Interferometric Imaging of Semiconductor Devices

by

Elise M. R. Michaels

Submitted in Partial Fulfillment

of the

Requirements for the Degree

Master of Science

Supervised by

Professor Alan M. Kadin

Department of Electrical and Computer Engineering

The College

School of Engineering and Applied Sciences

University of Rochester
Rochester, New York

1999

Interferometric Imaging of Semiconductor Devices

by

Elise M. R. Michaels

Submitted in Partial Fulfillment

of the

Requirements for the Degree

Master of Science

Supervised by

Professor Alan M. Kadin

Department of Electrical and Computer Engineering

The College

School of Engineering and Applied Sciences

University of Rochester
Rochester, New York

1999

Curriculum Vitae:

The author was born in Niskayuna, New York on 19 December 1967. She attended Hudson Valley Community College (HVCC), Troy, New York, from the Fall of 1986 to Spring of 1988, and graduated with an Associates of Applied Science in Electrical Engineering Technology, Laser/Electro-Optics Option, in 1988. Elise graduated Dean's List from HVCC. She then attended the University of Rochester from Fall 1988 to Summer 1991, when she graduated with the degree of Bachelor of Science in Optics, in 1991. She began her graduate studies at the University of Rochester in the Fall of 1992 in the field of Electrical Engineering. She pursued her research in Interferometric Imaging of Semiconductor Devices under the direction of Dr. William R. Donaldson in 1996 and 1997. She is a co-author of a paper "Integrated Circuit Tester Using Interferometric Imaging," presented at SPIE in conjunction with the Electrochemical Society's Annual Meeting, on 7 May 1997, in Montreal, Quebec, Canada, and published in *Electrochemical Society Proceedings Volume 97-12*.

Acknowledgments:

Dr. W. Donaldson, for all of the discussions about the experimental system, data, and the assistance in programming of PV Wave.

Professor A. Kadin, for the assistance in choosing my path, for the discussions of what else needed to be included in the Thesis and for the educational support.

Robert L. Michaels, my husband, for all the assistance around the house and taking care of our 5-year-old and 2-year-old daughters, while I acquired the data, processed the data, and wrote my thesis. Also, thank you for the assistance in the first two revisions of my thesis and the constant encouragement.

John Larkin, I learned a lot about the laser from you and you definitely made me think about how the system works, both the laser and my experimental setup. I know it was hard sharing the laser at the odd hours when I needed it quickly. Thank you for taking the time to read my thesis in an attempt to make it more presentable. Also for the practice at "Photons after Dark" to present my thesis before I defended it.

Doug Jacobs-Perkins, I appreciate the assistance in writing the computer program, the in-depth discussions on what performance was generated by the system, and the practice presentations.

Heidi Natel, If you hadn't been there for me, I wouldn't have been able to get this far. Thanks for the assistance with the finishing touches for my thesis.

This work was supported by the U.S. Department of Energy Office of Inertial Confinement Fusion under Cooperative Agreement No. DE_FC03-92SF19460, the University of Rochester, the New York State Energy Research and Development Authority, and an SBIR contract with Optimetrix from the U.S. Air Force Phillips Laboratory. The support of DOE does not constitute an endorsement by DOE of the views expressed in this thesis.

Abstract

A noninvasive optical diagnostic technique is described for testing the electrical state of an integrated circuit. The interferometric probe is capable of distinguishing between the on- and off-state of an N-Channel silicon FET with micron-size dimensions. The interferometric probe consists of a Twyman-Green interferometer with a large collimated beam that allows a large active area of the substrate to be examined. A sub-band gap, picosecond optical pulse is used to determine local variations in the index-of-refraction caused by dopants, heating, and changes in carrier concentrations under dynamic operating conditions. Interferometer imaging of the semiconductor is accomplished through the back surface (bottom of the chip), which is useful in examining flip-chip devices. Preliminary results demonstrating this technique using an Intel chip are presented, and possible improvements are suggested.

Table of Contents:

List of Tables:

List of Figures:

I: Introduction:

Semiconductors have become very widely used in society. They are found in every home, in products such as televisions, radios, home computers, and in every work place, such as supermarket scanners and gas pumps. The integrated circuit (IC), one of the most widely used semiconductor technologies, is becoming one of the more difficult items to test, due to several factors including flip-chip technology, packing density, and trace size. In order to increase the speed of the semiconductor integrated circuit, its features have been made smaller and smaller. Microprocessors are presently being tested with microprobes, which make direct contact with the chip. However, the microprobe strongly perturbs the high frequency signal in the circuit, and it is also difficult to couple the output signal with the probe. This difficulty in testing has increased the desire to develop less invasive testing techniques. A new, interferometric, noninvasive test has been designed and tested. It involves the understanding of absorption coefficients and the index-of-refraction of a semiconductor device. The interferometer is an optical instrument which measures the changes in index-of-refraction. The phase of light reflected from the IC changes as the voltage applied to each MOSFET is varied between the quiescent state and applied voltage states, both with the gate and drain voltages on and off. Phase is measured by interfering light reflected from the IC with light from a reference arm; thus changes in voltage states affect the reflected light. It is this change which will be used to determine whether or not the device under test (DUT) is good or bad. This test is noninvasive and uses the combination of the interferometer and a normal chip socket to apply voltages through the normal chip pins. This system is extremely useful with the flip-chip technology since the DUT is probed from the polished backside of the chip. The probe beam is a picosecond infrared laser source. This picosecond source allows for the timing of the laser pulse to occur within the period of time that the gate pulse is on, so that the effect of the drain activity is recorded and can be analyzed.

The basic related theory of semiconductors and interferometers is presented. The theory is followed by a review of other diagnostic techniques for semiconductor devices, and the present experimental setup. The experimental performance is examined, and its implications for the further development of this technique discussed.

II.a: Semiconductor Theory:

A brief review of semiconductor theory is provided, so that the relationship between the absorption coefficient and the index-of-refraction may be understood. The experiment described here effectively measures the absorption coefficient and index-of-refraction of a semiconductor device. These optical properties are sensitive to heat due to electrical flow, dopants in the device's structure, and changes in carrier concentration due to injection.

A semiconductor has two bands: the valence band and the conduction band (Fig. 1). An increase in conductivity may be obtained when electrons from the semiconductor's filled band (valence band) jump to the unoccupied or slightly occupied band (conduction band).[1,2]

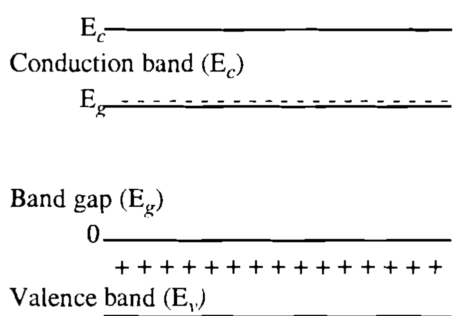


Fig. 1: The energy bands of a semiconductor.

The intrinsic carrier concentration[3,4] is described as $n=p$, where n is the intrinsic carrier concentration of electrons in the conduction band and p is the intrinsic carrier concentration of holes in the valence band. The bandgap, band edge masses and the temperature all influence the intrinsic carrier concentration. There are formulas to solve for the intrinsic carrier concentration:

$$n = N_c \exp\left(-\frac{E_C - E_F}{kT}\right) \qquad 1.1$$

$$p = N_V \exp\left(-\frac{E_F - E_V}{kT}\right) \qquad 1.2$$

$$E_F = E_i = \frac{E_C - E_V}{2} + \frac{3kT}{4}\ln\left(\frac{m_p}{m_n}\right) \qquad 1.3$$

where $N_C = 2\left(\frac{k_B m_n T}{2\pi \hbar^2}\right)^{3/2}$ is the density of states in the conduction band, $N_V = 2\left(\frac{k_B m_p T}{2\pi \hbar^2}\right)^{3/2}$ is the density of states in the valence band, E_i is the intrinsic Fermi level, m_n is the mass of an electron, m_p is the mass of a hole, E_v is the energy of the valence band, and E_c is the energy of the conduction band.

In the case of a direct bandgap material, like GaAs, when light is directed onto the semiconductor, the light may excite an electron in the valence band to move to the conduction band. This process occurs in Si (with its indirect band gap) only if there are four particles (electron, hole, photon, and a phonon) interacting (See fig. 2):

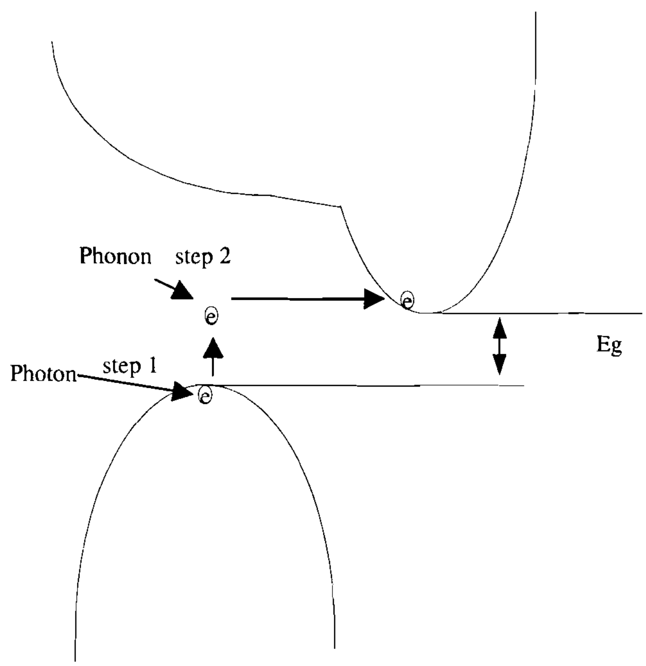


Fig. 2: Band structure of Si, and the movement of an electron from the valence band to the conduction band.

The absorption band edge is smeared out in energy. The minimum photon energy needed to create an electron-hole pair in silicon is slightly higher than the bandgap energy, since some of the energy of the photon is used to create a phonon. Phonon energy is dissipated into the device as heat. The absorption may be measured in three ways: transmission measurements, spectroscopic ellipsometry, and normal-incidence reflectance measurements.[5] We chose to use the transmission method because it was readily available.

The absorption coefficient may affect other optical constants of a semiconductor device. One of the more important constants to be determined is the index-of-refraction of the device. This measurement may be made in either reflectance or in transmission. If the photon energy is too high, the device will become opaque unless the thickness of the semiconductor is very small. The absorption coefficient (α) is also known as the optical penetration depth when it is inverted. The absorption coefficient is denoted in the following formula

$$\alpha = \frac{4\kappa\pi}{\lambda_0} = \frac{2\omega\kappa}{c} = \frac{\omega\varepsilon_2}{nc} \qquad 1.4$$

where c is the speed of light, ω is the frequency of the light, λ_o is the wavelength in free space, n is the index-of-refraction, and the dielectric field $\varepsilon_2 = 2n\kappa$, where κ is the imaginary component of the complex index-of-refraction[6,13,14] ($n+i\kappa$). The Kramers-Kronig dispersion relation is used to relate the real and imaginary components of the complex index-of-refraction.[6,7,8,9] The response functions of a medium are a combination of the susceptibility and dielectric function. These response functions allow the determination of a medium's response to certain stimulus.[6] This relation is also dependent on polarization of a monochromatic wave. The Kramers-Kronig relation is useful in determining the change in the index-of-refraction given the absorption. [7,8,9]

Heating of the device, partially caused by currents, is another aspect of semiconductors influencing the index-of-refraction.[10] Heat is generated as energy is lost, for instance when a charge is passed through a resistance. This conversion is known as Joule heating.[3,11] The electrical conductivity changes as Joule heating affects the device. This change affects the index-of-refraction, by changing the carrier density in eq. 1.1. This is to say that if the temperature increases the intrinsic carrier density increases, which from eq. 1.1 and eq. 1.4 will increase the index-of-refraction. The change in index-of-refraction is given by the following formula:

$$\Delta n = \frac{n \Delta \mathrm{N} \mathrm{e}^2}{2 \varepsilon \omega^2 \mathrm{m}^*} \qquad 1.5$$

where Δn is the change of the index-of-refraction, ΔN is the excess free carriers, e is the electron free charge, ω is the angular frequency of the light, m^* is the carrier effective mass, and ε is the dielectric constant.[12] A submicron FET can have an excess free-carrier concentration of 10^{18} to 10^{20} cm^{-3} extended to a depth l of 0.25μm into the bulk. Inserting numbers for silicon and 1-μm radiation, the change in index-of-refraction is $\Delta n = 10^{-21} cm^{-3} \Delta N$.

The absorption coefficient may be used to calculate the transmitted intensity by using the following formula:

$$I = (1 - R)^2 I_0 e^{-\alpha l} \qquad 1.6$$

Where I_o is the incident intensity, R is the reflection and l is the thickness of the material. [6] The reflection, R, is given by[13,14]:

$$R = \frac{[n-1]^2 + \left[\alpha c / 2\omega\right]^2}{[n+1]^2 + \left[\alpha c / 2\omega\right]^2}. \qquad 1.7$$

The index-of-refraction and the absorption coefficient are also used to calculate a phase change, $\Delta\psi$, by using the following equation:

$$\tan\Delta\psi = \frac{2n}{n^2 + \left(\alpha c / 2\omega\right)^2 - 1}. \qquad 1.8$$

Semiconductors are usually described by characteristics, such as the carrier concentration, dielectric constant, and excess free carriers. These characteristics may vary spatially within a semiconductor crystal. An n-channel transistor that we examined is operated in enhancement mode, which means it is normally off. The n-channel is generally preferred over the p-channel device since the electron mobility in Si is larger than the mobility of holes (n>p, $\mu_e > \mu_h$) and electrons are the primary current in the n-channel device. For an example of this see fig. 8.8 of B. G. Streetman's *Solid State Electronic Devices, fourth edition.*[15] The transistor acts as an on/off switch, and requires that a voltage be applied to the gate in order to induce a conduction channel between the drain and the source. If a voltage is applied to the drain the electrons will flow from the source to the drain along the conducting channel. When the conduction channel is constricting the path along which current flows and the drain current appears to saturate, the pinch-off point of the device has been reached. Beyond the pinch-off point the drain current remains the same. The drain current is controlled by a gate electrode, which is insulated from the source and drain by an oxide layer. When no voltage is applied to the gate, the drain and the source act like two p-n junctions, which are connected back to back. Therefore, a MOS circuit's dc input impedance may be very large. The gate electrode generally uses either polysilicon or a combination of silicide and polysilicon. We are interested in the case of Polysilicon.

