 fprintf(obj1, ['smua.source.leveli = ', num2str(idbias), ' ']); %Set ...
    drain bias voltage
119 fprintf(obj1, ['delay(2) ']); ,...
    %Enable Output
120
121 fprintf(obj1, ['for i = 1,', num2str(vgssteps), 'do ', ...
122     'smub.source.levelv = vgsloop ', ... %Set gate voltage
123     'smua.source.leveli = ', num2str(idbias), ' ', ...
    %Set drain bias voltage
124     'smua.source.output = smua.OUTPUT_ON ', ...
    %Measure w/shutter closed
125     'smub.source.output = smub.OUTPUT_ON ', ...
    %Enable Output
126     'waitcomplete() ', ...
127     'smub.measure.v(smub.nvbuffer1) ', ...
    %Measure gate voltage
128     'smua.measure.v(smua.nvbuffer1) ', ...
    %Measure drain voltage
129     'smua.measure.i(smua.nvbuffer2) ', ...
    %Measure drain current
130     'waitcomplete() ', ...
    %Measure w/shutter open
```

C. MATLAB Test Scripts

```
131         'digio.writebit(2,1) ',...           ...
           %Set digio bit 2 to High for shutter open
132         'delay(',num2str(shutterdelay),') ',...
133         'smub.measure.v(smub.nvbuffer1) ',...           ...
           %Measure gate voltage
134         'smua.measure.v(smua.nvbuffer1) ',...           ...
           %Measure drain voltage
135         'smua.measure.i(smua.nvbuffer2) ',...           ...
           %Measure drain current
136         'waitcomplete() ',...
137         'smua.source.output = smua.OUTPUT_OFF ',...           ...
           %Disable output
138         'smub.source.output = smub.OUTPUT_OFF ',...           ...
           %Disable output
139         'digio.writebit(2,0) ',...           ...
           %Set digio bit 2 to Low for shutter close
140         'vgsloop = vgsloop + stepsize ',...           ...
           %Increment loop
141         'if (vgsloop == loopcheck) then loopend = 1 end ',...
142     'end ']);
143 loopend = 0;
144 while loopend == 0
145     fprintf(obj1, 'print(loopend)'); %check the loopend variable
146     loopend = fscanf(obj1);
147     %pause(.1);
148 end
149 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
150
151 %% Retrieve data from keithley
152 fprintf(obj1, 'display.clear()');
153 fprintf(obj1, 'display.settext("$BSending Data$B")');
154
155 fprintf(obj1, 'print(smub.nvbuffer1.basetimestamp)'); % print the ...
    timestamp of the buffer
156 temp = fscanf(obj1);
157 vgsbuffbasetime=str2num(temp);
158 clear temp;
159 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
160
161 fprintf(obj1, 'printbuffer(1,smub.nvbuffer1.n,smub.nvbuffer1.readings)');
162 temp = fscanf(obj1);
163 vgsbuffmeas = cell2mat(textscan(temp, '%f64', 'delimiter', ','));
164 clear temp;
```

```

165 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
166
167 fprintf(obj1, 'printbuffer(1,smub.nvbuffer1.n,smub.nvbuffer1.timestamps)');
168 temp = fscanf(obj1);
169 vgsbuffftime = cell2mat(textscan(temp, '%f64', 'delimiter', ','));
170 clear temp;
171 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
172
173 fprintf(obj1, 'print(smua.nvbuffer1.basetimestamp)');
174 temp = fscanf(obj1);
175 vdsbuffbasetime=str2num(temp);
176 clear temp;
177 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
178
179 fprintf(obj1, 'printbuffer(1,smua.nvbuffer1.n,smua.nvbuffer1.readings)');
180 temp = fscanf(obj1);
181 vdsbuffmeas = cell2mat(textscan(temp, '%f64', 'delimiter', ','));
182 clear temp;
183 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
184
185 fprintf(obj1, 'printbuffer(1,smua.nvbuffer1.n,smua.nvbuffer1.timestamps)');
186 temp = fscanf(obj1);
187 vdsbuffftime = cell2mat(textscan(temp, '%f64', 'delimiter', ','));
188 clear temp;
189 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
190
191 fprintf(obj1, 'print(smua.nvbuffer2.basetimestamp)');
192 temp = fscanf(obj1);
193 idsbuffbasetime=str2num(temp);
194 clear temp;
195 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
196
197 fprintf(obj1, 'printbuffer(1,smua.nvbuffer2.n,smua.nvbuffer2.readings)');
198 temp = fscanf(obj1);
199 idsbuffmeas = cell2mat(textscan(temp, '%f64', 'delimiter', ','));
200 clear temp;
201 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
202
203 fprintf(obj1, 'printbuffer(1,smua.nvbuffer2.n,smua.nvbuffer2.timestamps)');

```

C. MATLAB Test Scripts

```
204 temp = fscanf(obj1);
205 idsbufftime = cell2mat(textscan(temp, '%f64', 'delimiter', ','));
206 clear temp;
207 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
208
209 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
210
211 %% Print to Keithley screen and beep complete sequence
212 fprintf(obj1, 'display.clear()');
213 fprintf(obj1, 'display.settext("Test Complete")');
214 fprintf(obj1, 'beeper.beep(0.12, 500)'); %send beeper tone ...
    (duration,frequency)
215 fprintf(obj1, 'beeper.beep(0.12, 800)'); %send beeper tone ...
    (duration,frequency)
216 fprintf(obj1, 'beeper.beep(0.12, 1000)'); %send beeper tone ...
    (duration,frequency)
217 fprintf(obj1, 'beeper.beep(0.12, 2000)'); %send beeper tone ...
    (duration,frequency)
218
219 %% Data Reduction
220 vgsmeastime = vgsbufftime+vgsbuffbasetime; %Add the basetime to each ...
    measurement time
221 vdsmeastime = vdsbufftime+vdsbuffbasetime;
222 idsmeastime = idsbufftime+idsbuffbasetime;
223
224 %take the buffers, average and split
225 vgsbuffmeas(1) = vgsbuffmeas(2); % Replace first entry due to Keithley ...
    error on first measurement
226 vdsbuffmeas(1) = vdsbuffmeas(2);
227 idsbuffmeas(1) = idsbuffmeas(2);
228
229 %Reduce data for open and closed shutter measurements
230 %creates and open and closed vector from the read buffer which correspond
231 %to each pair of measurements.
232 j=1;
233 for i=1:vgssteps
234     vgsclosed(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        vgsbuffmeas((j*measnum)-measnum+1):(j*measnum));
235     vdsclsd(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        vdsbuffmeas((j*measnum)-measnum+1):(j*measnum));
236     idsclosed(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        idsbuffmeas((j*measnum)-measnum+1):(j*measnum));
237     vgsclsdtime(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        vgsbufftime((j*measnum)-measnum+1):(j*measnum));
```

C. MATLAB Test Scripts

```
238     vdsclsdtime(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        vdsbufftime(((j*measnum)-measnum+1):(j*measnum));
239     idsclsdtime(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        idsbufftime(((j*measnum)-measnum+1):(j*measnum));
240     j=j+1;
241
242     vgsopen(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        vgsbuffmeas(((j*measnum)-measnum+1):(j*measnum));
243     vdsopen(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        vdsbuffmeas(((j*measnum)-measnum+1):(j*measnum));
244     idsopen(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        idsbuffmeas(((j*measnum)-measnum+1):(j*measnum));
245     vgsopentime(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        vgsbufftime(((j*measnum)-measnum+1):(j*measnum));
246     vdsopentime(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        vdsbufftime(((j*measnum)-measnum+1):(j*measnum));
247     idsopentime(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        idsbufftime(((j*measnum)-measnum+1):(j*measnum));
248     j=j+1;
249 end
250 vgsdiff = vgsopen-vgsclsd;
251 vdsdiff = vdsopen-vdsclsd;
252 idsdiff = idsopen-idsclsd;
253
254 %FFT of signals
255 for i=1:measnum-1
256     clsdtimediff(i)=vdsclsdtime(i+1)-vdsclsdtime(i);
257 end
258 NFFT = 2^nextpow2(measnum); % Next power of 2 from length of y
259 Fs = 1/mean(clsdtimediff);
260 f = (0:NFFT/2-1)*Fs/NFFT;
261 for i=1:vgsteps
262     tempfft = fft(vdsclsd(((i-1)*measnum+1):(i*measnum))-mean(vdsclsd(
263 ((i-1)*measnum+1):(i*measnum))),NFFT)/NFFT; %Mean subtracted
264     tempfft = tempfft(1:NFFT/2);
265     vdsclsdfft(i,:) = abs(tempfft);
266     clear tempfft;
267     tempfft = fft(vdsopen(((i-1)*measnum+1):(i*measnum))-mean(vdsopen(
268 ((i-1)*measnum+1):(i*measnum))),NFFT)/NFFT; %Mean subtracted
269     tempfft = tempfft(1:NFFT/2);
270     vdsopenfft(i,:) = abs(tempfft);
271     vdsfftdiff(i,:) = vdsopenfft(i,:)-vdsclsdfft(i,:);
272     clear tempfft;
273 end
274 [fftmaxdiff(1,:),fftmaxdiff(2,:)] = max(vdsopenfft(:,25:NFFT/2), [], 2);
275 fftmaxdiff(2,:)=fftmaxdiff(2,:)+25;
```

C. MATLAB Test Scripts

```
276 %if measnum > 1
277     for i=1:(vgssteps)
278         vgsavgmeasclosed(i) = ...
                mean(vgsclosed((i*measnum)-(measnum-1)):i*measnum)); ...
                %Averages the submeasurements for each vgs step
279         vdsavgmeasclosed(i) = ...
                mean(vdsclosed((i*measnum)-(measnum-1)):i*measnum));
280         idsavgmeasclosed(i) = ...
                mean(idsclosed((i*measnum)-(measnum-1)):i*measnum));
281         vgsavgmeasopen(i) = ...
                mean(vgsopen((i*measnum)-(measnum-1)):i*measnum)); ...
                %Averages the submeasurements for each vgs step
282         vdsavgmeasopen(i) = ...
                mean(vdsopen((i*measnum)-(measnum-1)):i*measnum));
283         idsavgmeasopen(i) = ...
                mean(idsopen((i*measnum)-(measnum-1)):i*measnum));
284         vgsavgmeasdiff(i) = ...
                mean(vgsdiff((i*measnum)-(measnum-1)):i*measnum)); ...
                %Averages the submeasurements for each vgs step
285         vdsavgmeasdiff(i) = ...
                mean(vdsdiff((i*measnum)-(measnum-1)):i*measnum));
286         idsavgmeasdiff(i) = ...
                mean(idsdiff((i*measnum)-(measnum-1)):i*measnum));
287     end
288 %end
289 %% Plots
290 mkdir([foldername]);
291 %Plots the average for each series of submeasurements
292 figure(1);
293 idvar = 1E-7;
294 [AX,H1,H2] = ...
        plotyy(vgsavgmeasclosed,vdsavgmeasclosed,vgsavgmeasopen,idsavgmeasopen);
295 set(AX(1),'fontsize',14,'Position',[0.12 0.17 0.72 0.72]);
296 set(AX(2),'xtick',[],'YLim',[(idbias-idvar) (idbias+idvar)],'fontsize',14);
297 set(H1,'marker','*');
298 set(H2,'marker','*');
299 set(get(AX(1),'Ylabel'),'String','Avg Source-Drain Voltage, Vds ...
        (V)','fontsize',14);
300 set(get(AX(2),'Ylabel'),'String','Avg Source-Drain Current, Id ...
        (A)','fontsize',14);
301 set(gcf,'color','w');
302 %xlim([vgsstart vgsstop]);
303 title(['Vgs Sweep (Avg of ',num2str(measnum),' Measurements; Id = ...
        ',num2str(idbias),' A); Shutter Closed']);
304 xlabel('Gate Voltage, Vgs (V)');
305 grid on;
```

C. MATLAB Test Scripts