MOSFET's vary in size, from large power devices to devices that must be scaled down in order to increase the number of components per IC. As the component shrinks there is a problem that arises, which is the short-channel effect. In this effect, as the channel length is reduced the depletion layer width (areas with reduced carrier concentration), of the source and drain junctions become comparable to the channel length. When the drain and source widths equal the channel length, punch-through will occur, whereby the gate loses control of the current, thus limiting the device's operation for short-channel MOSFET's (a current-voltage characteristic curve may be examined in fig.9.10 of J. Singh[3]). The small MOSFET's size, and its material characteristics make it difficult to test an individual device imbedded in an IC with conventional methods. Another important point is that the carrier concentration relates to the absorption and the index-of-refraction, both of which are related to optical properties. This allows the semiconductor at least partially to be tested optically.

II.b: Interferometry Theory:

Interferometers have been used for many different types of measurements. The index-of-refraction is one of the most commonly measured quantities. An interferometer is a device, which divides a light wave into two different beams, not necessarily with the same amplitude. They each propagate through different arms of the interferometer, and are then recombined to form a single beam, where measurements are made by studying the interference pattern of the combined beam. There are different types of interferometer; the one to be described is the Michelson interferometer.

The Michelson interferometer will have a sinusoidal interference pattern when there is a path length or time change.[16,17] The amplitude increases and decreases with τ for a single frequency, where $\tau = \frac{2n\Delta l}{c}$ is the delay between the two arms of the interferometer, and Δl is half the shift of the path length change. The Michelson Interferometer is a linear system.[18] This is to say, as the input intensity is increased, the output intensity is increased by the same factor. It is also known that no interferometer system is able to completely copy the input of the system to the output of the system. This loss is due to smoothing from the system.

The main feature of the interferometer is the generated fringe. These fringe patterns lead us to the Huygen's and Fresnel theory. These fringe theories allow an approximation of the *U* scalar to be given as:

$$|U(x)| = \sqrt{[I(x)]} \qquad 2.0$$

where *I(x)* is the intensity as a function of position within the interference pattern, and *U(x)* is the complex amplitude of Maxwell's equations. At a point *x* and a time *t* the actual scalar components of the field vector are rapidly fluctuating real functions of ($e^{(ikx-i\omega t)}$). This rapidly fluctuating field will lead to the intensity, which in turn will generate the absolute complex amplitude when the square root of the intensity is taken. The scalar wave function is usually adequate in describing the electromagnetic radiation. In general, electromagnetic radiation implies a vector field which in turn satisfies Maxwell's equations. Radiation which has a small bandwidth as compared with the mean frequency ν_o, $\Delta\nu \langle\langle \nu_o$, is defined to be quasi-monochromatic.

The interferometer produces images of some object plane. A coherent object occurs when an object plane contains a plane of variable transmission, which is illuminated by a small source with a constant phase. At the image plane the complex amplitudes from each object point add. The total image is obtained as the convolution of the amplitude distribution over the object $U(\xi)$ and the amplitude-spread function $F(\xi)$ is given by

$$U(\rho) = \left(\frac{A}{\lambda}\right)\sin\varphi\sin\varphi' \int_{-\infty}^{\infty} f(\eta)\exp\{(2\pi i\eta)(\xi'-\xi)\}\partial\eta \quad 2.1$$

$$U(\rho) = \left(\frac{A}{\lambda}\right)\sin\varphi\sin\varphi' F(\xi'-\xi) \quad 2.2$$

$$F(\xi'-\xi) = \frac{\lambda U(\rho)}{A\sin\varphi\sin\varphi'} \quad 2.3$$

where A is a constant, $sin\ \varphi$ and $sin\ \varphi'$ are the numerical apertures in object and image space, λ is the wavelength, $\xi = (x\sin\varphi)/\lambda$, $\xi' = (x'\sin\varphi')/\lambda$, $\eta = u/(z\sin\varphi)$, $\eta' = u'/(z'\sin\varphi')$. This is the spread function for coherent imagery. Its transform, the pupil function[19]

$$f(\eta) = \exp(ik\omega(\eta)) \quad \text{over } \mathcal{A} \quad 2.4$$

$$f(\eta) = 0 \quad \text{elsewhere} \quad 2.5$$

is then the transfer function, where $\mathcal{A}$ is the area of the pupil.

It should be pointed out that since light is a transverse wave it could exhibit polarization effects. By examining a wave propagating along the z-direction, it is apparent that separate treatments are needed for the wave's two components of the electric vector, and therefore for the two complex analytic signals E_x and E_y which are associated with them.

There are two types of coherence, temporal and spatial. Both types of coherence are manifested in the Michelson interferometer. The first type of coherence to be discussed is temporal. Temporal coherence characterizes the phase variations in time from its initial state. The Michelson interferometer arms act as delay lines for the divided amplitude of the source beam. This delay allows the exploration of the phase relationship between light emitted at different times. The source frequency distribution and the wave's properties, such as the intensity of the source and path differences, are associated with temporal coherence as the pulse propagates along the optic axis. The temporal coherence appears as a cosinusoidal variation of intensity in time. This cosinusoidal variation can be observed by changing the path differences, as well as with time differences $\tau = \frac{\rho}{c}$ for the two arms. The total phase variation is $1+\cos(2\pi\nu\tau)$ for radiation of a single frequency $\nu = \frac{c}{\lambda}$. For a range of frequencies, the total intensity is proportional to the integral over all the relevant frequencies. The cosine transform G_c for a source has a bandwidth $\Delta\nu = (\nu - \nu_o)$, where ν_o is the initial frequency. Therefore

$$I(z) = G_c(o) + G_c(\tau)\cos(2\pi\nu_o\tau). \qquad 2.6$$

The modulation is still sinusoidal for a single frequency, but the amplitude, G_c decreases with τ. There is a limited temporal range over which fringes may be observed. The radiation temporal coherence increases as the bandwidth decreases.

The second type of coherence is spatial coherence. Spatial coherence, unlike temporal coherence, is associated with the wave's properties transverse to the direction of propagation. It is dependent on the wave front of the light. A plane wave has a uniform phase in a plane perpendicular to the direction of propagation. If the plane wave is perfect it is said to be spatially coherent. Spatial incoherence is associated with departures of the wave from the ideal plane wave. It should be noted that both temporal and spatial coherence are equally important. The spatial coherence is apparent with the Michelson interferometer. Spatial coherence may be demonstrated with a source placed a distance x away from a plane with two pinholes (p1 and p2). These pinholes will act as two separate point sources, which simulate a plane wave and will interfere with one another further down the system, at the object plane, generating an interference pattern similar to an interferometer. These pinholes are placed fairly close to one another (see fig. 3). The light then passes through this plane which causes a diffraction pattern on the next plane placed y away from the first plane. This is known as the Young's interference experiment.

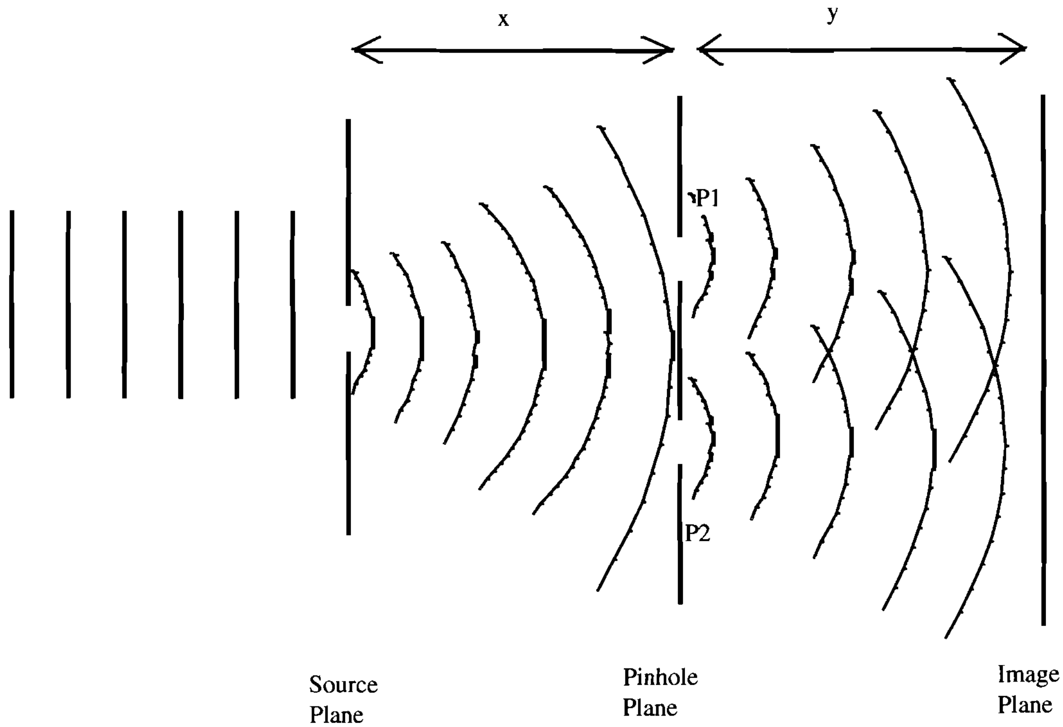


Fig. 3: Young's experiment

The interferogram is a total irradiance from a maximum value to a minimum value, where the maximum is known as total constructive interference, and the minimum is known as total destructive interference.[19] Constructive interference occurs when the phase angle, ψ, is equal to even multiples of π ($0, 2\pi, 4\pi, \ldots$), while the destructive interference occurs when the phase angle, ψ, is equal to odd multiples of π (π, 3π, $5\pi, \ldots$). This is demonstrated in the formula of total irradiance, I,

$$I = I_r + I_t + 2\sqrt{I_r I_t}\cos(\psi) \qquad 2.7$$

where ψ is a phase angle, I_r is the reference arm irradiance, and I_t is the test arm irradiance.[15, 19, 25] This formula allows the phase to be calculated. By determining the relative amplitude of I_r and I_t the loss, which is related to the absorption can be determined. The absorption and phase can assist in determining the index-of-refraction and the free carrier concentration of a semiconductor device, as will be demonstrated below.

Before the irradiance curve produced by eq 2.7 may be used the alignment of the interferometer must be optimized. Once the optimization is complete eq. 2.7 is used to determine the fringe sensitivity of the interferometer, which indicates how much of a wave difference is apparent. The phase delay $\Delta\Psi$ (fringe sensitivity) follows from the comparison of a set of data which is unbiased and a biased set of data. The difference between two measurements ($\Delta\psi$) is used to obtain the change ΔN in the free carrier concentration. This is calculated from the semiconductor theory by

$$\Delta N = \frac{\varepsilon\omega^2 m^* \lambda}{q^2 n \pi d}\Delta\psi \qquad 2.8$$

where d is the thickness of the semiconductor, and $\Delta n = \frac{\lambda}{2\pi d}\Delta\psi$ is a variation in the index-of-refraction. The free carrier concentration is related to the index-of-refraction from eq 1.6. Therefore this measurement requires the knowledge of the index-of-refraction, n, and the equivalent effective mass in Si, m^*, which is approximated by

$$m^* = \frac{\hbar^2}{\left(d^2\mathcal{E} / dk^2\right)} \cong 1.1m.^{2,15,20,21} \qquad 2.9$$

Two-beam imaging interferometers are affected by several system variables. The system variables are: shear, s, which is a lateral separation of two images (these two images are generated by the two arms of the interferometer), shift, h, which is the longitudinal separation of two images, tilt, t', which is a lateral separation of two images of a point on the source, and lead, l', which is the longitudinal separation of two images of a point on the source. Finally, there is a delay, τ, between the two arms of the interferometer, which is caused by different path lengths as well as propagation through different indices of refraction. Ideally, the two arms of the interferometer should transmit equal intensities, I_r and I_t, and the two beams should not have any changes in their polarization; this, however, never occurs in practice. These conditions can be resolved in several ways.

The Twyman-Green interferometer is an upgraded Michelson interferometer.[22] Both of these systems have similar optical components as shown in fig. 4.

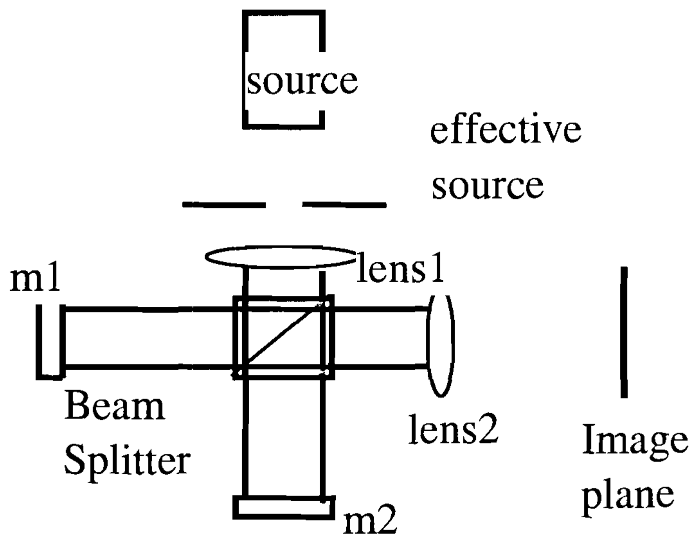


Fig. 4. Michelson Interferometer

The components shown in fig. 4 are as follows: The lenses are representative of collimating systems, one at the input and one at the output. The lenses are located a distance d from the reference position at the end of each arm. A stop at the focus of $lens_1$ (f_1) is the effective source of the system. The interferogram is viewed a distance of $g=f_2^2/(d-f_2)$ from $lens_2$'s focus (f_2). The m_2 mirror is a distance e (a small distance) away from the reference position with a rotation of θ about its normal axis. The Michelson interferometer design was changed to work with collimated light. Defects in a specimen are observed by differences in optical path, which appears as a departure from fringe straightness. The Twyman-Green interferometer allows a whole substrate to be imaged and simultaneously analyzed. Since this interferometer may use a short pulsed laser, a short photographic exposure time is possible, thus reducing system vibration effects.

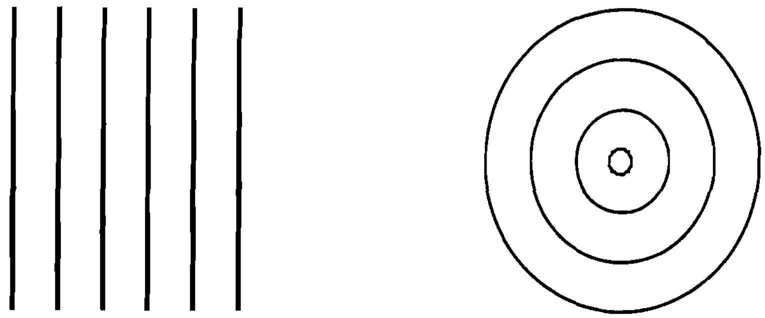
The interferogram is viewed from what is called the plane of observation. This plane is located a distance, z,

$$z = \frac{(f_1)^2}{f_1 + f_2 - 2d + \frac{(f_2)^2}{g}} \qquad 2.10$$

from the source. A shear is represented by $s = 2\theta(z - f_1 - \frac{zd}{f_1})$, where θ is the angle of tilt on the mirror with-respect-to the incident beam and a shift represented by $h = \frac{2ez^2}{(f_1)^2}$. The shear vanishes when $g = \frac{(f_2)^2}{(d - f_2)}$, and $z = \frac{(f_1)^2}{(f_1 - d)}$. It has been determined that the shift is always present, even if the shear is not. This is where the Fizeau fringes or test fringes are localized.[23] With the condition listed for g and z above, the plane of observation should be at an equivalent image plane of the mirrors.

The Michelson interferometer is able to display Haidinger fringes as well as Fizeau fringes. Haidinger fringes are an image of the source that is at infinity. They are only visible if the first set of fringes have been spread out completely or if the aperture of the mirrors or other test surface are stopped down so that the aperture is smaller than the fringe spacing. The Haidinger fringes are centered circles of the source and are known as fringes of equal inclination. These fringes lose contrast as one of the mirrors or test devices is moved further away from the other. Another effect is that the spacing of the fringes is decreased at the same time the fringes lose contrast. Haidinger fringes are present when there is no shear present. They may be dependent on either source radius of a centered source or width for an annular source off center.

Fizeau fringes occur when there is tilt. These Fizeau fringes are straight lines that are parallel to the axis of tilt and are known as fringes of equal thickness. Fringes of equal thickness are contours of the wedge between one mirror and the image of the other which is reflected by the beam splitter. Tilting either mirror will smooth the fringing in order to remove the wedge. The Hansen's criterion will give reasonable visibility of the Fizeau fringes and the Haidinger fringes with a Michelson interferometer. The Hansen's criterion is a good alignment tool, but should be avoided when taking the images of the device under test, due to the possible influence of the fringes on the phase changes. In some cases visibility of these fringes may vary with the shape of the source. The fringe patterns are important since they will help with the alignment of the interferometer. The Fizeau Fringes are adjusted by tilting one of the mirrors. The Haidinger fringes are adjusted by adjusting the length of one of the interferometer arms.



Fizeau Fringes Haidinger Fringes

Fig. 5: Fizeau and Haidinger Fringes

Defects and aberrations may be measured with an interferometer.[24] An interferometer is able to measure irregular surfaces, non-uniform indices as well as bad design of mirrors or test devices. These errors appear as differences in the image from the object (source). Since aberrations and wavelength comparisons have little effect on the image quality, an accuracy of approximately 0.1 of a focal length is easily obtained for optical testing. The adjustment of the fringe patterns creates a more stable image (no fringe moving with air turbulence and optical mount instabilities). This will help with the determination of the phase calculation and the index-of-refraction with respect to the carrier concentration.

III: Previous Diagnostic Techniques:

The technology to test integrated circuits (ICs) has made major advancements over the last 30 years. The technology has used a variety of measurements, from small signal conductance, to MOSFET parameter extraction using conduction. Further approaches include voltage fluctuations of a constant current power supply, interferometric measurements of sheet charge density and electron-hole pair recombination. The limitation of these techniques has demonstrated a need for non-invasive area probing especially for flip-chip technology. This is the motivation that led us to develop interferometric measurements for several devices within an IC. Some of the more pertinent diagnostic techniques leading to the noninvasive infrared interferometer test method will be discussed. We used several features from different diagnostic techniques to establish our noninvasive technique, which we expect will be able to detect localized heating in an IC.

One of the initial methods of examining self-heating in a CMOS IC was developed by G. Ghibaudo.[27] The conduction method uses the drain current and the transconductance to obtain the MOSFET parameters, such as the threshold voltage, the low field mobility and the mobility attenuation coefficient. By using the MOSFET parameters the heat removal paths may be examined to determine the heating affects on the IC. The measurement of accurate direct conductance at high frequencies requires a special chip carrier, by which a drain voltage is applied via a microstrip line, while a ground plane is used for the source. An external static gate bias is supplied with a decoupled thin film microwave capacitor mounted as close to the sample as possible. The effects of the bonding pads and wires are experimentally compensated for, by using the on-chip calibration standard. This technique may be used on any chip layout, although it is desirable to characterize these effects using a normal chip layout (no special measurement paths). The conduction technique generates results comparable to thermal resistance values obtained from noise thermometry and gate resistance measurements of similar devices.[25] Due to the lack of accurate information about the thermal properties of various materials used in a CMOS process, the conduction technique is difficult to use. There are several issues with this technique: channel temperatures do not instantaneously follow the device power dissipation due to the devices' finite thermal capacity, the conduction must be measured over a wide frequency range to get the thermal time constants, capacity of test equipment may be exceeded due to the time period, ambient temperature during the measurements must be carefully controlled for consistency in the data, and at high frequencies the small drain conductance of a MOSFET becomes increasingly difficult and introduces a large imaginary part in the observed impedance. Additionally, the measurement of self-heating in a SOI MOSFET requires a special design layout to measure the thermal capacitance of the device. It is desirable to characterize these effects using a normal layout configuration. One investigation measured the effects of Silicon Dioxide (SiO_2)[26] films of varying thicknesses. This is important since the SiO_2 is often part of the heat removal path for IC's. This experiment verified the thermal conductivity values of IC's.[27,28,29] The MOSFET values for threshold voltage, low-field mobility and mobility attenuation coefficient, are used with the drain current and the transconductance transfer characteristics enabling the user to determine other MOSFET values for the DUT.

Thermal properties of MOSFET devices can be examined using Laser Scanning Microscopes (LSM).[30,31] This is a simple method providing reliable values of the threshold voltage, the low field mobility and the mobility attenuation coefficient. LSM is confocal requiring a pinhole and is therefore diffraction limited. This method requires the overlaying of several images, such as the fluorescence image of the photoresist and the brightfield image where the photoresist is barely visible, to add additional contrast to the images. In general, top-side imaging of a DUT tends to destroy it, and most of the devices are only visible from the back side due to the metallization layer. This is a point-by-point probing of a small area of the DUT. There are several issues that need to be addressed with the LSM system, such as the proportion of the laser radiation which is absorbed as the laser probes the DUT. This is difficult to determine, due to the sharp increase of the absorption coefficient near the absorption edge. The LSM needs to have precise motor control in order to use the confocal LSM technique. The LSM has three parts: optical, electronic and image acquisition. The optical system consists of two He-Ne lasers of which only one is injected at a time. The laser propagates through an Acousto-optical modulator, into a pick-off experiments in LSM were performed by B. Bossmann, et al,[31] H. Komoda, et al,[32] and the Center for Photonics Research at the Boston University.[33] The additional experiments have made minor improvements to the LSMs being used to probe ICs. The LSM technique has been instrumental in improvements to optical probing techniques of ICs. The Optical Beam Induced current (OBIC), which is a part of the LSM does not always accurately detect the failure locations. The OBIC imaging has limitations such as it is not able to probe internally on new chip packages such as the flip-chip packaging. There is also a non-uniform light transmission through the top layer of the IC. This non-uniform transmission is caused by the metallization layers which interfere with the IR sources probing from the top-side. This is easily solved by probing from the back of the IC.[34] Probing from the back has the advantage that an image is used to determine the state of an IC, and for the most part is a noninvasive test method. There are several problems with the method used to acquire the data, in that the probe beams required have been known to damage the DUT. The use of several lasers to acquire the data and analyze the effects is inconvenient. This is where a single source would be simple. The present LSM requires that two images be added which has additional chances to induce error in the data analysis, due to image misalignments. The images need to be overlapped due to the low intensities reflected and transmitted by the system. Without this feature accurate location of defects is difficult.

Variations on the LSM method induced voltage alterations allowing the user to cause less damage to the IC. The voltage alteration techniques are fairly new and take advantage of the interactions between a scanning electron or photon beam and a biased IC. The charge carriers are separated by built-in potential between areas of different Fermi levels. By biasing the IC the Fermi level will change, therefore altering the magnitude of the electron-hole pairs. These systems can quickly localize open interconnections within an IC. The LIVA (Light-Induced Voltage Alteration) uses a He-Ne laser (at 633nm), as well as an external 1152nm He-NE laser and an external 1064nm 1.2W Nd:YAG laser to probe the electrical stimulus for the IC. The LIVA uses the photon generation of electron-hole pairs to generate information about the ICs defects and functions. The LIVA has the additional advantage that the IC voltages are easier to use then the IC currents. The LIVA has a disadvantage that it is unable to probe past an optically opaque layer of an IC, due to the prevention of photon transmission. This disadvantage would make it impossible to probe flip-chip devices with the visible laser, although LIVA probing of flip-chip devices is possible with an IR laser source. An additional disadvantage of the LIVA is probing the back of a chip is more difficult than using infrared (IR) reflected light microscopy. A similar experimental method is the Charged-Induced Voltage Alteration (CIVA) which uses an electron beam instead of a laser (LIVA) to locate areas with defects within the DUT. The electron beam energy (EBE) is set by biasing the DUT with a constant voltage source while the electron beam is off. The IC current is monitored as the electron beam energy is increased. The EBE varies due to the DUT location within the IC, the deeper the MOSFET or contact the higher the EBE. A disadvantage is that LSM causes radiation damage to the DUT, which may be avoided by making sure that the energies from the electron beam are not higher than needed to probe the desired area of the DUT. An advantage is the voltage alteration test method should be more sensitive and produce better spatial resolution than an LSM alone. The IC must be operated with a constant current source, while the supply voltage is monitored. [35] This system may be used for analog or digital signals and has a very high spatial resolution, and a bandwidth is limited only by the DUT. Since this method conveniently uses an image to analyze the defect location this imaging theory will be useful for future experiments.

Another useful method of noninvasive probing of the internal nodes of an IC was done by J. Wiesenfeld.[36] Wiesenfeld uses an electro-optic sampling technique, which is sensitive to voltage changes. The voltage sensitivity relies upon the electric field produced in an Electro-optic crystal by the signal voltage of the device under test. Birefringence in the Electro-optic crystal changes the state of the polarization of an ultrashort-duration laser pulse allowing the electric field to be sampled. This technique was expanded to do sub-picosecond responses of electronic devices, by W. Donaldson, et al.[37] Improvements to the noninvasive test methodology of Electro-optic probing continues. Such improvements are the use of femtosecond probe pulses and bringing the Electro-optic crystal closer to the points being measured.[38,39,40,41]

An interferometric technique by Breglio, et al, has been used to measure the recombination lifetime of electron-hole pairs. The optical measurement is advantageous in that it is contactless, and thus noninvasive. The time resolution has been increased due to short pulse lasers. This allows the time resolution to be on the order of sub-picoseconds. The optical methods are all reliant on monitoring the absorption coefficient induced by the free carriers.[42] In general, the carrier concentration is estimated. When there is an excess of free carriers within a semiconductor, the optical properties vary as well.[36] This contactless nondestructive method uses a probe beam and an excitation beam. This allows for the separation of the absorption and modulation frequency.[43]

The desire for an IC test system which is easy to use and provides useful data is still in demand. The test system should be noninvasive and nondestructive; it should be able to probe flip-chip technology, as well as probing a large chip area at one time. An interferometer such as the Twyman-Green is ideal for testing ICs, since it is a noninvasive and nondestructive probing technique. A short pulse laser determines the temporal resolution. Submicron spatial resolution may be obtained with appropriate optics. The laser pulse is split by a beam splitter to propagate in two different arms of the interferometer, one is a reference arm while the other is the DUT. The DUT needs to be biased by an external power source and a pulse is applied to the gate of the MOSFET. The timing of the gate pulse to the laser pulse is important, so that the operational effects of the device are captured. By using an Infrared laser, light will be able to probe through the backside of an IC. The metallization layers of the DUT make probing the IC the only way to detect intensity changes. The detected intensity changes of the DUT relate to the absorption coefficient, the phase change and the index-of-refraction. These changes in absorption, phase, and index-of-refraction relate to the operational state of the DUT.

IV: Experimental System:

The experimental laser system is shown in Fig. 6, followed by a description.