```
306 print('-dpng', [foldername, '\', filespecifier, '-avgsclosed.png']);
307 %%
308 %Plots the average for each series of submeasurements
309 figure(2);
310 [AX,H1,H2] = ...
    plotyy(vgsavgmeasopen,vdsavgmeasopen,vgsavgmeasopen,idsavgmeasopen);
311 set(AX(1), 'fontsize', 14, 'Position', [0.12 0.17 0.72 0.72]);
312 set(AX(2), 'xtick', [], 'YLim', [(idbias-idvar) (idbias+idvar)], 'fontsize', 14);
313 set(H1, 'marker', '*');
314 set(H2, 'marker', '*');
315 set(get(AX(1), 'Ylabel'), 'String', 'Avg Source-Drain Voltage, Vds ...
    (V)', 'fontsize', 14);
316 set(get(AX(2), 'Ylabel'), 'String', 'Avg Source-Drain Current, Id ...
    (A)', 'fontsize', 14);
317 set(gcf, 'color', 'w');
318 title(['Vgs Sweep (Avg of ', num2str(measnum), ' Measurements; Id = ...
    ', num2str(idbias), ' A); Shutter Open']);
319 xlabel('Gate Voltage, Vgs (V)');
320 grid on;
321 print('-dpng', [foldername, '\', filespecifier, '-avgsoopen.png']);
322 %%
323 %Plots vds measurements vs time; chopper overlaid for reference
324 figure(3);
325 t=vdsmeastime(1):.0001:vdsmeastime(length(vdsmeastime));
326 chopperwave = ...
    mean(vdsbuffmeas)+0.2*max(vdsbuffmeas)*square(2*pi*chopperfreq*t);
327 %plot(vdsmeastime,vdsbuffmeas,'*',t,chopperwave);
328 plot(vdsmeastime,vdsbuffmeas,'-*');
329 xlabel('Time (s)');
330 ylabel('Source-Drain Voltage (V)');
331 title('Source-Drain Voltage (V) vs Time (s)');
332 set(gca, 'fontsize', 14);
333 set(gcf, 'color', 'w');
334 grid on;
335 print('-dpng', [foldername, '\', filespecifier, '-vds-time.png']);
336 %%
337 %Plots ids measurements vs time; chopper overlaid for reference
338 % figure(4);
339 % t=idsmeastime(1):.0001:idsmeastime(length(idsmeastime));
340 % chopperwave = ...
    mean(idsbuffmeas)+0.1*mean(idsbuffmeas)*square(2*pi*chopperfreq*t);
341 % %plot(idsmeastime,idsbuffmeas,'*',t,chopperwave);
342 % plot(idsmeastime,idsbuffmeas,'*');
343 % set(gca, 'fontsize', 14);
344 % set(gcf, 'color', 'w');
345 % grid on;
```

```

346 % xlabel('Time (s)');
347 % ylabel('Source-Drain Current (A)');
348 % title('Source-Drain Current (A) vs Time (s)');
349 % print('-dpng', [foldername, '\', filespecifier, '-ids-time.png']);
350
351 %%
352 %Plots vds difference vs vgs;
353 figure(5);
354 plot(vgsopen, vdsdiff, '*');
355 set(gca, 'fontsize', 14);
356 set(gcf, 'color', 'w');
357 xlabel('Gate Voltage, Vgs (V)');
358 ylabel('Source-Drain Voltage Difference (V)');
359 title('Source-Drain Voltage Difference (V) vs Vgs (s)');
360 grid on;
361 print('-dpng', [foldername, '\', filespecifier, '-vdsdiffvgs.png']);
362 %%
363 %Plots vds difference vs time;
364 % figure(6);
365 % plot(vdsopentime, vdsdiff, '*');
366 % set(gca, 'fontsize', 14);
367 % set(gcf, 'color', 'w');
368 % xlabel('Time (s)');
369 % ylabel('Source-Drain Voltage Difference (V)');
370 % title('Source-Drain Voltage Difference (V) vs Time');
371 % grid on;
372 % print('-dpng', [foldername, '\', filespecifier, '-vdsdifftime.png']);
373 %%
374 %Plots avg vds difference vs vgs;
375 figure(7);
376 plot(vgsavgmeasopen, vdsavgmeasdiff, '-*');
377 set(gca, 'fontsize', 14);
378 set(gcf, 'color', 'w');
379 xlabel('Gate Voltage, Vgs (V)');
380 ylabel('Source-Drain Voltage Difference (V)');
381 title('Avg Source-Drain Voltage Difference (V) vs Vgs (V)');
382 grid on;
383 %print('-dpng', [foldername, '\', filespecifier, '-vdsavgdiffvgs.png']);
384 print('-dpng', [foldername, '\vdsavgdiff.png']);
385 %%
386 %Plots fft of vds open and closed;
387 figure(8);
388 plot(f(2:length(f)), vdslosedfft(2:length(vdslosedfft)), f(2:length(f)),
389 vdsopenfft(2:length(vdsopenfft))); %No DC
390 set(gca, 'fontsize', 14);
391 set(gcf, 'color', 'w');

```

```

392 legend('Shutter Closed','Shutter Open');
393 xlabel('Frequency (Hz)');
394 ylabel('|Vds|');
395 title('FFT of Vds');
396 grid on;
397 %ylim([-0E-3 1E-3]);
398 %xlim([0 50]);
399 print('-dpng', [foldername, '\', filespecifier, '-vdsfft.png']);
400 %%
401 %Plots fft of vds open and closed;
402 figure(9);
403 clf(9);
404 cc=hsv(vgssteps);
405 hold on;
406 for i=1:vgssteps
407     plot(f(2:length(f)),vdsfftdiff(i,2:length(vdsfftdiff)), 'color',cc(i,:)); ...
         %No DC
408 end
409 hold off;
410 set(gca, 'fontsize',14);
411 set(gcf, 'color', 'w');
412 xlabel('Frequency (Hz)');
413 ylabel('|Vds|');
414 title('Difference of FFT Open/Closed Shutter of Vds');
415 grid on;
416 %ylim([-0E-3 1E-5]);
417 %xlim([825 875]);
418 legendmatrix=cellstr(num2str(vgsavgmeasopen(1:vgssteps)', 'Vgs = %f'));
419 legend(legendmatrix);
420 print('-dpng', [foldername, '\', filespecifier, '-vdsfftdiff.png']);
421 %%
422 %Plots fft of vds open and closed;
423 figure(10);
424 plot(vgsavgmeasopen,fftmaxdiff(1,:), '-*');
425 labels = cellstr( num2str(fftmaxdiff(2,:)) ); %' # labels correspond ...
         to their order
426 text(vgsavgmeasopen,fftmaxdiff(1,:), labels, ...
         'VerticalAlignment','bottom', ...
         'HorizontalAlignment','right')
427
428 set(gca, 'fontsize',14);
429 set(gcf, 'color', 'w');
430 xlabel('Vgs (V)');
431 ylabel('Max (|Vds|)');
432 title('Maximum of Vds FFT Difference vs Gate Voltage (Max Freq Noted)');
433 grid on;
434 print('-dpng', [foldername, '\', filespecifier, '-vdsfftmaxdiff.png']);

```

C. MATLAB Test Scripts

```
435 %% Make a wave to compare expected fft
436 t=vdscllosedtime(1):1/Fs:vdscllosedtime(length(vdscllosedtime));
437 %chopperwave = ...
    mean(vdscllosed)+range(vdscllosed)*square(2*pi*chopperfreq*t); %shifted
438 chopperwave = square(2*pi*chopperfreq*t); %shifted
439 NFFT = 2^nextpow2(length(chopperwave)); % Next power of 2 from length ...
    of y
440 %Fs = chopperfreq;
441 f = (0:NFFT/2-1)*Fs/NFFT;
442 sqwvfft = fft(chopperwave,NFFT)/NFFT;
443 sqwvfft = sqwvfft(1:NFFT/2);
444 sqwvfft = abs(sqwvfft);
445 figure(11);
446 subplot(2,1,1); plot(t,chopperwave); xlim([0 1/chopperfreq]); ...
    xlabel('Time (s)');
447 subplot(2,1,2); plot(f(2:length(f)),sqwvfft(2:length(sqwvfft))); ...
    xlabel('Freq (Hz)');%No DC
448 print('-dpng', [foldername,'\',filespecifier,'-synfft.png']);
449 %% Reset Keithley & Save Data
450 save([foldername,'\',filespecifier,'.mat']);
451 fprintf(obj1, '*RST'); %command to reset the keithley
```

fixedvgs_sweepvds_measureId.m

```
1 %% Fixed Vgs Sweep Vds - Source Gate Voltage and Drain Voltage, Measure ...
   Drain Current
2 % This script sweeps the drain voltage for a fixed gate voltage bias and
3 % measures the source drain current.
4
5 %close all;
6 %for z=1:10
7 clearvars -except obj1 k testnum;
8 testnum=testnum+1;
9
10 % Timing Variables
11 cd '/Users/gfertig/Dropbox/Thesis/THz Project/Testing/20140307 Test Data'
12 foldername = [num2str(testnum), '-Old-Chip2-T2-B-Vdsswp-Paul'];
13 chopperfreq = 1; %Hz - For plots
14 nplc = .1; % Integration rate of measurement 1, .1, .01, .001 (0.001 finest)
15 measnum = 100; %Number of submeasurements per set ()
16 measinterval = 0.000; %Keithley delay BETWEEN submeasurements
17 measdelay = 0.000; %Keithley delay BEFORE a set of measurements occurs
18 shutterdelay = 0.000; % Delay (s) for shutter full open
19
20 % Gate Loop Variables
21 vgsbias = .48; %Fixed bias voltage for Vds
22 vdssteps = 25; %Number of steps for Vgs sweep %Max ...
   vgssteps*measnum=10000 for measnum=1000 is 10
23 vdsstart = 0; %Start Voltage for Vgs sweep
24 vdsstop = 3; %Stop Voltage for Vgs sweep
25 stepsize = (vdsstop - vdsstart) / (vdssteps); %Calculate sweep step size
26
27 vrange = 6; %Voltage source range: 100mV, 1V, 6V, 40V
28 irange = 100E-6; % Current source range: 100nA, 1uA, 100uA, 1mA, 10mA, ...
   100mA, 1A, 3A
29 %irange = 100E-9; % Current source range: 100nA, 1uA, 100uA, 1mA, 10mA, ...
   100mA, 1A, 3A
30 vlimit = 10; %Voltage limit
31 ilimit = 10E-3; %Current limit
32
33 buffappend = 1; %'0' tells the buffers to overwrite for each set of ...
   measurements, '1' appends
34 filespecifier = ...
   ['MeasId.SwpVds-CH', num2str(chopperfreq), '_MEAS', num2str(measnum), '_NPLC',
35 num2str(nplc), '_VgsBias', num2str(vgsbias)];
36 %% Reset Keithley
37 fprintf(obj1, '*RST'); %command to reset the keithley
```

C. MATLAB Test Scripts