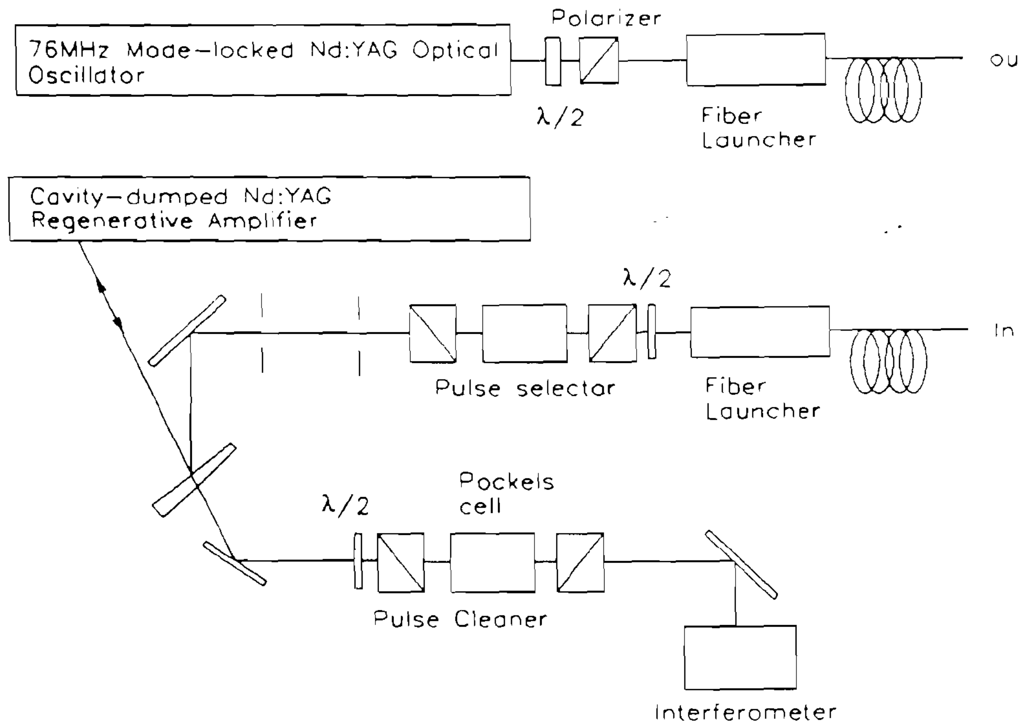


Fig. 6: The Laser system

The source for the laser system is a Nd:YAG 76MHz Mode-locked optical oscillator, cavity-dumped regenerative amplifier, up to 5-Hz power amplifier and SRS cell for optical pulsed shaping, as shown in fig. 6. The oscillator operates at 1.064μm. The regenerative amplifier is seeded with 100pS pulses, with energies less than 1nJ, which are amplified and switched out of the cavity-dumped regenerative amplifier (regen) with an output energy of approximately 150μJ. The output of the regen is a 200pS Gaussian pulse. This was measured with a visible range streak camera (the wavelength was frequency doubled with a KDP crystal). A series of Pockels cells are used to switchout a single pulse to inject into the Twyman-Green interferometer. Prior to the interferometer a half-wave plate and polarizer create a throttle that allows the user to determine the energy being released into the interferometer system.

The system timing is triggered by the General Electric CID510 (charge-injected device) camera, which was used to acquire the data from the interferometer. This timing trigger also supplies triggers to diagnostics, as well as to the gate generator used to supply a gate pulse to the Intel chip under test. The system is operated at a repetition rate of 30Hz.

The interferometer used for this experiment was a Twyman-Green (see fig. 7).

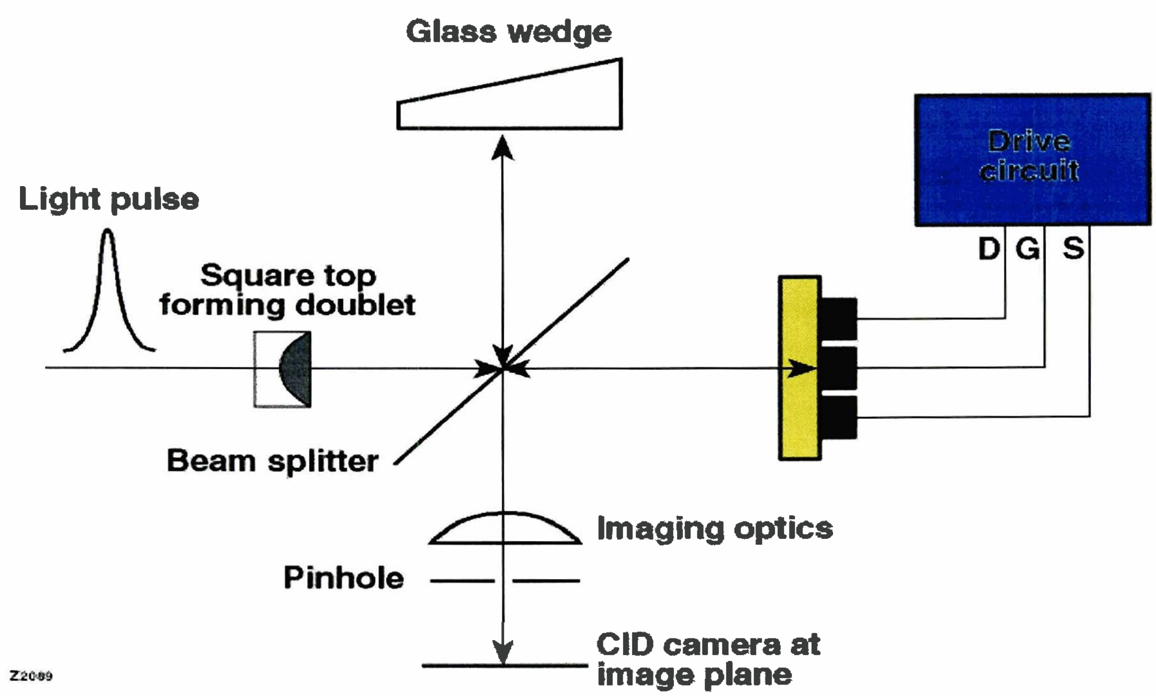


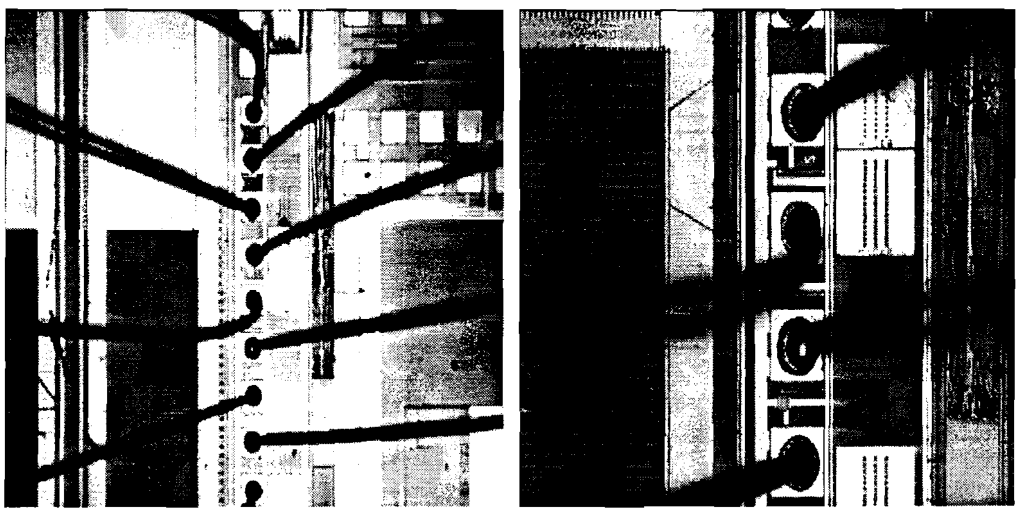
Figure 7: The experimental system which uses a Twyman-Green interferometer.

The interferometer consists of a 50/50 beam splitter, to divide the beam to propagate to both legs. The reference arm has one of the following in it: a glass wedge, a bad Intel chip, or a bad Intel chip which has an anti-reflectance coating. There is also an energy diagnostic, which is an EG&G FND-100 silicon photodiode, located behind the reference. The diagnostic output is sent into an SRS gated integrator, which is also triggered from the camera. The SRS gated integrator's output voltage monitor is fed into the input voltage monitor for the Asyst program. This allows a weighted comparison of the final data, to compensate for fluctuations in the regen output energy. The other arm has the device under test (DUT), which acts as a mirror. The DUT is an Intel Silicon microprocessor chip, in a 68-pin chip package, mounted so that the laser probes the back side of the chip. The back side of the chip has a hole in the package exposing several MOSFET's and a polished surface. Intel Corporation provided the microprocessor chips used for this experiment. The interferometer propagates a percentage of the input intensity to the devices under test. The devices reflect a percentage of the energy back to the beam splitter, where the reference beam recombines with the test beam. The recombined beams propagate the final distance through a microscope objective and an aperture at the focal plane of the lens, onto the CID camera where it is imaged. The GE CID512 camera has an array size of 512 x 512, with a pixel size of 15μm x 15μm. The imaging is done so that the drains and gate appear crisp and well defined. The images of the drains need to be as large as possible. The test uses the results from a dead Intel chip to establish what changes to the index-of-refraction occur. These changes should be minimal compared to an operational (good) Intel chip. The expected influence from the applied voltage for bad drains are: the index-of-refraction will have a minimal change from a biased condition to an unbiased condition: a similar result to a damaged drain would occur if the gate was damaged. It is possible to see slight changes in the index-of-refraction from the applied voltages in a bad Intel chip, which would not be as pronounced as a good Intel chip. The good Intel chip should demonstrate large changes from a quiescent state to a biased state where both the gate and drain are applied. Furthermore, it will be difficult to determine the difference between the gate and drain voltages applied individually, although there will be a significant change from the quiescent state. Although a bad chip may have many things wrong with it, the most common is the broken gate wire, yet on occasion a drain is bad. The broken gate is the most detrimental to the chip since it supplies the gate pulses to several drains.

This procedure is non-invasive, apart from the hole in the bottom of the chip package, which allows us to probe the devices from the bottom. This allows us to use our present wavelength of 1064nm, where Silicon is almost transparent. This technology of probing from the bottom will work well for flip-chip technology since the internal device leads will no longer be accessible to conventional probes. The drain and gate voltages are applied through the chip leads. A potentiometer (pot) is used for the variable input voltage range from 0 to 4 volts. The gate has a high impedance (approximately 1MΩ) while the drains are approximately 50 kΩ.

The drain voltages are generated by a nine-volt battery which is voltage-divided to be no more then 4.5 volts just prior to the adjustable pots. The same procedure is applied for the gate, except that it is an input signal, a square wave from 0 to 8 volts. The gate pulse is generated by a pulse generator, which gets its trigger from the DG535 which is triggered from the CID camera.

Prior to installing the Intel chip, a microphotograph of the chip is taken to get an image of the gate and drains (see Fig. 8).



a b

c

Fig. 8: a) The microphotograph of an Intel microprocessor (top view), b) a more detailed view of drains 3, 4 & 5, and c) a more detailed view of drains 2, & 3.

This makes the alignment easier, due to the fact that the components of the FET's are now known and easier to determine when imaged onto the CID camera. This image is taken from the top of the Intel chip instead of through the bottom where we were probing, because we were using a white light microscope to take the image. This just inverts the image. This image also shows the user if there is already a damaged wire going to the drains or the gate. When it is time to install the chip the user must be grounded to the table, and must not touch the chips' pins. A small amount of static could destroy the internal wiring or other sensitive structures of the device.

This method is also able to provide time-resolved diagnostics of semiconductor integrated circuits. The microprocessor's material interacts with the light probe, even though the material is transparent. The interference pattern generated at the image plane of the interferometer samples show local variations in the index-of-refraction, as voltages are applied to the test chip. These changes may be caused by several physical phenomena such as temperature changes due to the flow of electrical currents, charge injection into the depletion region of the transistor(s), and dopants in the material. These devices are comprised of static and dynamic parts. The most interesting of these is the dynamic components associated with normal operation of the device. The two components are easily separable by acquiring an interferogram of the DUT without any voltages applied (a quiescent state), and then a second interferogram when the device enters a known voltage state. The difference between two interferograms determines where and by how much the local index-of-refraction has changed. These steps are repeated for different bias conditions. The DUT is electronically synchronized to the laser probe pulse. This allows the acquisition of a series of images (interferograms) which maps the time evolution of the device. As long as the DUT is stable, an improved signal-to-noise ratio may be obtained by averaging multiple laser probe pulses. It should be possible in practice to test a device at or near its normal clock frequency, since the mode-locked laser typically operates between 50MHz to 100MHz.

The data is acquired by using an Asyst program on a 286 computer. The camera is used to provide an external trigger to the program. This allows only one pulse to be caught per image. The image size is 512 x 512. There are several different drain and gate voltages applied to the DUT. There are ten images taken for each bias condition. A Pentium 133MHz computer is used to analyzed the data with PV Wave, which is a data analysis program. The ten images for each set are summed and analyzed to get the phase from the intensity which is obtained with these four general images: the reference (I_{ref}), DUT (I_{DUT}), interferograms where the DUT is in a quiescent state (I_q), and interferograms where the DUT is biased (I_{bias}). These four values are used to calculate the phase and phase change from the intensity formula in the interferometry theory Eq.2.7, $\Psi_q = \mathrm{Cos}^{-1}((I_q - I_{ref} - I_{DUT})/(2*(I_{ref}*I_{dut})^{1/2})))$ and $\Psi_{bias} = \mathrm{Cos}^{-1}((I_{bias} - I_{ref} - I_{DUT})/(2*(I_{ref}*I_{dut})^{1/2})))$ for the quiescent state and biased state, respectively. The Ψ_q and Ψ_{bias} are used to calculate the interesting physical quantity $\Delta\Psi = (\Psi_{bias} - \Psi_q)$. Although the image acquired is 512 x 512, the area of interest is only about 50 x 200. This area is easily removed with PV Wave. By working with a smaller area, the analysis of the data is not as dependent on the speed of the computer. There is an inherent fringing caused by the interferometer which may be adjusted to minimize this effect. The program used to analyze the data allows the user to block out the area to be examined and magnifies the image. This allows better resolution of the devices and gate. In several cases the hole in the back side of the DUT was not positioned correctly during manufacturing to image the gate and the first drain. Although these devices were not visible, there were significant changes observed in the remaining four devices to determine the state (good or bad) of the DUT. These changes were sufficient to determine if these devices were good or bad, which was the ultimate goal. The index-of-refraction is obtainable with a dependence on the carrier density, which was not disclosed by Intel Corporation. It should also be possible to use spatial variations of the index-of-refraction to image fine structural details of the devices.