```
38 fprintf(obj1, 'smua.reset()'); % Reset SMU
39 fprintf(obj1, 'smub.reset()'); % Reset SMU
40 fprintf(obj1, 'errorqueue.clear()'); % Clear the error queue
41
42 %Display and Beep Sequence
43 fprintf(obj1, 'display.clear()');
44 fprintf(obj1, 'display.settext("$BTest in Progress$B")');
45 fprintf(obj1, 'beeper.beep(0.1, 500)'); %send beeper tone ...
    (duration,frequency)
46 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
47
48 %Digio reset
49 fprintf(obj1, ['digio.writebit(2,1) ']); %Set digio bit 2 to High for ...
    shutter open
50 pause(shutterdelay);
51 fprintf(obj1, ['digio.writebit(2,0) ']); %Set digio bit 2 to Low for ...
    shutter close
52 %% Configure Sources
53 % Source A Configuration - Drain
54 fprintf(obj1, ['smua.source.func = smua.OUTPUT_DCVOLTS']); %Select ...
    voltage source
55 fprintf(obj1, ['smua.source.rangev = ',num2str(vrange)]); % Set source ...
    function range
56 fprintf(obj1, ['smua.source.limiti = ',num2str(ilimit)]); %Set current ...
    limit
57 fprintf(obj1, ['smua.source.limitv = ',num2str(vlimit)]); %Set voltage ...
    limit
58 fprintf(obj1, ['smua.source.levelv = 0']); %Set source voltage to 0 to ...
    start
59
60 % Source B Configuration - Gate
61 fprintf(obj1, ['smub.source.func = smub.OUTPUT_DCVOLTS']);
62 %fprintf(obj1, ['smub.source.autorangev = smub.AUTORANGE_ON']); % ...
    Enable source autorange
63 fprintf(obj1, ['smub.source.rangev = ',num2str(vrange)]); % Set source ...
    range
64 fprintf(obj1, ['smub.source.limitv = ',num2str(vlimit)]); %Set voltage ...
    limit
65 fprintf(obj1, ['smub.source.limiti = ',num2str(ilimit)]); %Setcurrent limit
66 fprintf(obj1, ['smub.source.levelv = 0']); %Set source voltage to 0 to ...
    start
67 %% Configure Measurements & Buffers
68
69 % SMUA Measurement Settings - Drain (Measuring Current)
70 fprintf(obj1, 'smua.measure.autozero = smua.AUTOZERO_ONCE');
```

C. MATLAB Test Scripts

```
71 fprintf(obj1, ['smua.measure.rangei = ', num2str(irange)]); % Set ...
    current range
72 fprintf(obj1, ['smua.measure.nplc = ', num2str(nplc)]); % Measurement ...
    integration rate
73 fprintf(obj1, ['smua.measure.count = ', num2str(measnum)]); % Number of ...
    measurements to collect.
74 fprintf(obj1, ['smua.measure.delay = ', num2str(measdelay)]); % Set the ...
    delay before the first measurement
75 fprintf(obj1, ['smua.measure.interval = ', num2str(measinterval)]); % ...
    Set the delay between measurements
76
77 %SMUA Buffer 1 - Stores drain voltage
78 fprintf(obj1, 'smua.nvbuffer1.clear() '); % Clears the buffer
79 fprintf(obj1, 'smua.nvbuffer1.collecttimestamps = 1 '); % Enables ...
    timestamp collect
80 fprintf(obj1, ['smua.nvbuffer1.appendmode = ', num2str(buffappend)]); ...
    %Overwrites previous measurements in buffer
81 fprintf(obj1, 'smua.nvbuffer1.timestampresolution = 0.000001 '); %Sets ...
    timestamp resolution to the lus (finest)
82
83 %SMUA Buffer 2 - Stores drain current
84 fprintf(obj1, 'smua.nvbuffer2.clear() '); % Clears the buffer
85 fprintf(obj1, 'smua.nvbuffer2.collecttimestamps = 1 '); % Enables ...
    timestamp collect
86 fprintf(obj1, ['smua.nvbuffer2.appendmode = ', num2str(buffappend)]); ...
    %Overwrites previous measurements in buffer
87 fprintf(obj1, 'smua.nvbuffer2.timestampresolution = 0.000001 '); %Sets ...
    timestamp resolution to the lus (finest)
88
89 %SMUB Measurement Settings - Gate (Measuring Voltage)
90 fprintf(obj1, 'smub.measure.autozero = smub.AUTOZERO_ONCE');
91 fprintf(obj1, ['smub.measure.nplc = ', num2str(nplc)]); % Measurement ...
    integration rate
92 fprintf(obj1, ['smub.measure.rangev = ', num2str(vrange)]); % Set ...
    measure range
93 fprintf(obj1, ['smub.measure.count = ', num2str(measnum)]); % Number of ...
    measurements to collect.
94 fprintf(obj1, ['smub.measure.delay = ', num2str(measdelay)]); % Set the ...
    delay before the first measurement
95 fprintf(obj1, ['smub.measure.interval = ', num2str(measinterval)]); % ...
    Set the delay between measurements
96
97 %SMUB Buffer 1
98 fprintf(obj1, 'smub.nvbuffer1.clear() '); % Clears the buffer
99 fprintf(obj1, 'smub.nvbuffer1.collecttimestamps = 1 '); % Enables ...
    timestamp collect
```

C. MATLAB Test Scripts

```
100 fprintf(obj1, ['smub.nvbuffer1.appendmode = ', num2str(buffappend)]); ...
    %Overwrites previous measurements in buffer
101 fprintf(obj1, 'smub.nvbuffer1.timestampresolution = 0.000001 '); %Sets ...
    timestamp resolution to the 1us (finest)
102 %% Execute sweep
103 fprintf(obj1, 'display.clear()');
104 fprintf(obj1, 'display.screen = 2');
105
106 vdsloop = vdsstart;
107 fprintf(obj1, ['vdsloop = ', num2str(vdsstart)]);
108 fprintf(obj1, ['stepsize = ', num2str(stepsize)]);
109 fprintf(obj1, ['loopend = 0']);
110 fprintf(obj1, ['loopcheck = ', num2str(vdsstop-stepsize)]);
111
112 fprintf(obj1, ['smua.source.output = smua.OUTPUT_ON ']); ...
    %Measure w/shutter closed
113 fprintf(obj1, ['smub.source.output = smub.OUTPUT_ON ']); ,...
    %Enable Output
114 fprintf(obj1, ['smua.source.levelv = vdsloop ']); %Set drain voltage
115 fprintf(obj1, ['smub.source.levelv = ', num2str(vgsbias), ' ']); %Set ...
    gate bias voltage
116 fprintf(obj1, ['delay(2) ']); ,...
    %Enable Output
117
118 fprintf(obj1, ['for i = 1,', num2str(vdssteps), 'do ', ...
119     'smua.source.levelv = vdsloop ', ... %Set drain voltage
120     'smub.source.levelv = ', num2str(vgsbias), ' ', ...
    %Set gate bias voltage
121     'smua.source.output = smua.OUTPUT_ON ', ...
    %Measure w/shutter closed
122     'smub.source.output = smub.OUTPUT_ON ', ...
    %Enable Output
123     'waitcomplete() ', ...
124     'smub.measure.v(smub.nvbuffer1) ', ...
    %Measure gate voltage
125     'smua.measure.v(smua.nvbuffer1) ', ...
    %Measure drain voltage
126     'smua.measure.i(smua.nvbuffer2) ', ...
    %Measure drain current
127     'waitcomplete() ', ...
    %Measure w/shutter open
128     'digio.writebit(2,1) ', ...
    %Set digio bit 2 to High for shutter open
129     'delay(', num2str(shutterdelay), ') ', ...
130     'smub.measure.v(smub.nvbuffer1) ', ...
    %Measure gate voltage
```

```

131         'smua.measure.v(smua.nvbuffer1) ',...           ...
           %Measure drain voltage
132         'smua.measure.i(smua.nvbuffer2) ',...           ...
           %Measure drain current
133         'waitcomplete() ',...
134         'smua.source.output = smua.OUTPUT_OFF ',...     ...
           %Disable output
135         'smub.source.output = smub.OUTPUT_OFF ',...     ...
           %Disable output
136         'digio.writebit(2,0) ',...                       ...
           %Set digio bit 2 to Low for shutter close
137         'vdsloop = vdsloop + stepsize ',...             ...
           %Increment loop
138         'if (vdsloop == loopcheck) then loopend = 1 end ',...
139     'end ']);
140 loopend = 0;
141 while loopend == 0
142     fprintf(obj1, 'print(loopend)'); %check the loopend variable
143     loopend = fscanf(obj1);
144     %pause(.1);
145 end
146 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
           (duration,frequency)
147 %% Retrieve data from keithley
148 fprintf(obj1, 'display.clear()');
149 fprintf(obj1, 'display.settext("$BSending Data$B)');
150
151 fprintf(obj1, 'print(smub.nvbuffer1.basetimestamp)'); % print the ...
           timestamp of the buffer
152 temp = fscanf(obj1);
153 vgsbuffbasetime=str2num(temp);
154 clear temp;
155 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
           (duration,frequency)
156
157 fprintf(obj1, 'printbuffer(1,smub.nvbuffer1.n,smub.nvbuffer1.readings)');
158 temp = fscanf(obj1);
159 vgsbuffmeas = cell2mat(textscan(temp, '%f64', 'delimiter', ','));
160 clear temp;
161 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
           (duration,frequency)
162
163 fprintf(obj1, 'printbuffer(1,smub.nvbuffer1.n,smub.nvbuffer1.timestamps)');
164 temp = fscanf(obj1);
165 vgsbufftime = cell2mat(textscan(temp, '%f64', 'delimiter', ','));
166 clear temp;

```

```

167 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
168
169 fprintf(obj1, 'print(smua.nvbuffer1.basetimestamp)');
170 temp = fscanf(obj1);
171 vdsbuffbasetime=str2num(temp);
172 clear temp;
173 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
174
175 fprintf(obj1, 'printbuffer(1,smua.nvbuffer1.n,smua.nvbuffer1.readings)');
176 temp = fscanf(obj1);
177 vdsbuffmeas = cell2mat(textscan(temp, '%f64', 'delimiter', ','));
178 clear temp;
179 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
180
181 fprintf(obj1, 'printbuffer(1,smua.nvbuffer1.n,smua.nvbuffer1.timestamps)');
182 temp = fscanf(obj1);
183 vdsbufftime = cell2mat(textscan(temp, '%f64', 'delimiter', ','));
184 clear temp;
185 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
186
187 fprintf(obj1, 'print(smua.nvbuffer2.basetimestamp)');
188 temp = fscanf(obj1);
189 idsbuffbasetime=str2num(temp);
190 clear temp;
191 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
192
193 fprintf(obj1, 'printbuffer(1,smua.nvbuffer2.n,smua.nvbuffer2.readings)');
194 temp = fscanf(obj1);
195 idsbuffmeas = cell2mat(textscan(temp, '%f64', 'delimiter', ','));
196 clear temp;
197 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
198
199 fprintf(obj1, 'printbuffer(1,smua.nvbuffer2.n,smua.nvbuffer2.timestamps)');
200 temp = fscanf(obj1);
201 idsbufftime = cell2mat(textscan(temp, '%f64', 'delimiter', ','));
202 clear temp;
203 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
204 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)

```

C. MATLAB Test Scripts

```
205 %% Print to Keithley screen and beep complete sequence
206 fprintf(obj1, 'display.clear()');
207 fprintf(obj1, 'display.setText("Test Complete")');
208 fprintf(obj1, 'beeper.beep(0.12, 500)'); %send beeper tone ...
    (duration,frequency)
209 fprintf(obj1, 'beeper.beep(0.12, 800)'); %send beeper tone ...
    (duration,frequency)
210 fprintf(obj1, 'beeper.beep(0.12, 1000)'); %send beeper tone ...
    (duration,frequency)
211 fprintf(obj1, 'beeper.beep(0.12, 2000)'); %send beeper tone ...
    (duration,frequency)
212 %% Data Reduction
213 vgsmeastime = vgsbufftime+vgsbuffbasetime; %Add the basetime to each ...
    measurement time
214 vdsmeastime = vdsbufftime+vdsbuffbasetime;
215 idsmeastime = idsbufftime+idsbuffbasetime;
216
217 %take the buffers, average and split
218 vgsbuffmeas(1) = vgsbuffmeas(2); % Replace first entry due to Keithley ...
    error on first measurement
219 vdsbuffmeas(1) = vdsbuffmeas(2);
220 idsbuffmeas(1) = idsbuffmeas(2);
221
222 %Reduce data for open and closed shutter measurements
223 %creates and open and closed vector from the read buffer which correspond
224 %to each pair of measurements.
225 j=1;
226 for i=1:vdssteps
227     vgsdclosed(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        vgsbuffmeas(((j*measnum)-measnum+1):(j*measnum));
228     vdsdclosed(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        vdsbuffmeas(((j*measnum)-measnum+1):(j*measnum));
229     idsclosed(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        idsbuffmeas(((j*measnum)-measnum+1):(j*measnum));
230     vgsdclosedtime(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        vgsbufftime(((j*measnum)-measnum+1):(j*measnum));
231     vdsdclosedtime(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        vdsbufftime(((j*measnum)-measnum+1):(j*measnum));
232     idsclosedtime(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        idsbufftime(((j*measnum)-measnum+1):(j*measnum));
233     j=j+1;
234
235     vgsdopen(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        vgsbuffmeas(((j*measnum)-measnum+1):(j*measnum));
236     vdsdopen(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        vdsbuffmeas(((j*measnum)-measnum+1):(j*measnum));
```

C. MATLAB Test Scripts

```
237     idsopen(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        idsbuffmeas(((j*measnum)-measnum+1):(j*measnum));
238     vgsopentime(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        vgsbufftime(((j*measnum)-measnum+1):(j*measnum));
239     vdsopentime(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        vdsbufftime(((j*measnum)-measnum+1):(j*measnum));
240     idsopentime(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        idsbufftime(((j*measnum)-measnum+1):(j*measnum));
241     j=j+1;
242 end
243 vgsdiff = vgsopen-vgsclosed;
244 vdsdiff = vdsopen-vdsclosed;
245 idsdiff = idsopen-idsclosed;
246
247 %FFT of signals
248 for i=1:measnum-1
249     closedtimediff(i)=vdsclosedtime(i+1)-vdsclosedtime(i);
250 end
251
252 NFFT = 2^nextpow2(measnum); % Next power of 2 from length of y
253 Fs = 1/mean(closedtimediff);
254 f = (0:NFFT/2-1)*Fs/NFFT;
255
256 for i=1:vdssteps
257     signal = ...
        idsclosed(((i-1)*measnum+1):(i*measnum))-mean(idsclosed(((i-1)*
258 measnum+1):(i*measnum)));
259     window = hamming(length(signal));
260     signal = signal.*window';
261     tempfft = fft(signal,NFFT)/NFFT; %Mean subtracted
262     tempfft = tempfft(1:NFFT/2);
263     idsclosedfft(i,:) = abs(tempfft);
264     clear tempfft signal;
265     signal = idsopen(((i-1)*measnum+1):(i*measnum))-mean(idsopen(((i-1)*
266 measnum+1):(i*measnum)));
267     window = hamming(length(signal));
268     signal = signal.*window';
269     tempfft = fft(signal,NFFT)/NFFT; %Mean subtracted
270     tempfft = tempfft(1:NFFT/2);
271     idsopenfft(i,:) = abs(tempfft);
272     idsfftndiff(i,:) = idsopenfft(i,:)-idsclosedfft(i,:);
273     clear tempfft signal;
274 end
275
276 %[fftmaxdiff(1,:),fftmaxdiff(2,)] = max(vdsopenfft(:,25:NFFT/2), [],2);
277 %fftmaxdiff(2,)=fftmaxdiff(2,)+25;
```

C. MATLAB Test Scripts

```
278
279 for i=1:(vdssteps)
280     vgsavgmeasclosed(i) = ...
        mean(vgsclosed((i*measnum)-(measnum-1)):i*measnum)); %Averages ...
        the submeasurements for each vgs step
281     vdsavgmeasclosed(i) = ...
        mean(vdsclosed((i*measnum)-(measnum-1)):i*measnum));
282     idsavgmeasclosed(i) = ...
        mean(idsclosed((i*measnum)-(measnum-1)):i*measnum));
283     vgsavgmeasopen(i) = ...
        mean(vgsopen((i*measnum)-(measnum-1)):i*measnum)); %Averages ...
        the submeasurements for each vgs step
284     vdsavgmeasopen(i) = mean(vdsopen((i*measnum)-(measnum-1)):i*measnum));
285     idsavgmeasopen(i) = mean(idsopen((i*measnum)-(measnum-1)):i*measnum));
286     vgsavgmeasdiff(i) = ...
        mean(vgsdiff((i*measnum)-(measnum-1)):i*measnum)); %Averages ...
        the submeasurements for each vgs step
287     vdsavgmeasdiff(i) = mean(vdsdiff((i*measnum)-(measnum-1)):i*measnum));
288     idsavgmeasdiff(i) = mean(idsdiff((i*measnum)-(measnum-1)):i*measnum));
289 end
290
291 mkdir([foldername]);
292 %% Plots
293 %Shutter closed averages
294 figure(1);
295 idvar = 1E-7;
296 [AX,H1,H2] = ...
        plotyy(vdsavgmeasclosed,vgsavgmeasclosed,vdsavgmeasclosed,idsavgmeasclosed);
297 set(AX(1),'fontsize',14,'Position',[0.12 0.17 0.72 0.72]);
298 set(AX(2),'xtick',[],'fontsize',14);
299 %set(AX(2),'xtick',[],'YLim',[(idbias-idvar) ...
        (idbias+idvar)],'fontsize',14);
300 set(H1,'marker','*');
301 set(H2,'marker','*');
302 set(get(AX(1),'Ylabel'),'String','Avg Gate Voltage, Vgs ...
        (V)','fontsize',14);
303 set(get(AX(2),'Ylabel'),'String','Avg Source-Drain Current, Id ...
        (A)','fontsize',14);
304 set(gcf,'color','w');
305 %xlim([vgsstart vgsstop]);
306 title(['Vds Sweep (Avg of ',num2str(measnum),' Measurements; Vgs = ...
        ',num2str(vgsbias),' V); Shutter Closed']);
307 xlabel('Drain Voltage, Vds (V)');
308 grid on;
309 print('-dpng',[foldername,'\',filespecifier,'-avgsclosed.png']);
310 %%
```

```

311 %Shutter open averages
312 figure(2);
313 [AX,H1,H2] = ...
    plotyy(vdsavgmeasopen,vgsavgmeasopen,vdsavgmeasopen,idsavgmeasopen);
314 set(AX(1),'fontsize',14,'Position',[0.12 0.17 0.72 0.72]);
315 set(AX(2),'xtick',[],'fontsize',14);
316 %set(AX(2),'xtick',[],'YLim',[(idbias-idvar) ...
    (idbias+idvar)],'fontsize',14);
317 set(H1,'marker','*');
318 set(H2,'marker','*');
319 set(get(AX(1),'Ylabel'),'String','Avg Gate Voltage, Vgs ...
    (V)','fontsize',14);
320 set(get(AX(2),'Ylabel'),'String','Avg Source-Drain Current, Id ...
    (A)','fontsize',14);
321 set(gcf,'color','w');
322 title(['Vds Sweep (Avg of ',num2str(measnum),' Measurements; Vgs = ...
    ',num2str(vgsbias),' V); Shutter Open']);
323 xlabel('Drain Voltage, Vds (V)');
324 grid on;
325 print('-dpng',[foldername,'\',filespecifier,'-avgsopen.png']);
326 %%
327 %Ids vs time
328 figure(3);
329 plot(idsmeastime,idsbuffmeas,'-*');
330 xlabel('Time (s)');
331 ylabel('Source-Drain Current (A)');
332 title('Source-Drain Current (A) vs Time (s)');
333 set(gca,'fontsize',14);
334 set(gcf,'color','w');
335 grid on;
336 print('-dpng',[foldername,'\',filespecifier,'-ids-time.png']);
337 %%
338 %Plots Ids difference vs vgs;
339 figure(5);
340 plot(vdsopen,idsdiff,'*');
341 set(gca,'fontsize',14);
342 set(gcf,'color','w');
343 xlabel('Drain Voltage, Vds (V)');
344 ylabel('Source-Drain Current Difference (A)');
345 title('Source-Drain Current Difference (A) vs Vds (V)');
346 grid on;
347 print('-dpng',[foldername,'\',filespecifier,'-idsdiffvds.png']);
348 %%
349 %Plots Ids difference vs time;
350 figure(6);
351 plot(idsopentime,idsdiff,'*');

```