V Data and Experimental Results:

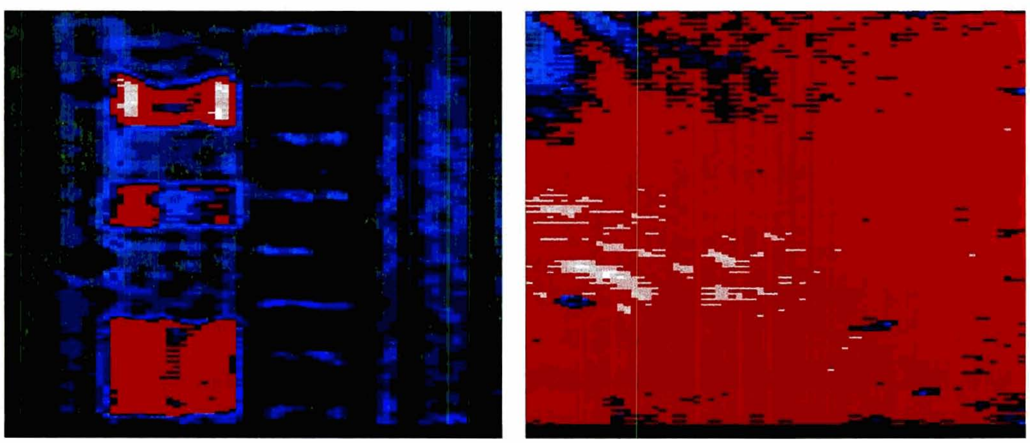
The noninvasive optical diagnostic technique which evaluates the operational state of an integrated circuit is tested. The chip's condition (dead or good) is determined in several ways: putting the output of the IC's gate into a scope, or measuring the drain voltage just prior to the MOSFET using a Digital Volt Meter (DVM). The gate pulse is easy enough to determine its operational state: if it is functioning the gate pulse will display on the scope. The DVM will display the individual drain voltage if it is operational or it will display 0V if the individual drain is damaged. There are several different conditions, which will be examined. They are listed in Table 1:

Test Condition number	Device Under Test (DUT)	Reference
1	Dead chip	Anti-reflective coated Dead chip
2	Good chip	Anti-reflective coated Dead chip
3	Dead chip	Dead chip
4	Good chip	Dead chip
5	Dead chip	Wedge
6	Good chip	Wedge

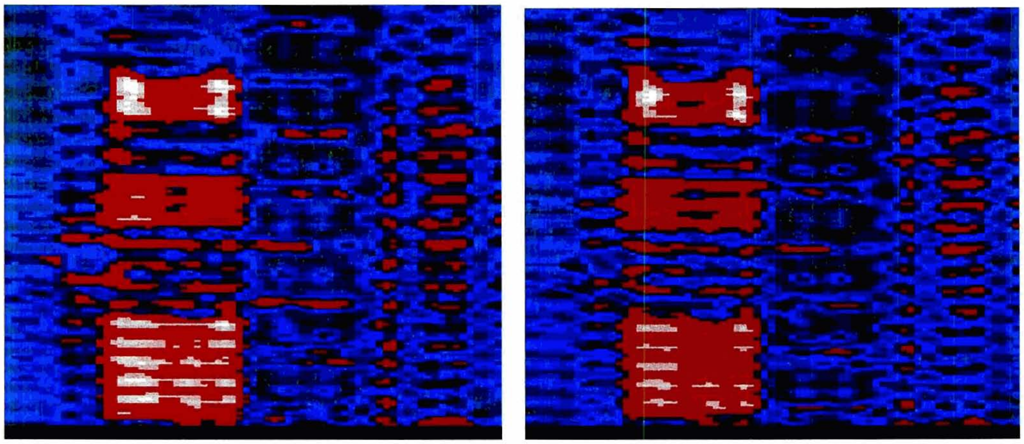
Table 1: Test conditions to be examined.

Each condition is biased with several different voltage states which are: the gate pulse applied both with and without all the drain voltages, all the drain voltages without the gate pulse, the quiescent state, the varying of the drain voltage applied, and the gate pulse with varying drains that are biased (on) and unbiased (off). This set of test points (varying voltage states) is useful in comparing the differences between the DUT which is good and the one which is bad. It is important to note that the gate is shared between all of the drains. The reference arm component is varied to determine which combination of DUT arm and reference arm images generates the best results as well as the easiest results to interpret. The experimental data is in Appendix A.

The data is processed using the program "main121b.pro" (Appendix B), danalysis.pro (Appendix B) and the individual data program, of which an example is "d1117btest.pro" (Appendix B). The individual data program sums the images of the reference beam (I_{ref}), the DUT beam ($I_{DUT,}\ \Psi_o$), the quiescent states intensity and phase (I_q, Ψ_q), and the intensity and phase of the DUT under bias plus the reference beams (I_{bias}, Ψ_{bias}). These image combinations with the intensity formula from Eq 2.6 generate a phase array. The phase from the quiescent state is $\Psi_q=\cos^{-1}((I_q-I_{ref}-I_{DUT})/(\ 2*(I_{ref}*I_{DUT})^{1/2}))$, while the phase from the biased image is $\Psi_{bias}=\cos^{-1}((I_{bias}-I_{ref}-I_{DUT})/(2*(I_{ref}*I_{DUT})^{1/2}))$. The two resultant phases are then subtracted to get a change in phase $\Delta\Psi=(\Psi_{bias} - \Psi_q)$, which is the physical quantity of interest. These phase arrays are then evaluated with a histogram to eliminate any anomalous data, such as pixel values which are low or high in the region of the MOSFET; there may be only a few points in the image which define the MOSFET. Once the points within the histogram region are selected and all the other points are replaced with 1000's, the phase is recalculated and the 1000's are eliminated in order to reduce the error bars. This recalculated phase from a biased DUT is then compared with the phase of the quiescent state. The quiescent state is used as a reference for the biased conditions. This information is then plotted and sent to the file listed above in table 5.2. The program used the histogram of the phase to determine the device and its size. This is accomplished by selecting the histogram minima and adding them into the program, which then blocks off the areas between minima and only display the phase values between those minima. The displaying of the regions between minimas is done so that the user may determine if a device is prominent within that range. The program will then display a range between minima and ask the user to determine if that is a possible section of the MOSFET device. If it is not, the program moves on to the next range and repeats this until all the minima have been evaluated. If no device is found the program uses the original phase to evaluate the phase change. An example of the reference image, DUT image, and DUT plus reference image is shown in Fig. 9.



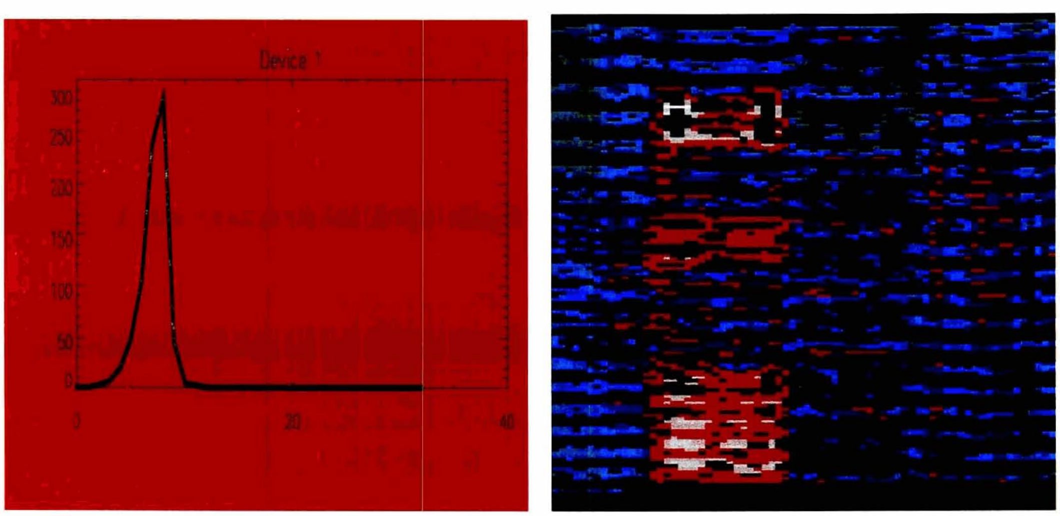
a b



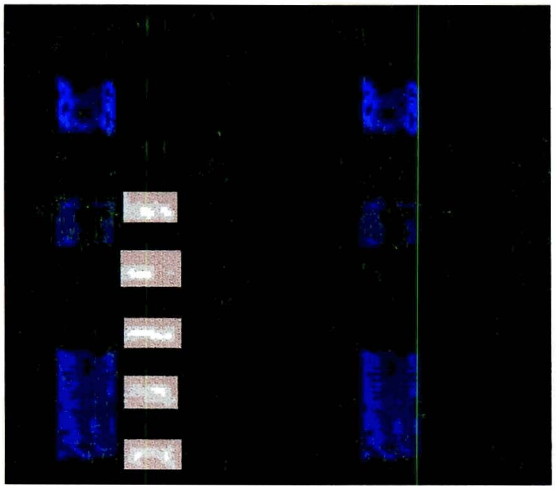
c d

Fig 9: Images to be processed: a) is the DUT image, b) is the reference image, c) DUT plus reference image, and d) A biased image.

The images shown in Fig. 9 are important since they are used to calculate the phase and the phase changes. The image in Fig. 9a is the DUT without the reference; it is easy to locate the devices (MOSFET's) to be tested from this image. The reference image in Fig. 9b is the reflected beam from a glass wedge. The total irradiance is from the interferometer mixing the electric fields from the reference and the device (E_{ref} and E_{DUT}, respectively). Since there is a difference in path lengths a phase angle, Ψ_o, is introduced. Thus the total irradiance is calculated by multiplying the complex conjugates of the electric fields (E_{ref} + $E_{DUT}e^{i\Psi_o}$) times a constant, $I=\varepsilon c E^2/2$. Thus the total irradiance, I, may be written as, $I=I_{ref} + I_{DUT} + 2(I_{ref}*I_{DUT})^{1/2} \cos\Psi_o$. The image in Fig. 9c is the combined image of the DUT and the reference. The image is used to reference the biased condition to the quiescent condition. The final image in Fig. 9d is a biased DUT. It is apparent from examining the images in Fig. 9c and 9d that some intensity variations exist, which generate phase changes. Then the program takes these images and generates the data images in Fig. 10. The phase histogram in Fig. 10a is used to eliminate any anomalous phase angles. This is accomplished by blocking out areas where minima appear. The result of removing anomalous phase angles is shown in Fig. 10b. The differences between Fig. 9d and Fig. 10b are quite apparent, such as the black areas (zeros) around the devices. This method of device selection will eliminate the phase angles from around the MOSFET which are not within the histogram range selected. This elimination is done so that the error from pixels that are out of range and not really a section of the MOSFET will not bias the data. The final Fig. 10c is an image of the DUT and reference beam with the devices to be examined (right) and the user defined device locations demonstrated by the mask over the image (left). The image in Fig. 10c will assist in the determination of whether a device is located in that region of Fig.10b, and if that region should be accepted for the phase angle calculation.



a b



c

Fig 10: Processed data: a) Phase histogram, b) Image of one of the ranges between minima, and c) the image of the DUT with masks for each device (left) and its original unchanged image (right).

The active regions of figure 10 are processed to determine if operating the device produces a phase difference. The information from the file is inputted into Excel to be easily compared between the similar test conditions (for example test condition 5 & 6). This information is demonstrated for selected files, since the information does not change much between test conditions. The test conditions to be examined are the reference and the DUT being both an operational and non-operational Intel microprocessor chip (see Fig. 11, 12, and 13).

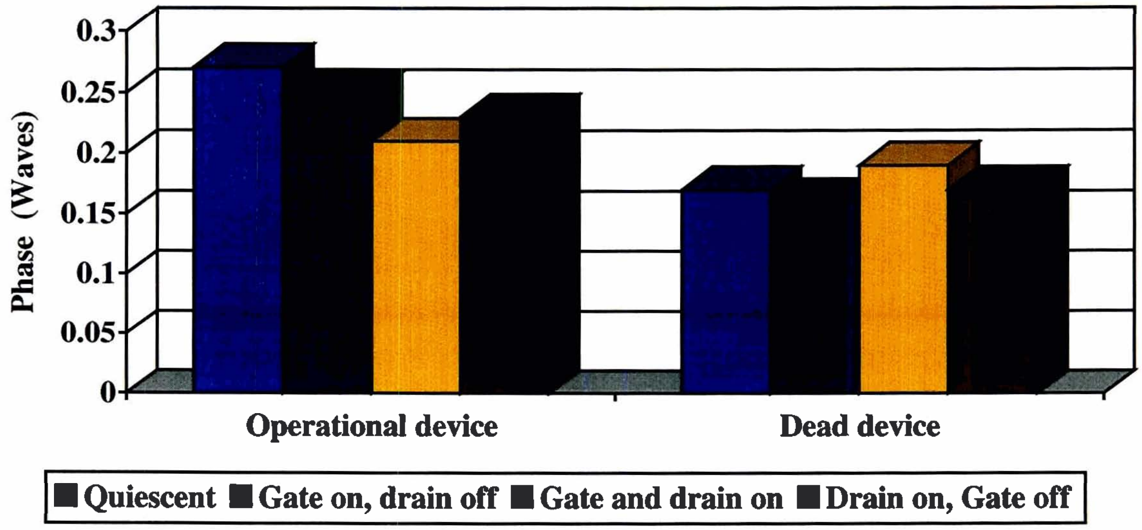


Fig 11: The comparison between an operational DUT and a dead DUT, where the reference is a wedge.

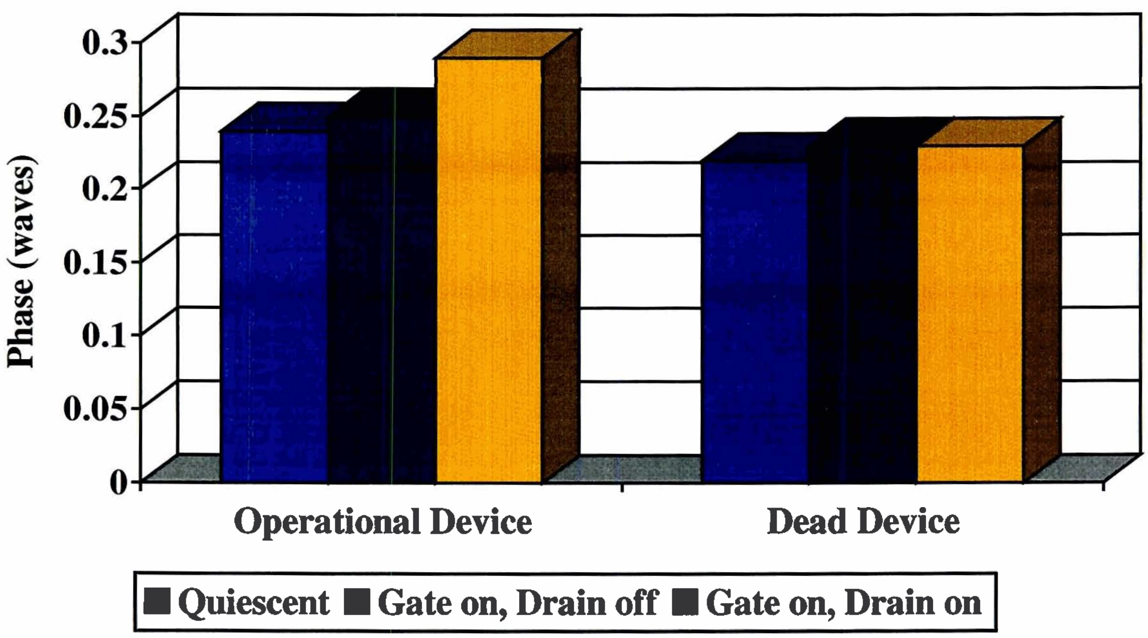


Fig 12: The comparison between an operational and dead DUT, where the

reference is a Dead chip.

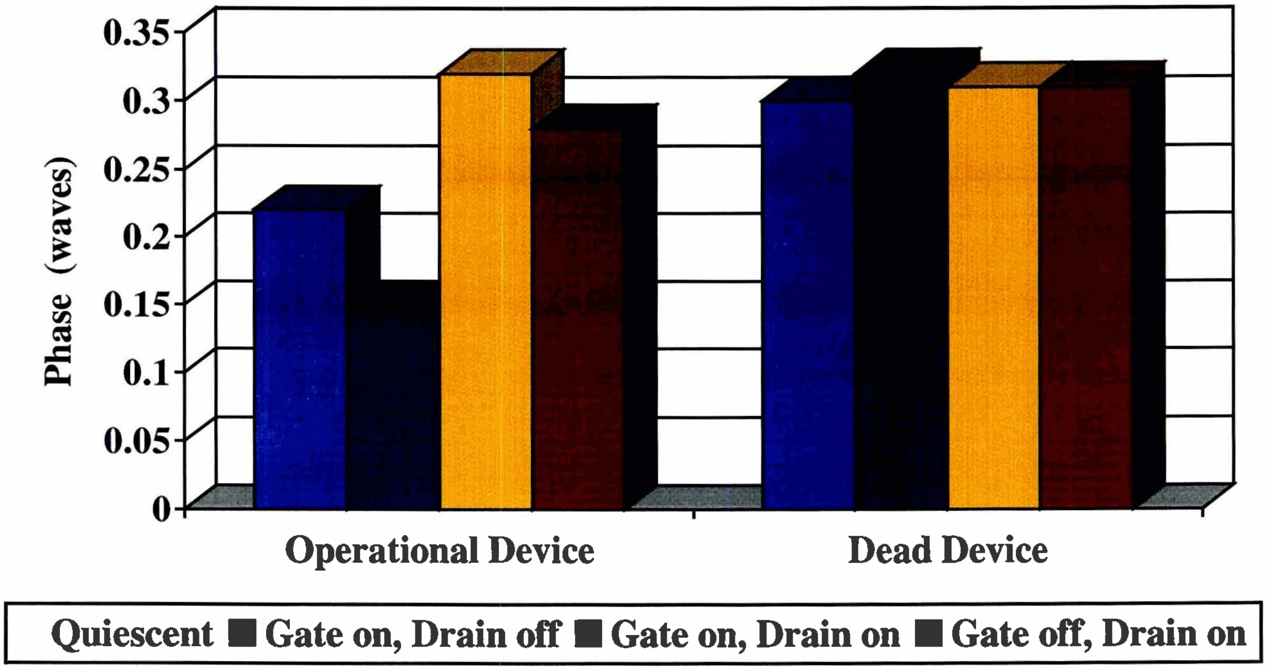


Fig 13: The comparison between an operational and dead DUT, where the reference is an AR coated dead chip

A device's operational state may be determined by examining the phase of each voltage state from quiescent, gate on and drain off, gate off and drain on, and the final voltage combination with both the gate and drain voltages on. If the device is operational, the largest change should be between the quiescent state and the gate plus drain voltages on. If the device is not operational, the changes between all the voltage states should be very small (on the order of less then 0.03 waves of change). These results are demonstrated by fig 11-13. When examining the device for contact sizes, it was discovered that the images were not magnified enough. The magnification would have to be increased to 8-12 times larger than the devices' actual size. The present magnification is 4-6 times larger. The device pads and locations are easily distinguishable. The images didn't appear to have any large changes in the phase around these devices, as if there were a detectable pattern. The phase calculation did however enable the user to determine whether a chip was operational or not (see fig 11-13).

The index-of-refraction is also influenced by the change in temperature. This typical change in the index-of-refraction is one part in 10^{-4} to 10^{-5} for each degree Kelvin. A temperature change may occur due to current flows in the semiconductor, which is I^2R heating. Two situations may develop with device heating: high thermal conductivity will diffuse the energy over a large volume if the device is imbedded in the silicon substrate, or a buried SiO_2 layer can insulate the device from the substrate. The first situation of diffusing the energy is where the temperature rises a few degrees and it is distributed over as much as 500μm (a typical substrate thickness). Therefore, a small change in index-of-refraction is offset by a large path length change. In the second situation, a temperature change of $100^{o}C$ changes the wavelength, which again leads to a few-percent change in the optical path length. The resolution of the system is estimated to be 0.02 waves, which is limited by our CID camera. This estimation was based on several similar measurements. The silicon FET's which were examined easily demonstrated this order of magnitude.

VI Conclusion:

An interferometric method for noninvasive testing of integrated circuits has been described. There were three sets of data taken with good (operational) chips and 3 sets taken with dead (non-operational) chips. It has been determined that the AR-coated reference chip does not generate easily interpreted data, unlike the glass wedge and uncoated AR dead chip. The phase changes we can determine are on average above 0.1 wave. The imaging was not good enough to distinguish the individual FET sub-components. This could work if the images were magnified by another factor of 2-4 times larger. Another way to improve the phase would be to remove the fringing from the interferometer, by using Fast Fourier transforms. An important factor in the data is the regen stability, which limits reproducibility of the images.

There are several ways to improve the system, such as by adjusting the magnification of the chip's devices onto the CID. This increased magnification should allow for a better phase analysis of the actual device, which should also enable the user to locate the device's contacts. The interferometer's inherent fringe pattern needs to be removed either by alignment and/or by Fast Fourier transforms. This will also lessen any large phase fluctuations.

An additional improvement would be to separate the gates for all of the devices. This would increase the acquisition speed since the gate pulse would be shorter and no longer dependent on the capacitance of the large serpentine device.

This separation would allow us to use the 76 MHz oscillator to obtain several pulses during one camera frame, thus reducing the energy and the averaging errors of the present method. At this speed the setup could be used right on the manufacturing line. The systems limiting factor would then be the diffraction limited spot size.

Bibliography:

[1] P. Y. Yu and M. Cardona, *Fundamentals of Semiconductors Physics and Material Properties*, Springer, New York, 1996.
[2] D. A. Fraser, *The Physics of Semiconductor Devices*, Fourth edition, Clarendon Press, Oxford 1986.
[3] J. Singh, *Semiconductor Devices: An Introduction*, McGraw-Hill, 1994
[4] S. M. Sze, *Semiconductor Devices Physics and Technology*, New York, Wiley & Sons, 1985.
[5] K. Bucher, J. Bruns, and H. G. Wagemann, "Absorption Coefficient of Silicon: An Assessment of Measurements and the Simulation of Temperature Variation", J. Applied Physics, vol. 75, Jan 15,1994, 1127-32.
[6] C. F. Klingshirn, *Semiconductor Optics*, New York: Springer-Verlag 1995.
[7] M. L. Cohen and J. R. Chelikowsky, *Electronic Structure and Optical Properties of Semiconductors*, 2nd Edition, New York: Springer-Verlag 1989.
[8] R. T. Phillips, *Coherent Optical Interactions in Semiconductors*, Series B: Physics vol. 330, NATO ASI series, New York: Plenum Press 1994.
[9] R. A. Soref and B. R. Bennett, " Electrooptical Effects in Silicon", IEEE Journal of Quantum Electronics, Vol. QE-23, January 1987, 123-9.
[10] J. D. Kraus, *Electromagnetics*, Third edition, McGraw-Hill, New York, 1984.
[11] H. C. Ohanian, *Physics*, Second edition, W. W. Norton & Co., New York, 1989.
[12] G. Breglio, A. Cutolo, P.Spirito, and L. Zeni, "Interferometric measurement of electron-hole pair recombination lifetime as a function of the injection level," IEEE Electron Device Lett. 14(10), 487-9 (1993).
[13] N. Peyghambarian, S. W. Koch, and A. Mysyrowicz, *Introduction to Semiconductor Optics*, New Jersey, Prentice Hall, 1993
[14] H. Haug and S. W. Koch, *Quantum Theory of the Optical and Electrical Properties of Semiconductors*, New Jersey, World of Science, 1990
[15] B. G. Streetman, *Solid State Electronic Devices*, fourth edition, New York, Prentice Hall, 1995.
[16] W. H. Steel, *Interferometry*, 1st and 2nd editions, New York: Cambridge University Press 1967 & 1987.
[17] P. Hariharan, Selected Papers on Interferometry, SPIE Milestone series, vol. MS28, Washington: SPIE - The International Society for Optical Engineering, 1991, 129-143.
[18] J. Dyson, *Interferometry as a Measuring Tool*, London: Machinery Publishing, 1970.
[19] W. T. Welford, *Aberrations of Optical Systems*, Pennsylvania: Adam Hilger, 1989.
[20] R. Eisberg, R. Resnick, *Quantum Physics of Atoms, Molecules, Solids, Nuclei, and Particles, 2nd Edition*, New York, Wiley, 1985
[21] R. L. Liboff, *Introductory Quantum Mechanics*, New York, Addison-Wesley, 1989

[22] R. Guenther, *Modern Optics*, New York: Wiley, 1990.
[23] E. Hecht, *Optics*, 2nd Edition, Massachusetts: Addison-Wesley, 1987.
[24] L. J. Pinson, *Electro-Optics*, New York: Wiley, 1985.
[25] B. M. Tenbroek, M. S. L. Lee, W. Redman-White, R. J. T. Bunyan, & M. J. Uren, *'Self-Heating Effects in SOI MOSFET's and their Measurement by Small Signal Conductance Techniques,'* IEEE Transactions of Electron Devices, Vol. 43, No. 12, December 1996, 2240-8.
[26] M. B. Kleiner, S. A. Kuhn, and W. Weber, *'Thermal Conductivity of Thin Silicon Dioxide Films in Integrated Circuits,'* Editions Frontierres, 1995, 473-6.
[27] G. Ghibaudo, *'New Method for the Extraction of MOSFET Parameters,'* Electronic Letters, Vol. 9, No. 9, April 1988, 543-5.
[28] W. Redman-White, M. S. L. Lee, B. M. Tenbroek, M. J. Uren and R. J. T. Bunyan, *'Direct Extraction of MOSFET Dynamic Thermal Characteristics From Standard Transistor Structures Using Small Signal Measurements,'* Electronic Letters, Vol 29, No. 13, June 1993, 1180-1.
[29] L. T. Su, J. E. Chung, D. A. Antoniadis, K. E. Goodson, and M. I. Flik, *'Measurement and Modeling of Self-Heating in SOI NMOSFET's,'* IEEE Transactions on Eletron Devices, Vol. 14, No. 1, January 1994, 69-74.
[30] E. Ziegler, *'Applications of an Infrared Laser Scan Microscope in Device Testing,'* Microelectronic Engineering, Vol. 12, 1990, 73-80.
[31] B. Bossmann, P. Baurschmidt, K. Hussey, and E. Black, *'Failure Analysis Techniques with the Confocal Laser Scanning Microscope,'* ISTFA '92: The 18th International Symposium for Testing and Failure Analysis, October 1992, 351-61.
[32] H. Komoda and K. Shimizu, *'Optical Beam Induced Current Techniques for Failure Analysis of Very Large Scale Integrated Circuits Devices,'* Japanese Journal of Applied Physics, Vol. 33, 1994, Pt 1, No. 6A, 3393-401.
[33] L. Peach, *'Near-Field Microscopy Maps Semiconductors,'* Laser Focus World, January 1997, 38-40.
[34] C. Xu & W. Denk,*'2-Photon Optical Beam Integrated Circuits,'* Applied Physics Letters, Vol. 71, No. 18, November, 1997, 2578-80.
[35] E. Cole Jr., *'Electron and Optical Beam Testing of Integrated Circuits using CIVA, LIVA, and LECIVA,'* Microelectronic Engineering, Vol. 31, 1996, 13-24.
[36] J. M. Wiensenfeld, *'Electro-optic Sampling of High-speed Devices and Integrated Circuits,'* IBM Journal of Research and Development, Vol. 34, 1990, 141-61.
[37] W. R. Donaldson, L. Kingsley, M. Weiner, A. Kim, & R. Zeto, *'Electro-Optic Imaging of the Internal Fields in a GaAs photoconductive switch,'* Journal of Applied Physics, Vol. 68, December 1990, 6453-7.
[38] B. H. Kolner & D. M. Bloom, *'Electrooptic Sampling in GaAs Integrated Circuits,'* IEEE Journal of Quantum Electronics, Vol. QE-22, No. 1, January 1986, 79-93.
[39] H. K. Heinrich, D. M. Bloom, & B. R. Hemenway, *'Noninvasive Sheet Charge Density Probe for Integrated Silicon Devices,'* Applied Physics Letters, Vol. 48, April 1986, 1066-8.

[40] D. Jacobs-Perkins, M. Currie, C. C. Wang, C. Williams, W. R. Donaldson, R. Sobolewski, & T. Y. Hsiang, '*Subpicosecond Imaging System Based on Electro-Optic Effect,*' IEEE Journal on Selected Topics in Quantum Electronics, 1998.
[41] R. Falk, J. Adams, S. Ferrier, and D. Capps, '*Dynamic Optical Probing of High-Power Photoconductors,*' IEEE International Pulsed Power Conference, Vol. 1, 1993, 88-91.
[42] G. Breglio, A. Cutolo, P. Spirito, & L. Zeni, '*Interferometric Measurement of Electron-Hole Pair Recombination Lifetime as a Function of Injection Level,*' IEEE Electron Device Letters, Vol. 14, No. 10, October 1993, 487-9.
[43] F. Sanii, F. Giles, R. Schwartz, and J. Gray, '*Contactless Nondestructive Measurement of Bulk and Surface Recombination using Frequency-modulation Free Carrier Absorption,*' Solid-State Electronics, Vol. 35, No. 3, 1992, 311-7.

Appendix A:

The experimental data is included below Appendix A. The following table is a guide to what each page title means.