```

352 set(gca,'fontsize',14);
353 set(gcf, 'color', 'w');
354 xlabel('Time (s)');
355 ylabel('Source-Drain Current Difference (A)');
356 title('Source-Drain Current Difference (A) vs Time (s)');
357 grid on;
358 print('-dpng', [foldername, '\', filespecifier, '-idsdifftime.png']);
359 %%
360 %Plots avg ids difference vs vds;
361 figure(7);
362 plot(vdsavgmeasopen, idsavgmeasdiff, '-*');
363 set(gca, 'fontsize', 14);
364 set(gcf, 'color', 'w');
365 xlabel('Drain Voltage, Vds (V)');
366 ylabel('Source-Drain Current Difference (A)');
367 title('Avg Source-Drain Current Difference (A) vs Vds (V)');
368 grid on;
369 print('-dpng', [foldername, '\', filespecifier, '-idsavgdiffvds.png']);
370 %%
371 %Plots fft of vds open and closed;
372 figure(8);
373 plot(f(2:length(f)), idsclosedfft(1,2:length(idsclosedfft)), f(2:length(f)),
374 idsopenfft(1,2:length(idsopenfft))); %No DC
375 set(gca, 'fontsize', 14);
376 set(gcf, 'color', 'w');
377 legend('Shutter Closed', 'Shutter Open');
378 xlabel('Frequency (Hz)');
379 ylabel('|Id|');
380 title('FFT of First Vds');
381 grid on;
382 %ylim([-0E-3 1E-3]);
383 %xlim([0 50]);
384 print('-dpng', [foldername, '\', filespecifier, '-idsfft.png']);
385 %%
386 % %Plots fft of vds open and closed;
387 figure(9);
388 clf(9);
389 cc=hsv(vdssteps);
390 hold on;
391 for i=1:vdssteps
392     plot(f(2:length(f)), idsfftdiff(i,2:length(idsfftdiff)), 'color', cc(i,:)); ...
        %No DC
393 end
394 hold off;
395 set(gca, 'fontsize', 14);
396 set(gcf, 'color', 'w');

```

```

397 xlabel('Frequency (Hz)');
398 ylabel('|Ids|');
399 title('Difference of FFT Open/Closed Shutter of Ids');
400 grid on;
401 %ylim([-0E-3 1E-5]);
402 %xlim([825 875]);
403 legendmatrix=cellstr(num2str(vdsavgmeasopen(1:vdssteps),'Vds = %f'));
404 legend(legendmatrix);
405 print('-dpng', [foldername,'\',filespecifier,'-idsfftiff.png']);
406 %% Make a wave to compare expected fft
407 % t=vdscllosedtime(1):1/Fs:vdscllosedtime(length(vdscllosedtime));
408 % %chopperwave = ...
    mean(vdscllosed)+range(vdscllosed)*square(2*pi*chopperfreq*t); %shifted
409 % chopperwave = square(2*pi*chopperfreq*t); %shifted
410 % NFFT1 = 2^nextpow2(length(chopperwave)); % Next power of 2 from ...
    length of y
411 % %Fs = chopperfreq;
412 % f1 = (0:NFFT1/2-1)*Fs/NFFT1;
413 % sqwvfft = fft(chopperwave,NFFT1)/NFFT1;
414 % sqwvfft = sqwvfft(1:NFFT1/2);
415 % sqwvfft = abs(sqwvfft);
416 % figure(11);
417 % subplot(2,1,1); plot(t,chopperwave); xlim([0 1/chopperfreq]); ...
    xlabel('Time (s)');
418 % subplot(2,1,2); plot(f1(2:length(f1)),sqwvfft(2:length(sqwvfft))); ...
    xlabel('Freq (Hz)');%No DC
419 % print('-dpng', [foldername,'\',filespecifier,'-synfft.png']);
420
421 %% Reset Keithley & Save Data
422 save([foldername,'\',filespecifier,'.mat']);
423 fprintf(obj1, '*RST'); %command to reset the keithley
424 %pause(5);
425 %end

```

sweepvgs_fixedvds_measureId.m

```
1 %% Sweeps Vgs w/Fixed Vds - Source Gate Voltage and Drain Voltage, ...
   Measure Drain Current
2 % This script sweeps the gate voltage for a fixed drain voltage bias and
3 % measures the source drain current.
4
5 close all;
6 clearvars -except obj1 k testnum;
7 testnum=testnum+1;
8
9 % Timing Variables
10 cd '/Users/gfertif/Dropbox/Thesis/THz Project/Testing/20140225 Test Data'
11 foldername = [num2str(testnum),'-OldBoard-T1-Gunn-Id'];
12 chopperfreq = 100; %Hz - For plots
13 nplc = .1; % Integration rate of measurement 1,.1,.01,.001 (0.001 finest)
14 measnum = 100; %Number of submeasurements per set ()
15 measinterval = 0.0; %Keithley delay BETWEEN submeasurements
16 measdelay = 00; %Keithley delay BEFORE a set of measurements occurs
17 shutterdelay = 0.00; % Delay (s) for shutter full open
18 newdelay = 0; %Delay(s) before an open/close set of measurements
19
20 % Gate Loop Variables
21 vdsbias = 1.0; %Fixed bias voltage for Vds
22 vgssteps = 25; %Number of steps for Vgs sweep %Max ...
   vgssteps*measnum=10000 for measnum=1000 is 10
23 vgsstart = 0; %Start Voltage for Vgs sweep
24 vgsstop = 1; %Stop Voltage for Vgs sweep
25 stepsize = (vgsstop - vgsstart) / (vgssteps); %Calculate sweep step size
26
27 vrange = 6; %Voltage source range: 100mV, 1V, 6V, 40V
28 irange = 100E-6; % Current source range: 100nA, 1uA, 100uA, 1mA, 10mA, ...
   100mA, 1A, 3A
29 vlimit = 10; %Voltage limit
30 ilimit = 10E-3; %Current limit
31
32 buffappend = 1; %'0' tells the buffers to overwrite for each set of ...
   measurements, '1' appends
33 filespecifier = ...
   ['MeasId.SwpVgs-CH',num2str(chopperfreq),'_MEAS',num2str(measnum),'_NPLC',
34 num2str(nplc),'_VdsBias',num2str(vdsbias)];
35 %% Reset Keithley
36 fprintf(obj1, '*RST'); %command to reset the keithley
37 fprintf(obj1, 'smua.reset()'); % Reset SMU
38 fprintf(obj1, 'smub.reset()'); % Reset SMU
```

```

39 fprintf(obj1, 'errorqueue.clear()'); % Clear the error queue
40
41 %Display and Beep Sequence
42 fprintf(obj1, 'display.clear()');
43 fprintf(obj1, 'display.settext("$BTest in Progress$B")');
44 fprintf(obj1, 'beeper.beep(0.1, 500)'); %send beeper tone ...
    (duration,frequency)
45 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
46
47 %Digio reset
48 fprintf(obj1, ['digio.writebit(2,1) ']); %Set digio bit 2 to High for ...
    shutter open
49 pause(shutterdelay);
50 fprintf(obj1, ['digio.writebit(2,0) ']); %Set digio bit 2 to Low for ...
    shutter close
51 %% Configure Sources
52 % Source A Configuration - Drain
53 fprintf(obj1, ['smua.source.func = smua.OUTPUT_DCVOLTS']); %Select ...
    voltage source
54 fprintf(obj1, ['smua.source.rangev = ',num2str(vrange)]); % Set source ...
    function range
55 fprintf(obj1, ['smua.source.limiti = ',num2str(ilimit)]); %Set current ...
    limit
56 fprintf(obj1, ['smua.source.limitv = ',num2str(vlimit)]); %Set voltage ...
    limit
57 fprintf(obj1, ['smua.source.levelv = 0']); %Set source voltage to 0 to ...
    start
58
59 % Source B Configuration - Gate
60 fprintf(obj1, ['smub.source.func = smub.OUTPUT_DCVOLTS']);
61 fprintf(obj1, ['smub.source.rangev = ',num2str(vrange)]); % Set source ...
    range
62 fprintf(obj1, ['smub.source.limitv = ',num2str(vlimit)]); %Set voltage ...
    limit
63 fprintf(obj1, ['smub.source.limiti = ',num2str(ilimit)]); %Setcurrent limit
64 fprintf(obj1, ['smub.source.levelv = 0']); %Set source voltage to 0 to ...
    start
65 %% Configure Measurements & Buffers
66
67 % SMUA Measurement Settings - Drain (Measuring Current)
68 fprintf(obj1, 'smua.measure.autozero = smua.AUTOZERO_ONCE');
69 %fprintf(obj1, 'smua.measure.autozero = smua.AUTOZERO_OFF');
70 %fprintf(obj1, 'smua.measure.autorangei = smua.AUTORANGE_ON'); % ...
    Enable measure autorange

```

C. MATLAB Test Scripts

```
71 fprintf(obj1, ['smua.measure.rangei = ', num2str(irange)]); % Set ...
    current range
72 fprintf(obj1, ['smua.measure.nplc = ', num2str(nplc)]); % Measurement ...
    integration rate
73 fprintf(obj1, ['smua.measure.count = ', num2str(measnum)]); % Number of ...
    measurements to collect.
74 fprintf(obj1, ['smua.measure.delay = ', num2str(measdelay)]); % Set the ...
    delay before the first measurement
75 fprintf(obj1, ['smua.measure.interval = ', num2str(measinterval)]); % ...
    Set the delay between measurements
76
77 %SMUA Buffer 1 - Stores drain voltage
78 fprintf(obj1, 'smua.nvbuffer1.clear() '); % Clears the buffer
79 fprintf(obj1, 'smua.nvbuffer1.collecttimestamps = 1 '); % Enables ...
    timestamp collect
80 fprintf(obj1, ['smua.nvbuffer1.appendmode = ', num2str(buffappend)]); ...
    %Overwrites previous measurements in buffer
81 fprintf(obj1, 'smua.nvbuffer1.timestampresolution = 0.000001 '); %Sets ...
    timestamp resolution to the 1us (finest)
82
83 %SMUA Buffer 2 - Stores drain current
84 fprintf(obj1, 'smua.nvbuffer2.clear() '); % Clears the buffer
85 fprintf(obj1, 'smua.nvbuffer2.collecttimestamps = 1 '); % Enables ...
    timestamp collect
86 fprintf(obj1, ['smua.nvbuffer2.appendmode = ', num2str(buffappend)]); ...
    %Overwrites previous measurements in buffer
87 fprintf(obj1, 'smua.nvbuffer2.timestampresolution = 0.000001 '); %Sets ...
    timestamp resolution to the 1us (finest)
88
89 %SMUB Measurement Settings - Gate (Measuring Voltage)
90 fprintf(obj1, 'smub.measure.autozero = smub.AUTOZERO_ONCE');
91 %fprintf(obj1, 'smua.measure.autozero = smua.AUTOZERO_OFF');
92 fprintf(obj1, ['smub.measure.nplc = ', num2str(nplc)]); % Measurement ...
    integration rate
93 fprintf(obj1, ['smub.measure.rangev = ', num2str(vrange)]); % Set ...
    measure range
94 fprintf(obj1, ['smub.measure.count = ', num2str(measnum)]); % Number of ...
    measurements to collect.
95 fprintf(obj1, ['smub.measure.delay = ', num2str(measdelay)]); % Set the ...
    delay before the first measurement
96 fprintf(obj1, ['smub.measure.interval = ', num2str(measinterval)]); % ...
    Set the delay between measurements
97
98 %SMUB Buffer 1
99 fprintf(obj1, 'smub.nvbuffer1.clear() '); % Clears the buffer
```

C. MATLAB Test Scripts