File Name for each data set	Device Under Test (DUT)	Reference
dat10-09-96.txt	Good chip	Anti-reflective coated Dead chip
dat10-11-96.txt	Good chip	Anti-reflective coated Dead chip
dat10-02-96.txt	Dead chip	Anti-reflective coated Dead chip
dat10-07-96.txt	Dead chip	Anti-reflective coated Dead chip
dat9-24-96.txt	Dead chip	Dead chip
dat9-26-96.txt	Dead chip	Dead chip
dat10-15-96.txt	Good chip	Dead chip
dat11-17-96a.txt	Good chip	Dead chip
dat11-17-96b.dat	Good chip	Wedge
dat11-17-96c.txt	Good chip	Wedge
dat4-19-97.txt	Dead chip	Wedge

```
; WAVE Version 6.10 (Windows 80x86)
; Journal File for emic@PLANO
; Working directory: C:\VNI\wave\bin\bin.i386nt
; Date: Thu Apr 09 13:52:10 1998"

;aa          176          98          238          240
;xy
       24        3       41       10
       25       23       44       32
       25       42       45       50
       25       61       47       69
;      26      122       44      132
;Device :          1
;V(gate)     V(drain)       Phase
;      0.000000      0.000000      0.234863
;       1.50000       1.50000      0.233708
        2.50000       2.50000      0.246332
        3.50000       3.50000      0.253279
;       4.50000       4.50000      0.241550
        6.00000       6.00000      0.244403
;       7.50000       7.50000      0.246674
        8.80000       8.80000      0.239039
;Device :          2
;V(gate)     V(drain)       Phase
       0.000000      0.000000      0.247848
;       1.50000       1.50000      0.228378
;       2.50000       2.50000      0.242582
        3.50000       3.50000      0.195891
        4.50000       4.50000      0.183510
        6.00000       6.00000      0.155893
        7.50000       6.60000      0.221161
        8.80000       6.60000      0.211213
;Device :          3
;V(gate)     V(drain)       Phase
;      0.000000      0.000000      0.237613
        1.50000       1.50000      0.174286
        2.50000       2.50000      0.202790
;       3.50000       3.50000      0.205793
        4.50000       4.50000      0.225928
        6.00000       6.00000      0.191793
        7.50000       7.50000      0.203746
;       8.80000       8.80000      0.200747
;Device :          4
;V(gate)     V(drain)       Phase
       0.000000      0.000000      0.237677
        1.50000       1.50000      0.233720
;       2.50000       2.50000      0.229487
        3.50000       3.50000      0.222489
        4.50000       4.50000      0.237097
        6.00000       6.00000      0.231784
        7.50000       7.50000      0.218094
;       8.80000       8.80000      0.230801
;Device :          5
;V(gate)     V(drain)       Phase
       0.000000      0.000000      0.240618
        1.50000       1.50000      0.230432
        2.50000       2.50000      0.244574
        3.50000       3.50000      0.239007
        4.50000       4.50000      0.231684
        6.00000       6.00000      0.250575
;       7.50000       6.60000      0.229342
;       8.80000       6.60000      0.234271
```

```
; WAVE Version 6.10 (Windows 80x86)
; Journal File for enat@PLANO
; Working directory: C:\VNI\wave\bin\bin.i386nt
; Date: Thu Apr 09 15:38:12 1998"

;aa          177          273          248          424
;xy
;       37        5       55       14
;       38       25       60       36
;       39       45       59       54
;       41       65       56       73
        39       85       54       96
;Device :        1
;V(gate)     V(drain)        Phase
;      0.000000      0.000000      0.274103
        2.00000       2.00000      0.275543
;       2.00000       2.00000      0.237870
;      0.000000      0.000000      0.264486
        2.00000       2.00000      0.271991
;      0.000000      0.000000      0.188177
        2.00000       2.00000      0.199015
        2.00000       2.00000      0.266277
;Device :        2
;V(gate)     V(drain)        Phase
       0.000000      0.000000      0.221421
        2.00000       4.00000      0.179462
        2.00000      0.000000      0.316296
;      0.000000      0.000000      0.275720
;       2.00000      0.000000      0.237636
;      0.000000      0.000000      0.373445
        2.00000      0.000000      0.325339
        2.00000       4.00000      0.239481
;Device :        3
;V(gate)     V(drain)        Phase
       0.000000      0.000000      0.285395
;       2.00000       2.00000      0.288877
        2.00000      0.000000      0.310550
       0.000000      0.000000      0.212650
;       2.00000      0.000000      0.237895
;      0.000000      0.000000      0.303518
        2.00000      0.000000      0.301719
;       2.00000       2.00000      0.240232
;Device :        4
;V(gate)     V(drain)        Phase
       0.000000      0.000000      0.221666
;       2.00000       4.00000      0.260874
;       2.00000       4.00000      0.259288
       0.000000      0.000000      0.222523
;       2.00000      0.000000      0.256603
;      0.000000       4.00000      0.276999
;       2.00000       4.00000      0.209097
        2.00000       4.00000      0.320244
;Device :        5
;V(gate)     V(drain)        Phase
       0.000000      0.000000      0.239487
        2.00000       2.00000      0.235393
        2.00000      0.000000      0.256973
       0.000000      0.000000      0.238642
        2.00000      0.000000      0.247786
;      0.000000      0.000000      0.215313
        2.00000      0.000000      0.224520
;       2.00000       2.00000      0.262913
```

```
; WAVE Version 6.10 (Windows 80x86)
; Journal File for enat@PLANO
; Working directory: C:\VNI\wave\bin\bin.i386nt
; Date: Thu Apr 09 15:56:03 1998"

;aa          157          94          230          254
;xy
       36       11       52       18
       35       31       52       39
       36       51       51       59
       34       71       51       79
       37      132       51      140
;Device :        1
;V(gate)     V(drain)       Phase
;      0.000000      0.000000      0.255193
;       2.00000       2.00000      0.244767
        2.00000       2.00000      0.262689
       0.000000      0.000000      0.290124
        2.00000       2.00000      0.217059
        2.00000       2.00000      0.289364
        2.00000       2.00000      0.264984
       0.000000      0.000000      0.252646
;Device :        2
;V(gate)     V(drain)       Phase
       0.000000      0.000000      0.251065
        2.00000       4.00000      0.261208
;       2.00000      0.000000      0.275931
;      0.000000      0.000000      0.318539
;       2.00000       4.00000      0.284611
;       2.00000      0.000000      0.325314
;       2.00000      0.000000      0.288363
       0.000000      0.000000      0.299165
;Device :        3
;V(gate)     V(drain)       Phase
;      0.000000      0.000000      0.227030
;       2.00000       2.00000      0.252368
;       2.00000      0.000000      0.250610
       0.000000      0.000000      0.273026
        2.00000       2.00000      0.296116
        2.00000      0.000000      0.286784
;       2.00000      0.000000      0.284684
       0.000000      0.000000      0.286802
;Device :        4
;V(gate)     V(drain)       Phase
       0.000000      0.000000      0.299345
;       2.00000       4.00000      0.308793
;       2.00000       4.00000      0.325942
;      0.000000      0.000000      0.333024
;       2.00000       4.00000      0.301021
;       2.00000       4.00000      0.351054
;       2.00000      0.000000      0.325276
       0.000000       4.00000      0.310522
;Device :        5
;V(gate)     V(drain)       Phase
       0.000000      0.000000      0.248860
        2.00000       2.00000      0.249352
;       2.00000      0.000000      0.264434
       0.000000      0.000000      0.261455
        2.00000       2.00000      0.261704
;       2.00000      0.000000      0.291926
;       2.00000      0.000000      0.251869
       0.000000      0.000000      0.260696
```

```
; WAVE Version 6.10 (Windows 80x86)
; Journal File for emic@PLANO
; Working directory: C:\VNI\wave\bin\bin.i386nt
; Date: Thu Apr 09 12:59:10 1998"

;aa          201          85          272          227
;xy
       33        8       47       15
       33       30       48       37
       34       49       48       57
       32       69       48       78
;      32       90       49       98
;Device :        1
;V(gate)     V(drain)       Phase
;      0.000000      0.000000      0.217848
;       2.00000       2.00000      0.231628
;       2.00000       2.00000      0.224359
;      0.000000      0.000000      0.224129
        2.00000       2.00000      0.241271
        2.00000       2.00000      0.213902
        2.00000       2.00000      0.253760
;      0.000000      0.000000      0.210661
;Device :        2
;V(gate)     V(drain)       Phase
       0.000000      0.000000      0.229651
;       2.00000       4.00000      0.240818
;       2.00000      0.000000      0.257412
       0.000000      0.000000      0.263341
;       2.00000       4.00000      0.246377
;       2.00000      0.000000      0.258213
;       2.00000      0.000000      0.297827
;      0.000000      0.000000      0.227672
;Device :        3
;V(gate)     V(drain)       Phase
       0.000000      0.000000      0.226222
;       2.00000       2.00000      0.247856
;       2.00000      0.000000      0.269718
;      0.000000      0.000000      0.260420
;       2.00000       2.00000      0.279286
;       2.00000      0.000000      0.253050
;       2.00000      0.000000      0.297865
;      0.000000      0.000000      0.222114
;Device :        4
;V(gate)     V(drain)       Phase
       0.000000      0.000000      0.206431
;       2.00000       4.00000      0.216640
;       2.00000       4.00000      0.244067
;      0.000000      0.000000      0.234937
        2.00000       4.00000      0.245017
        2.00000       4.00000      0.227040
;       2.00000      0.000000      0.284760
;      0.000000       4.00000      0.217014
;Device :        5
;V(gate)     V(drain)       Phase
;      0.000000      0.000000      0.209741
        2.00000       2.00000      0.229801
;       2.00000      0.000000      0.235062
       0.000000      0.000000      0.224591
        2.00000       2.00000      0.239874
        2.00000      0.000000      0.224909
;       2.00000      0.000000      0.276809
;      0.000000      0.000000      0.234324
```

```
; WAVE Version 6.10 (Windows 80x86)
; Journal File for emic@PLANO
; Working directory: C:\VNI\wave\bin\bin.i386nt
; Date: Thu Apr 09 11:53:19 1998"

;aa          221           55          274          194
;xy
       23        8       36       15
       22       27       36       35
;      24       46       38       55
       23       65       38       74
       26      123       39      131
;Device :        1
;V(gate)     V(drain)       Phase
      0.000000     0.000000     0.274597
       2.00000      2.00000     0.319842
       2.00000      2.00000     0.284650
      0.000000     0.000000     0.291071
       2.00000      2.00000     0.278064
       2.00000      2.00000     0.302174
       2.00000      2.00000     0.312196
      0.000000     0.000000     0.204507
;Device :        2
;V(gate)     V(drain)       Phase
      0.000000     0.000000     0.278149
;      2.00000      4.00000     0.294994
;      2.00000     0.000000     0.277553
;     0.000000     0.000000     0.275337
;      2.00000      4.00000     0.274948
;      2.00000     0.000000     0.257703
;      2.00000     0.000000     0.281839
      0.000000     0.000000     0.288155
;Device :        3
;V(gate)     V(drain)       Phase
      0.000000     0.000000     0.192180
       2.00000      2.00000     0.260153
;      2.00000     0.000000     0.227456
      0.000000     0.000000     0.230165
       2.00000      2.00000     0.215810
;      2.00000     0.000000     0.205880
;      2.00000     0.000000     0.214298
;     0.000000     0.000000     0.201884
;Device :        4
;V(gate)     V(drain)       Phase
;     0.000000     0.000000     0.209379
       2.00000      4.00000     0.240342
       2.00000      4.00000     0.221162
      0.000000     0.000000     0.212048
;      2.00000      4.00000     0.219627
;      2.00000      4.00000     0.210180
;      2.00000     0.000000     0.199114
      0.000000      4.00000     0.168077
;Device :        5
;V(gate)     V(drain)       Phase
      0.000000     0.000000     0.232394
       2.00000      2.00000     0.253639
       2.00000     0.000000     0.232038
      0.000000     0.000000     0.214318
       2.00000      2.00000     0.232781
       2.00000     0.000000     0.230565
       2.00000     0.000000     0.257535
      0.000000     0.000000     0.286428
```

```
; WAVE Version 6.10 (Windows 80x86)
; Journal File for emic@PLANO
; Working directory: C:\VNI\wave\bin\bin.i386nt
; Date: Thu Apr 09 13:13:51 1998"

;aa          143          178          210          317
;xy
;      34       8      46      16
       34      27      49      36
;      34      46      49      55
;      35      65      50      74
       35      85      50      94
;Device :        1
;V(gate)     V(drain)       Phase
      0.000000      0.000000      0.298836
       2.00000       2.00000      0.261623
;     0.000000      0.000000      0.117040
       2.00000       2.00000      0.241409
       2.00000       2.00000      0.227267
;      2.00000       2.00000      0.299638
      0.000000      0.000000      0.263039
;Device :        2
;V(gate)     V(drain)       Phase
      0.000000      0.000000      0.250969
       2.00000       4.00000      0.322637
      0.000000      0.000000      0.181702
;      2.00000       4.00000      0.286197
;      2.00000      0.000000      0.282292
;      2.00000      0.000000      0.262558
;     0.000000      0.000000      0.308466
;Device :        3
;V(gate)     V(drain)       Phase
;     0.000000      0.000000      0.197368
;      2.00000       2.00000      0.309826
;     0.000000      0.000000      0.169692
;      2.00000       2.00000      0.288240
       2.00000      0.000000      0.270588
       2.00000      0.000000      0.190407
;     0.000000      0.000000      0.301847
;Device :        4
;V(gate)     V(drain)       Phase
;     0.000000      0.000000      0.147734
       2.00000       4.00000      0.318741
;     0.000000      0.000000      0.225520
;      2.00000       4.00000      0.287830
       2.00000       4.00000      0.258401
       2.00000      0.000000      0.152067
      0.000000       4.00000      0.283121
;Device :        5
;V(gate)     V(drain)       Phase
      0.000000      0.000000      0.197692
       2.00000       2.00000      0.248743
;     0.000000      0.000000      0.237253
;      2.00000       2.00000      0.238933
       2.00000      0.000000      0.199754
;      2.00000      0.000000      0.201442
      0.000000      0.000000      0.202284
```

```
; WAVE Version 6.10 (Windows 80x86)
; Journal File for enat@PLANO
; Working directory: C:\VNI\wave\bin\bin.i386nt
; Date: Thu Apr 09 17:46:06 1998"

;aa          170            2          224          145
;xy
;       25        3       43        8
;       24       21       41       30
;       24       40       42       50
;       23       62       43       71
;       24       83       41       92
;Device :        1
;V(gate)    V(drain)       Phase
      0.000000     0.000000     0.230698
       2.00000      2.00000     0.270111
       2.00000      2.00000     0.243834
      0.000000     0.000000     0.236909
       2.00000      2.00000     0.210858
      0.000000     0.000000     0.226188
       2.00000      2.00000     0.220582
       2.00000      2.00000     0.210588
;Device :        2
;V(gate)    V(drain)       Phase
;     0.000000     0.000000     0.243050
;      2.00000      4.00000     0.291307
       2.00000     0.000000     0.293006
;     0.000000     0.000000     0.251668
;      2.00000     0.000000     0.246462
      0.000000     0.000000     0.270101
;      2.00000     0.000000     0.250822
       2.00000      4.00000     0.290390
;Device :        3
;V(gate)    V(drain)       Phase
;     0.000000     0.000000     0.276713
;      2.00000      2.00000     0.288171
;      2.00000     0.000000     0.299494
;     0.000000     0.000000     0.267262
;      2.00000     0.000000     0.269341
;     0.000000     0.000000     0.293889
;      2.00000     0.000000     0.279674
;      2.00000      2.00000     0.309718
;Device :        4
;V(gate)    V(drain)       Phase
      0.000000     0.000000     0.257189
       2.00000      4.00000     0.254942
       2.00000      4.00000     0.286652
      0.000000     0.000000     0.259183
;      2.00000     0.000000     0.257785
;     0.000000      4.00000     0.250057
;      2.00000      4.00000     0.261108
       2.00000      4.00000     0.280476
;Device :        5
;V(gate)    V(drain)       Phase
;     0.000000     0.000000     0.264846
       2.00000      2.00000     0.303683
       2.00000     0.000000     0.307728
      0.000000     0.000000     0.280048
       2.00000     0.000000     0.271666
      0.000000     0.000000     0.277461
       2.00000     0.000000     0.282882
       2.00000      2.00000     0.298216
```

```
; WAVE Version 6.10 (Windows 80x86)
; Journal File for emic@PLANO
; Working directory: C:\VNI\wave\bin\bin.i386nt
; Date: Thu Apr 09 10:06:23 1998


;aa         195          98          252          266
;xy
;       25        6       41       18
;       26       30       43       39
;       26       49       43       58
;       26       70       41       79
        26      130       42      138
;Device :       1
;V(gate)     V(drain)        Phase
;      0.000000      0.000000      0.245369
;       2.00000       2.00000      0.281408
;       2.00000       2.00000      0.330207
       0.000000      0.000000      0.241094
        2.00000       2.00000      0.276746
;      0.000000      0.000000      0.266653
        2.00000       2.00000      0.396618
;Device :       2
;V(gate)     V(drain)        Phase
       0.000000      0.000000      0.316569
;       2.00000       4.00000      0.343141
;       2.00000      0.000000      0.347181
;      0.000000      0.000000      0.283742
;       2.00000      0.000000      0.335371
;      0.000000      0.000000      0.260597
;       2.00000      0.000000      0.323697
;Device :       3
;V(gate)     V(drain)        Phase
;      0.000000      0.000000      0.208836
        2.00000       2.00000      0.257013
;       2.00000      0.000000      0.309784
;      0.000000      0.000000      0.214936
;       2.00000      0.000000      0.252374
;      0.000000      0.000000      0.261170
;       2.00000      0.000000      0.415570
;Device :       4
;V(gate)     V(drain)        Phase
       0.000000      0.000000      0.329552
;       2.00000       4.00000      0.368542
        2.00000       4.00000      0.359651
;      0.000000      0.000000      0.309754
;       2.00000      0.000000      0.359986
;      0.000000       4.00000      0.242102
;       2.00000       4.00000      0.299892
;Device :       6
;V(gate)     V(drain)        Phase
       0.000000      0.000000      0.197194
        2.00000       2.00000      0.257037
        2.00000      0.000000      0.308700
;      0.000000      0.000000      0.244148
        2.00000      0.000000      0.264579
;      0.000000      0.000000      0.288518
        2.00000      0.000000      0.389881
```

```
; WAVE Version 6.10 (Windows 80x86)
; Journal File for emic@PLANO
; Working directory: C:\VNI\wave\bin\bin.i386nt
; Date: Wed Apr 08 20:43:28 1998


;aa          234          96          287          239
;xy
       23        4       39       14
       23       24       39       34
;      23       44       40       54
       22       65       40       74
       23      125       40      135
;Device :        1
;V(gate)     V(drain)       Phase
      0.000000      0.000000      0.274416
;     0.000000      0.000000      0.258889
;      2.00000       2.00000      0.195099
       2.00000       2.00000      0.203523
;      2.00000       2.00000      0.249155
      0.000000      0.000000      0.308581
;Device :        2
;V(gate)     V(drain)       Phase
;     0.000000      0.000000      0.276752
;     0.000000      0.000000      0.265726
;      2.00000       4.00000      0.210756
;      2.00000      0.000000      0.226922
;      2.00000      0.000000      0.263923
;     0.000000      0.000000      0.318969
;Device :        3
;V(gate)     V(drain)       Phase
;     0.000000      0.000000      0.285629
      0.000000      0.000000      0.275884
       2.00000       2.00000      0.219798
;      2.00000      0.000000      0.236151
;      2.00000      0.000000      0.272816
;     0.000000      0.000000      0.325255
; % WINDOW: Can't create new metafile.
;Device :        4
;V(gate)     V(drain)       Phase
      0.000000      0.000000      0.292493
      0.000000      0.000000      0.285776
       2.00000       4.00000      0.232573
       2.00000       4.00000      0.250409
       2.00000      0.000000      0.271616
      0.000000       4.00000      0.337373
;Device :        6
;V(gate)     V(drain)       Phase
      0.000000      0.000000      0.286726
      0.000000      0.000000      0.281182
       2.00000       2.00000      0.231358
;      2.00000      0.000000      0.245942
;      2.00000      0.000000      0.262563
;     0.000000      0.000000      0.336931
```

```
; WAVE Version 6.10 (Windows 80x86)
; Journal File for emic@PLANO
; Working directory: C:\VNI\wave\bin\bin.i386nt
; Date: Thu Apr 09 11:25:22 1998


;aa          234           97           288           233
;xy;       23          3        40        12
;       22        24        41        33
;       22        44        40        52
;       23        63        40        74
;       22       124        41       133
;Device :         1
;V(gate)     V(drain)        Phase
;      0.000000      0.000000      0.164567
       0.000000      0.000000      0.144930
;       2.00000       2.00000      0.108498
        2.00000       2.00000      0.169016
        2.00000       2.00000      0.198865
;       2.00000       2.00000      0.287625
        2.00000       2.00000      0.233392
;      0.000000      0.000000      0.224298
;Device :         2
;V(gate)     V(drain)        Phase
;      0.000000      0.000000      0.175959
;      0.000000      0.000000      0.153677
;       2.00000       4.00000      0.130966
        2.00000      0.000000      0.190651
        2.00000       4.00000      0.218409
        2.00000      0.000000      0.294543
        2.00000      0.000000      0.234480
       0.000000      0.000000      0.234832
;Device :         3
;V(gate)     V(drain)        Phase
       0.000000      0.000000      0.172953
       0.000000      0.000000      0.149193
        2.00000       2.00000      0.133587
        2.00000      0.000000      0.186722
        2.00000       2.00000      0.214075
;       2.00000      0.000000      0.299864
;       2.00000      0.000000      0.231140
       0.000000      0.000000      0.229882
;Device :         4
;V(gate)     V(drain)        Phase
       0.000000      0.000000      0.155176
;      0.000000      0.000000      0.143557
;       2.00000       4.00000      0.121832
        2.00000       4.00000      0.180797
;       2.00000       4.00000      0.220676
;       2.00000       4.00000      0.302407
        2.00000      0.000000      0.220795
;      0.000000       4.00000      0.213366
;Device :         6
;V(gate)     V(drain)        Phase
       0.000000      0.000000      0.131917
       0.000000      0.000000     0.0830656
;       2.00000       2.00000     0.0782834
        2.00000      0.000000      0.160801
;       2.00000       2.00000      0.190233
;       2.00000      0.000000      0.257380
;       2.00000      0.000000      0.164844
       0.000000      0.000000      0.171535
```

```
; WAVE Version 6.10 (Windows 80x86)
; Journal File for enat@PLANO
; Working directory: C:\VNI\wave\bin\bin.i386nt
; Date: Thu Apr 09 16:49:40 1998"

;aa          201          22          237          137
;xy
;       10       10       19       16
;       10       24       20       30
;       10       39       19       44
;       10       53       20       59
;       10       67       20       74
;Device :        1
;V(gate)     V(drain)        Phase
        0.000000       0.000000       0.242723
        0.000000       0.000000       0.233645
         3.30000        3.30000       0.227659
        0.000000       0.000000       0.232798
        0.000000       0.000000       0.255605
         3.50000        3.50000       0.245723
        0.000000       0.000000       0.242197
        0.000000       0.000000       0.242723
;Device :        2
;V(gate)     V(drain)        Phase
;       0.000000       0.000000       0.266395
;       0.000000       0.000000       0.244199
;        3.30000        4.00000       0.206758
;       0.000000       0.000000       0.236297
;       0.000000        4.00000       0.213078
;        3.50000        4.00000       0.245606
;       0.000000       0.000000       0.254602
;       0.000000       0.000000       0.266395
;Device :        3
;V(gate)     V(drain)        Phase
;       0.000000       0.000000       0.265407
;       0.000000       0.000000       0.259973
         3.30000        2.00000       0.250188
        0.000000       0.000000       0.257370
;       0.000000       0.000000       0.259506
         3.50000        2.00000       0.263352
        0.000000       0.000000       0.271381
        0.000000       0.000000       0.265407
;Device :        4
;V(gate)     V(drain)        Phase
;       0.000000       0.000000       0.259317
        0.000000       0.000000       0.240313
;        3.30000        3.50000       0.256227
;       0.000000        3.50000       0.250700
        0.000000       0.000000       0.242100
;        3.50000        3.50000       0.236338
        0.000000        3.50000       0.239164
        0.000000       0.000000       0.259317
;Device :        5
;V(gate)     V(drain)        Phase
;       0.000000       0.000000       0.270937
;       0.000000       0.000000       0.241103
         3.30000        2.00000       0.240539
;       0.000000       0.000000       0.248319
;       0.000000        2.00000       0.230775
;        3.50000        2.00000       0.250961
        0.000000       0.000000       0.251604
        0.000000       0.000000       0.270937
```