```
100 fprintf(obj1, 'smub.nvbuffer1.collecttimestamps = 1 '); % Enables ...
    timestamp collect
101 fprintf(obj1, ['smub.nvbuffer1.appendmode = ', num2str(buffappend)]); ...
    %Overwrites previous measurements in buffer
102 fprintf(obj1, 'smub.nvbuffer1.timestampresolution = 0.000001 '); %Sets ...
    timestamp resolution to the lus (finest)
103 %% Execute sweep
104 fprintf(obj1, 'display.clear()');
105 fprintf(obj1, 'display.screen = 2');
106
107 vgsloop = vgsstart;
108 fprintf(obj1, ['vgsloop = ', num2str(vgsstart)]);
109 fprintf(obj1, ['stepsize = ', num2str(stepsize)]);
110 fprintf(obj1, ['loopend = 0']);
111 fprintf(obj1, ['loopcheck = ', num2str(vgsstop-stepsize)]);
112
113 fprintf(obj1, ['smua.source.output = smua.OUTPUT_ON ']); ...
    %Measure w/shutter closed
114 fprintf(obj1, ['smub.source.output = smub.OUTPUT_ON ']); ,...
    %Enable Output
115 fprintf(obj1, ['smub.source.levelv = vgsloop ']); %Set gate voltage
116 fprintf(obj1, ['smua.source.levelv = ', num2str(vdsbias), ' ']); %Set ...
    drain bias voltage
117 fprintf(obj1, ['delay(3) ']); ,...
    %Enable Output
118
119 fprintf(obj1, ['for i = 1, ', num2str(vgssteps), 'do ', ...
120     'smub.source.levelv = vgsloop ', ... %Set gate voltage
121     'smua.source.levelv = ', num2str(vdsbias), ' ', ...
    %Set drain bias voltage
122     'smua.source.output = smua.OUTPUT_ON ', ...
    %Measure w/shutter closed
123     'smub.source.output = smub.OUTPUT_ON ', ...
    %Enable Output
124     'delay(', num2str(newdelay), ') ', ...
125     'waitcomplete() ', ...
126     'smub.measure.v(smub.nvbuffer1) ', ...
    %Measure gate voltage
127     'smua.measure.v(smua.nvbuffer1) ', ...
    %Measure drain voltage
128     'smua.measure.i(smua.nvbuffer2) ', ...
    %Measure drain current
129     'waitcomplete() ', ...
    %Measure w/shutter open
130     'digio.writebit(2,1) ', ...
    %Set digio bit 2 to High for shutter open
131     'delay(', num2str(shutterdelay), ') ', ...
```

```

132         'smub.measure.v(smub.nvbuffer1) ',...           ...
           %Measure gate voltage
133         'smua.measure.v(smua.nvbuffer1) ',...           ...
           %Measure drain voltage
134         'smua.measure.i(smua.nvbuffer2) ',...           ...
           %Measure drain current
135         'waitcomplete() ',...
136         'smua.source.output = smua.OUTPUT_OFF ',...     ...
           %Disable output
137         'smub.source.output = smub.OUTPUT_OFF ',...     ...
           %Disable output
138         'digio.writebit(2,0) ',...                       ...
           %Set digio bit 2 to Low for shutter close
139         'vgsloop = vgsloop + stepsize ',...             ...
           %Increment loop
140         'if (vgsloop == loopcheck) then loopend = 1 end ',...
141     'end ']);
142 loopend = 0;
143 while loopend == 0
144     fprintf(obj1, 'print(loopend)'); %check the loopend variable
145     loopend = fscanf(obj1);
146     %pause(.1);
147 end
148 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
           (duration,frequency)
149 %% Retrieve data from keithley
150 fprintf(obj1, 'display.clear()');
151 fprintf(obj1, 'display.settext("$BSending Data$B)");
152
153 fprintf(obj1, 'print(smub.nvbuffer1.basetimestamp)'); % print the ...
           timestamp of the buffer
154 temp = fscanf(obj1);
155 vgsbuffbasetime=str2num(temp);
156 clear temp;
157 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
           (duration,frequency)
158
159 fprintf(obj1, 'printbuffer(1,smub.nvbuffer1.n,smub.nvbuffer1.readings)');
160 temp = fscanf(obj1);
161 vgsbuffmeas = cell2mat(textscan(temp, '%f64', 'delimiter', ','));
162 clear temp;
163 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
           (duration,frequency)
164
165 fprintf(obj1, 'printbuffer(1,smub.nvbuffer1.n,smub.nvbuffer1.timestamps)');
166 temp = fscanf(obj1);

```

```

167 vgsbufftime = cell2mat(textscan(temp, '%f64', 'delimiter', ','));
168 clear temp;
169 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
170
171 fprintf(obj1, 'print(smua.nvbuffer1.basetimestamp)');
172 temp = fscanf(obj1);
173 vdsbuffbasetime=str2num(temp);
174 clear temp;
175 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
176
177 fprintf(obj1, 'printbuffer(1,smua.nvbuffer1.n,smua.nvbuffer1.readings)');
178 temp = fscanf(obj1);
179 vdsbuffmeas = cell2mat(textscan(temp, '%f64', 'delimiter', ','));
180 clear temp;
181 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
182
183 fprintf(obj1, 'printbuffer(1,smua.nvbuffer1.n,smua.nvbuffer1.timestamps)');
184 temp = fscanf(obj1);
185 vdsbufftime = cell2mat(textscan(temp, '%f64', 'delimiter', ','));
186 clear temp;
187 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
188
189 fprintf(obj1, 'print(smua.nvbuffer2.basetimestamp)');
190 temp = fscanf(obj1);
191 idsbuffbasetime=str2num(temp);
192 clear temp;
193 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
194
195 fprintf(obj1, 'printbuffer(1,smua.nvbuffer2.n,smua.nvbuffer2.readings)');
196 temp = fscanf(obj1);
197 idsbuffmeas = cell2mat(textscan(temp, '%f64', 'delimiter', ','));
198 clear temp;
199 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
200
201 fprintf(obj1, 'printbuffer(1,smua.nvbuffer2.n,smua.nvbuffer2.timestamps)');
202 temp = fscanf(obj1);
203 idsbufftime = cell2mat(textscan(temp, '%f64', 'delimiter', ','));
204 clear temp;
205 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)

```

C. MATLAB Test Scripts

```
206 fprintf(obj1, 'beeper.beep(0.1, 2000)'); %send beeper tone ...
    (duration,frequency)
207 %% Print to Keithley screen and beep complete sequence
208 fprintf(obj1, 'display.clear()');
209 fprintf(obj1, 'display.settext("Test Complete")');
210 fprintf(obj1, 'beeper.beep(0.12, 500)'); %send beeper tone ...
    (duration,frequency)
211 fprintf(obj1, 'beeper.beep(0.12, 800)'); %send beeper tone ...
    (duration,frequency)
212 fprintf(obj1, 'beeper.beep(0.12, 1000)'); %send beeper tone ...
    (duration,frequency)
213 fprintf(obj1, 'beeper.beep(0.12, 2000)'); %send beeper tone ...
    (duration,frequency)
214 %% Data Reduction
215 vgsmeastime = vgsbufftime+vgsbuffbasetime; %Add the basetime to each ...
    measurement time
216 vdsmeastime = vdsbufftime+vdsbuffbasetime;
217 idsmeastime = idsbufftime+idsbuffbasetime;
218
219 %take the buffers, average and split
220 vgsbuffmeas(1) = vgsbuffmeas(2); % Replace first entry due to Keithley ...
    error on first measurement
221 vdsbuffmeas(1) = vdsbuffmeas(2);
222 idsbuffmeas(1) = idsbuffmeas(2);
223
224 %Reduce data for open and closed shutter measurements
225 %creates and open and closed vector from the read buffer which correspond
226 %to each pair of measurements.
227 j=1;
228 for i=1:vgssteps
229     vgsdclosed(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        vgsbuffmeas(((j*measnum)-measnum+1):(j*measnum));
230     vdsdclosed(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        vdsbuffmeas(((j*measnum)-measnum+1):(j*measnum));
231     idsclosed(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        idsbuffmeas(((j*measnum)-measnum+1):(j*measnum));
232     vgsdclosedtime(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        vgsbufftime(((j*measnum)-measnum+1):(j*measnum));
233     vdsdclosedtime(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        vdsbufftime(((j*measnum)-measnum+1):(j*measnum));
234     idsclosedtime(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        idsbufftime(((j*measnum)-measnum+1):(j*measnum));
235     j=j+1;
236
237     vgsdopen(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        vgsbuffmeas(((j*measnum)-measnum+1):(j*measnum));
```

C. MATLAB Test Scripts

```
238     vdsopen(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        vdsbuffmeas(((j*measnum)-measnum+1):(j*measnum));
239     idsopen(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        idsbuffmeas(((j*measnum)-measnum+1):(j*measnum));
240     vgsopentime(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        vgsbufftime(((j*measnum)-measnum+1):(j*measnum));
241     vdsopentime(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        vdsbufftime(((j*measnum)-measnum+1):(j*measnum));
242     idsopentime(((i-1)*measnum)+1):((i-1)*measnum)+measnum) = ...
        idsbufftime(((j*measnum)-measnum+1):(j*measnum));
243     j=j+1;
244 end
245 vgsdiff = vgsopen-vgsclosed;
246 vdsdiff = vdsopen-vdsclosed;
247 idsdiff = idsopen-idsclosed;
248
249 %FFT of signals
250 for i=1:measnum-1
251     closedtimediff(i)=vdsclosedtime(i+1)-vdsclosedtime(i);
252 end
253
254 NFFT = 2^nextpow2(measnum); % Next power of 2 from length of y
255 Fs = 1/mean(closedtimediff);
256 f = (0:NFFT/2-1)*Fs/NFFT;
257
258 for i=1:vgsteps
259     signal = idsclosed(((i-1)*measnum+1):(i*measnum))-mean(idsclosed(
260 ((i-1)*measnum+1):(i*measnum)));
261     window = hamming(length(signal));
262     signal = signal.*window';
263     tempfft = fft(signal,NFFT)/NFFT; %Mean subtracted
264     tempfft = tempfft(1:NFFT/2);
265     idsclosedfft(i,:) = abs(tempfft);
266     clear tempfft signal;
267     signal = idsopen(((i-1)*measnum+1):(i*measnum))-mean(idsopen(
268 ((i-1)*measnum+1):(i*measnum)));
269     window = hamming(length(signal));
270     signal = signal.*window';
271     tempfft = fft(signal,NFFT)/NFFT; %Mean subtracted
272     tempfft = tempfft(1:NFFT/2);
273     idsopenfft(i,:) = abs(tempfft);
274     idsfftndiff(i,:) = idsopenfft(i,:)-idsclosedfft(i,:);
275     clear tempfft signal;
276 end
277
278 %[fftmaxdiff(1,:),fftmaxdiff(2,:)] = max(vdsopenfft(:,25:NFFT/2),[],2);
```

C. MATLAB Test Scripts

```
279 %fftmaxdiff(2,:)=fftmaxdiff(2,:)+25;
280
281 for i=1:(vgssteps)
282     vgsavgmeasclosed(i) = ...
        mean(vgsclosed((i*measnum)-(measnum-1)):i*measnum)); %Averages ...
        the submeasurements for each vgs step
283     vdsavgmeasclosed(i) = ...
        mean(vdsclosed((i*measnum)-(measnum-1)):i*measnum));
284     idsavgmeasclosed(i) = ...
        mean(idsclosed((i*measnum)-(measnum-1)):i*measnum));
285     vgsavgmeasopen(i) = ...
        mean(vgsopen((i*measnum)-(measnum-1)):i*measnum)); %Averages ...
        the submeasurements for each vgs step
286     vdsavgmeasopen(i) = mean(vdsopen((i*measnum)-(measnum-1)):i*measnum));
287     idsavgmeasopen(i) = mean(idsopen((i*measnum)-(measnum-1)):i*measnum));
288     vgsavgmeasdiff(i) = ...
        mean(vgsdiff((i*measnum)-(measnum-1)):i*measnum)); %Averages ...
        the submeasurements for each vgs step
289     vdsavgmeasdiff(i) = mean(vdsdiff((i*measnum)-(measnum-1)):i*measnum));
290     idsavgmeasdiff(i) = mean(idsdiff((i*measnum)-(measnum-1)):i*measnum));
291 end
292
293 mkdir([foldername]);
294 %% Plots
295 %Shutter closed averages
296 figure(1);
297 idvar = 1E-7;
298 [AX,H1,H2] = ...
        plotyy(vgsavgmeasclosed,vdsavgmeasclosed,vgsavgmeasclosed,idsavgmeasclosed);
299 set(AX(1),'fontsize',14,'Position',[0.12 0.17 0.72 0.72]);
300 set(AX(2),'xtick',[],'fontsize',14);
301 %set(AX(2),'xtick',[],'YLim',[(idbias-idvar) ...
        (idbias+idvar)],'fontsize',14);
302 set(H1,'marker','*');
303 set(H2,'marker','*');
304 set(get(AX(1),'Ylabel'),'String','Avg Source-Drain Voltage, Vds ...
        (V)','fontsize',14);
305 set(get(AX(2),'Ylabel'),'String','Avg Source-Drain Current, Id ...
        (A)','fontsize',14);
306 set(gcf,'color','w');
307 %xlim([vgsstart vgsstop]);
308 title(['Vgs Sweep (Avg of ',num2str(measnum),' Measurements; Vds = ...
        ',num2str(vdsbias),' V); Shutter Closed']);
309 xlabel('Gate Voltage, Vgs (V)');
310 grid on;
311 print('-dpng',[foldername,'\',filespecifier,'-avgsclosed.png']);
```

```

312 %%
313 %Shutter open averages
314 figure(2);
315 [AX,H1,H2] = ...
    plotyy(vgsavgmeasopen,vdsavgmeasopen,vgsavgmeasopen,idsavgmeasopen);
316 set(AX(1),'fontsize',14,'Position',[0.12 0.17 0.72 0.72]);
317 set(AX(2),'xtick',[],'fontsize',14);
318 %set(AX(2),'xtick',[],'YLim',[(idbias-idvar) ...
    (idbias+idvar)],'fontsize',14);
319 set(H1,'marker','*');
320 set(H2,'marker','*');
321 set(get(AX(1),'Ylabel'),'String','Avg Source-Drain Voltage, Vds ...
    (V)','fontsize',14);
322 set(get(AX(2),'Ylabel'),'String','Avg Source-Drain Current, Id ...
    (A)','fontsize',14);
323 set(gcf,'color','w');
324 title(['Vgs Sweep (Avg of ',num2str(measnum),' Measurements; Vds = ...
    ',num2str(vdsbias),' V); Shutter Open']);
325 xlabel('Gate Voltage, Vgs (V)');
326 grid on;
327 print('-dpng',[foldername,'\',filespecifier,'-avgsopen.png']);
328 %%
329 %Ids vs time
330 figure(3);
331 plot(idsmeastime,idsbuffmeas,'-*');
332 xlabel('Time (s)');
333 ylabel('Source-Drain Current (A)');
334 title('Source-Drain Current (A) vs Time (s)');
335 set(gca,'fontsize',14);
336 set(gcf,'color','w');
337 grid on;
338 print('-dpng',[foldername,'\',filespecifier,'-ids-time.png']);
339 %%
340 %Plots Ids difference vs vgs;
341 figure(5);
342 plot(vgsopen,idsdiff,'*');
343 set(gca,'fontsize',14);
344 set(gcf,'color','w');
345 xlabel('Gate Voltage, Vgs (V)');
346 ylabel('Source-Drain Current Difference (A)');
347 title('Source-Drain Current Difference (A) vs Vgs (V)');
348 grid on;
349 print('-dpng',[foldername,'\',filespecifier,'-idsdiffvgs.png']);
350 %%
351 %Plots Ids difference vs time;
352 figure(6);

```

```

353 plot(idsopentime,idsdiff,'*');
354 set(gca,'fontsize',14);
355 set(gcf,'color','w');
356 xlabel('Time (s)');
357 ylabel('Source-Drain Current Difference (A)');
358 title('Source-Drain Current Difference (A) vs Time (s)');
359 grid on;
360 print('-dpng',[foldername,'\',filespecifier,'-idsdifftime.png']);
361 %%
362 %Plots avg ids difference vs vgs;
363 figure(7);
364 plot(vgsavgmeasopen,idsavgmeasdiff,'-*');
365 set(gca,'fontsize',14);
366 set(gcf,'color','w');
367 xlabel('Gate Voltage, Vgs (V)');
368 ylabel('Source-Drain Current Difference (A)');
369 title('Avg Source-Drain Current Difference (A) vs Vgs (V)');
370 grid on;
371 print('-dpng',[foldername,'\',filespecifier,'-idsavgdiffvgs.png']);
372 %%
373 %Plots fft of vds open and closed;
374 % figure(8);
375 % ...
    plot(f(2:length(f)),idsclosedfft(1,2:length(idsclosedfft)),f(2:length(f)),
376 idsopenfft(1,2:length(idsopenfft))); %No DC
377 % set(gca,'fontsize',14);
378 % set(gcf,'color','w');
379 % legend('Shutter Closed','Shutter Open');
380 % xlabel('Frequency (Hz)');
381 % ylabel('|Id|');
382 % title('FFT of First Vds');
383 % grid on;
384 % ylim([-0E-3 1E-3]);
385 % xlim([0 50]);
386 % print('-dpng',[foldername,'\',filespecifier,'-idsfft.png']);
387 %%
388 %Plots fft of vds open and closed;
389 % figure(9);
390 % clf(9);
391 % cc=HSV(vgssteps);
392 % for i=1:vgssteps
393 %     hold on;
394 %     ...
    plot(f(2:length(f)),idsfftdiff(i,2:length(idsfftdiff)),'color',cc(i,:)); ...
    %No DC
395 %     hold off;

```

C. MATLAB Test Scripts

```
396 % end
397 % set(gca,'fontsize',14);
398 % set(gcf, 'color', 'w');
399 % xlabel('Frequency (Hz)');
400 % ylabel('|Ids|');
401 % title('Difference of FFT Open/Closed Shutter of Ids');
402 % grid on;
403 % ylim([-0E-3 1E-5]);
404 % xlim([825 875]);
405 % legendmatrix=cellstr(num2str(vgsavgmeasopen(1:vgssteps)'),'Vgs = %f');
406 % legend(legendmatrix);
407 % print('-dpng', [foldername,'\',filespecifier,'-idsfft.png']);
408 %% Make a wave to compare expected fft
409 t=vdscllosedtime(1):1/Fs:vdscllosedtime(length(vdscllosedtime));
410 % %chopperwave = ...
    mean(vdscllosed)+range(vdscllosed)*square(2*pi*chopperfreq*t); %shifted
411 % chopperwave = square(2*pi*chopperfreq*t); %shifted
412 % NFFT1 = 2^nextpow2(length(chopperwave)); % Next power of 2 from ...
    length of y
413 % %Fs = chopperfreq;
414 % f1 = (0:NFFT1/2-1)*Fs/NFFT1;
415 % sqwvfft = fft(chopperwave,NFFT1)/NFFT1;
416 % sqwvfft = sqwvfft(1:NFFT1/2);
417 % sqwvfft = abs(sqwvfft);
418 % figure(11);
419 % subplot(2,1,1); plot(t,chopperwave); xlim([0 1/chopperfreq]); ...
    xlabel('Time (s)');
420 % subplot(2,1,2); plot(f1(2:length(f1)),sqwvfft(2:length(sqwvfft))); ...
    xlabel('Freq (Hz)');%No DC
421 % print('-dpng', [foldername,'\',filespecifier,'-synfft.png']);
422
423 %% Reset Keithley & Save Data
424 save([foldername,'\',filespecifier,'.mat']);
425 fprintf(obj1, '*RST'); %command to reset the keithley
```

Bibliography

- [1] S. Hayashi and K. Kawase. Recent optical and photonic technologies. <http://www.intechopen.com/books/export/citation/BibTex/recent-optical-and-photonic-technologies/terahertz-wave-parametric-sources>, 2010-01-01.
- [2] Hyperphysics. Light and vision. <http://hyperphysics.phy-astr.gsu.edu/hbase/atmos/blusky.html>, Accessed June 2014.
- [3] MMW Concepts. Atmospheric attenuation chart. <http://www.mmwconcepts.com>, 2014.
- [4] C. M. Armstrong. The truth about terahertz. *IEEE Spectrum*, 49(9):36–41, 2012.
- [5] T.G. Phillips and J. Keene. Submillimeter astronomy [heterodyne spectroscopy]. *Proceedings of the IEEE*, 80(11):1662–1678, Nov 1992.
- [6] Terahertz Spectral Database. National institute of standards and technology: National measurement laboratory. <http://webbook.nist.gov/chemistry/thz-ir/>, Accessed July 2014.
- [7] Digital Barriers. Thruvision. www.digitalbarriers.com/ThruVision, May 2014.
- [8] Fraunhofer IPM. T-cognition 2.0, terahertz spectrometer. http://www.ipm.fraunhofer.de/content/dam/ipm/de/PDFs/produktblaetter/MC/130130_T-Cognition_2%200_M_1.pdf, 2014.
- [9] Progress report on safe visitor: approaching a practical instrument for terahertz security screening. *Proc. SPIE*, 7670:767005–767005–8, 2010.
- [10] S.U. Hwu, K.B. deSilva, and C.T. Jih. Terahertz (thz) wireless systems for space applications. *Sensors Applications Symposium (SAS), 2013 IEEE*, pages 171–175, Feb 2013.
- [11] TeraView Limited. Teraview website. <http://www.teraview.com/index.html>, May 2014.

- [12] T. Ytterdal, Y. Cheng, and T.A. Fjeldly. Device modeling for analog and rf cmos circuit design: Mosfet device physics and operation. *John Wiley & Sons. Ltd*, 2003.
- [13] W. Wilson. Introduction to physical electronics. *Rice University Course: ELEC 305* <http://cnx.org/content/col10114/latest/>, 2014.
- [14] E.F. Schubert. Ecse-6290 semiconductor devices and models ii: Mosfet - basics, 2003. (*Rensselaer Polytechnic Institute*), <http://www.ecse.rpi.edu/~schubert/2003-Course-ECSE-6290-SDM-2/> (Accessed 4 Jun 14).
- [15] A. Rogalski and F. Sizov. Terahertz detectors and focal plane arrays. *Opto-Electronics Review*, 19(3):346–404, 2011.
- [16] W. Knap, V. Kachorovskii, Y. Deng, S. Rumyantsev, J.-Q. Lü, R. Gaska, M. S. Shur, G. Simin, X. Hu, M. Asif Khan, C. A. Saylor, and L. C. Brunel. Nonresonant detection of terahertz radiation in field effect transistors. *Journal of Applied Physics*, 91(11):9346–9353, 2002.
- [17] F. Schuster, D. Coquillat, H. Videlier, M. Sakowicz, F. Teppe, L. Dussopt, B. Giffard, T. Skotnicki, and W. Knap. Broadband terahertz imaging with highly sensitive silicon cmos detectors. *Opt. Express*, 19(8):7827–7832, Apr 2011.
- [18] Custom Microwave Incorporated. Cmi website. <http://www.custommicrowave.com/index.php>, Accessed 17Jul14.
- [19] R. Tauk, F. Teppe, S. Boubanga, D. Coquillat, W. Knap, Y. M. Meziani, C. Gallon, F. Boeuf, T. Skotnicki, C. Fenouillet-Beranger, D. K. Maude, S. Rumyantsev, and M.S. Shur. Plasma wave detection of terahertz radiation by silicon field effects transistors: Responsivity and noise equivalent power. *Applied Physics Letters*, 89(25):253511–253511–3, Dec 2006.
- [20] Q. Mao, X. Tu, L. Xu, C. Wan, L. Kang, J. Chen, and P. Wu. Effective receiving area of the antenna-coupled terahertz detector. *Optical Engineering*, 53(2):023103, 2014.
- [21] J.W. Fleming. High-resolution submillimeter-wave fourier-transform spectrometry of gases. *Microwave Theory and Techniques, IEEE Transactions on*, 22(12):1023–1025, Dec 1974.
- [22] T.G. Blaney. Infrared and millimeter waves. *Submillimeter Tehniques*, Vol. 3, 1980.

- [23] CEIS. Center for emerging and innovative sciences. <http://www.ceis.rochester.edu/about/index.html>, 2014.
- [24] D.J. Newman, P.P.K. Lee, A.P. Sacco, T.B. Chamberlain, D.A. Willems, R.D. Fiete, M.V. Bocko, Z. Ignovic, J.L. Pipher, C.W. McMurtry, X.C. Zhang, D.B. Rhodes, and Z. Ninkov. Compact thz imaging detector. *Proc. SPIE*, 8716:87160A–87160A–11, 2013.
- [25] D. Arnone, C. Ciesla, and M. Pepper. Terahertz imaging comes into view. *Physics World*, 2001.
- [26] T. May, G. Zieger, S. Anders, V. Zakosarenko, H.-G. Meyer, M. Schubert, M. Starkloff, M. Rößler, G. Thorwirth, and U. Krause. Safe visitor: visible, infrared, and terahertz object recognition for security screening application. *Proc. SPIE*, 7309:73090E–73090E–8, 2009.
- [27] E. Brundermann, H.W. Hubers, and M. F. Kimmitt. Terahertz techniques. *Springer Berlin Heidelberg*, 151., 2012.
- [28] Kumud R.J. and G. Singh. Terahertz planar antennas for next generation communication. *Springer International Publishing Switzerland*, 2014.
- [29] M. Perenzoni and D. Paul. Physics and applications of terahertz radiation. *Springer Series in Optical Sciences*, 2014.
- [30] D. Leisawitz, W.C. Danchi, M.J. DiPirro, L.D. Feinberg, D.Y. Gezari, M. Hagopian, W.D. Langer, J.C. Mather, S.H. Moseley Jr, M. Shao, R.F. Silverberg, J.G. Staguhn, M.R. Swain, H.W. Yorke, and X. Zhang. Scientific motivation and technology requirements for the spirit and specs far-infrared/submillimeter space interferometers. *Proc. SPIE*, 4013:36–46, 2000.
- [31] S. Solomon. Stratospheric ozone depletion: A review of concepts and history. *Reviews of Geophysics*, 37(3):275–316, 1999.
- [32] T. Nagatsuma, S. Horiguchi, Y. Minamikata, Y. Yoshimizu, S. Hisatake, S. Kuwano, N. Yoshimoto, J. Terada, and H. Takahashi. Terahertz wireless communications based on photonics technologies. *Opt. Express*, 21(20):23736–23747, Oct 2013.
- [33] C.K. Sun. In vivo noninvasive thz imaging of blood glucose level. *Asia Communications and Photonics Conference*, page ATh4H.6, 2012.

- [34] Intel Corporation. 3-d, 22nm: New technology delivers an unprecedented combination of performance and power efficiency. <http://www.intel.com/content/www/us/en/silicon-innovations/intel-22nm-technology.html>, 2014.
- [35] M. Kanellos. Moore’s law to roll on for another decade. <http://news.cnet.com/2100-1001-984051.html>, 2003.
- [36] C. Fonstad Jr. Microelectronics devices and circuits,fall 2009. (*Massachusetts Institute of Technology: MIT OpenCourseWare*), <http://ocw.mit.edu> (Accessed 4Jun14). License: Creative Commons BY-NC-SA.
- [37] M. Dyakonov and M. Shur. Shallow water analogy for a ballistic field effect transistor: New mechanism of plasma wave generation by dc current. *Phys. Rev. Lett.*, 71:2465–2468, Oct 1993.
- [38] M. Dyakonov and M. Shur. Plasma wave electronics: Novel terahertz devices using two dimensional electron fluid. *IEEE Transactions on Electron Devices*, 43(10), 1996.
- [39] W. Knap, M. Dyakonov, D. Coquillat, F. Teppe, N. Dyakonova, J. Lusakowski, K. Karpierz, M. Sakowicz, G. Valusis, D. Seliuta, I. Kasalynas, A. Fatimy, Y.M. Meziani, and T. Otsuji. Field effect transistors for terahertz detection: Physics and first imaging applications. *Journal of Infrared, Millimeter, and Terahertz Waves*, 30(12):1319–1337, 2009.
- [40] A. Gutin, S. Nahar, M. Hella, and M. Shur. Modeling terahertz plasmonic si fets with spice. *Terahertz Science and Technology, IEEE Transactions on*, 3(5):545–549, Sept 2013.
- [41] D.B. Veksler. Plasma wave electronics devices for thz detection. *Rensselaer Polytechnic Institute*, 2007.
- [42] D. Veksler, A. Muravjov, V. T. Kachorovskii, Kachorovskii, K. Salama, X. Zhang, and M. Shur. Imaging of field effect transistors by focused terahertz radiation. *Solid State Electronics*, 2009.
- [43] J.Q. Lu and M.S. Shur. Terahertz detection by high-electron-mobility transistor: Enhancement by drain bias. *Applied Physics Letters*, 78(17):2587–2588, 2001.
- [44] T. A. Elkhatib, V. Yu. Kachorovskii, W. J. Stillman, S. Rumyantsev, X.-C. Zhang, and M. S. Shur. Terahertz response of field-effect transistors in saturation regime. *Applied Physics Letters*, 98(24):–, 2011.

- [45] D. Veksler, F. Teppe, A. P. Dmitriev, V. Yu. Kachorovskii, W. Knap, and M. S. Shur. Detection of terahertz radiation in gated two-dimensional structures governed by dc current. *Phys. Rev. B*, 73:125328, Mar 2006.
- [46] D. Wolf and E. Holler. Bistable current fluctuations in reverse-biased p-n junctions of germanium. *Journal of Applied Physics*, 38(1), pp. 189-192, 1967.
- [47] S. Hsu and R. Whittier. Physical model for burst noise in semiconductor devices. *Solid-State Electronics*, 13, pp. 1055-1071, 1970.
- [48] K. Kandiah, M. Deighton, and F. Whiting. "a physical model for random telegraph signal currents in semiconductor devices. *Journal of Applied Physics*, 66, pp. 937-948, 1989.
- [49] Virginia Diodes Incorporated. Vdi website. <http://vadiodes.com/index.php/en/>, Accessed 17Jul14.
- [50] H.T. Friis. A note on a simple transmission formula. *Proceedings of the IRE*, 34(5):254–256, May 1946.
- [51] A. Lisauskas, D. Glaab, H. G. Roskos, E. Oejefors, and U. R. Pfeiffer. Terahertz imaging with si mosfet focal-plane arrays. *Proc. SPIE*, 7215:72150J–72150J–11, 2009.
- [52] Keithley. Semiconductor device test applications guide. *Series 2600A System SourceMeter Instruments*, 2013.
- [53] UC Berkeley. Spice circuit simulation software. <http://bwracs.eecs.berkeley.edu/Courses/IcBook/SPICE/>, 2014.
- [54] D. Blostein and A. Grbavec. Recognition of mathematical notation. *Handbook of Character Recognition and Document Image Analysis*, pages 557–582, 2001.
- [55] Cadence. Cadence spectre. http://www.cadence.com/products/cic/spectre_circuit, 2014.
- [56] W.L. Chan, J. Deibel, and D.M. Mittleman. Imaging with terahertz radiation. *Reports on Progress in Physics*, 70(8):1325–1379, 2007.
- [57] A. Gutin, T. Ytterdal, V. Kachorovskii, A. Muraviev, and M. Shur. Thz spice for modeling detectors and nonquadratic response at large input signal. *Sensors Journal, IEEE*, 13(1):55–62, Jan 2013.

- [58] J.R. Janesick. Photon transfer: Dn to λ :. *Society of Photo Optical*, 2007.
- [59] G. Lepage, J. Bogaerts, and G. Meynants. Time-delay-integration architectures in cmos image sensors. *Electron Devices, IEEE Transactions on*, 56(11):2524–2533, 2009.
- [60] A. Lisauskas, M. Bauer, S. Boppel, M. Mundt, B. Khamaisi, E. Socher, R. Venckevičius, L. Minkevičius, I. Kašalynas, D. Seliuta, G. Valušis, V. Krozer, and H.G. Roskos. Exploration of terahertz imaging with silicon mosfets. *Journal of Infrared, Millimeter, and Terahertz Waves*, 35(1):63–80, 2014.
- [61] P.H. Siegel. Terahertz technology. *Microwave Theory and Techniques, IEEE Transactions on*, 50(3):910–928, 2002.
- [62] Synopsys. Technology computer aided design (tcad). <http://www.synopsys.com/tools/tcad>, 2014.
- [63] M. Tonouchi. Cutting-edge terahertz technology. *NATURE PHOTONICS*, 1(2):97–105, 2007.
- [64] H. Yanming. A terahertz radiation detection system design and device modeling. *University of Connecticut*, 2003.