Appendix B:

The PV Wave data analysis programs are in this section. The first program is main48.pro. This program has all the sub programs called from each data set program, where a data set is a part of a test condition listed in Appendix A. The main program has the sub-program to sum images, locate five devices, perform checks, calculate the phase, calculate histograms, and plot data. The data program sums only the data which is relevant to the test condition and that day

```
;Program to do the essential analysis of all data.
;Program name: main1.pro adapted from intel4.pro

;*******************************************************
;sumwgt sums the images and weights each one with its diode reading.
;*******************************************************

pro sumwgt,series1,pimg,sdimg,aa
print,1,'\\plu\images\intel\september\'
print,2,'\\plu\april\'
print,3,'\\plu\images\intel\april2\'
print,4,'\\plu\april3\'
print,5,'\\plu\images\intel\april4\'
print,6,'d:\'
print,7,'E:\'
read,'number of appropriate directory :',dirnum
case dirnum of
1:dirname='\\plu\images\intel\september\'
2:dirname='\\plu\april\'
3:dirname='\\plu\images\intel\april2\'
4:dirname='\\plu\april3\'
5:dirname='\\plu\images\intel\april4\'
6:dirname='D:\'
7:dirname='E:\'
endcase
; find sum of series with bad image elimination
   fileid=bytarr(10)    ; generate lower case letters from a to j
   fileid(*)=97+indgen(10) ; store letters a ascii byte values
   pimg=fltarr(aa(2)-aa(0)+1,aa(3)-aa(1)+1)
   sdimg=fltarr(aa(2)-aa(0)+1,aa(3)-aa(1)+1)
   all=fltarr(aa(2)-aa(0)+1,aa(3)-aa(1)+1,10)
   pimg(*,*)=0.0
   txt=' '
;  INGORE DIODE INFO - DOES NOT SEEM TO MAKE A BIG DIFFERENCE
;    openr,txtlun,/get_lun,dirname+series1+'.txt'
;    openr,txtlun,/get_lun,dirname+series1+'.txt'
;     readf,txtlun,txt & print,txt
;     readf,txtlun,txt & print,txt
;     readf,txtlun,txt & print,txt
;     readf,txtlun,txt & print,txt
;     readf,txtlun,txt & print,txt
   hello=' '
   wgt=0
   window,0,xsize=aa(2)-aa(0)+1,ysize=aa(3)-aa(1)+1,title=series1,xpos=100,ypos=100
   window,2,xsize=512,ysize=512,title=series1+' elements', xpos=150,ypos=100
   for i=0,9 do begin
    filename=dirname+series1+string(fileid(i))+'.img'
    print,filename
    junk=findfile(filename,count=filex)
    if filex eq 1 then begin
     ;readf,txtlun,txt & print,txt
     ;strdio=strmid(txt,39,9)
     ;diode=strdio+0.0
     ;print,'diode= ',diode
     openr,unit,/get_lun,filename
     a=assoc(unit,bytarr(512,512))
     junk=a(0)
         junk=shift(junk,0,100)
     ; junk=float(junk)/diode
     wset,0      ;sets the window to 0
     print,total(junk(aa(0):aa(2),aa(1):aa(3)))
     tvscl,junk(aa(0):aa(2),aa(1):aa(3))   ;sends the section of the image to the
```

```
                              ;masked window.
     wset,2                        ;sets the window to 2
     tvscl,junk
     read,' say y or n',hello
     if hello eq 'y' then begin
       pimg=pimg+junk(aa(0):aa(2),aa(1):aa(3))
       all(*,*,wgt)=junk(aa(0):aa(2),aa(1):aa(3))
       wgt=wgt+1
     end
     free_lun,unit
    endif
  endfor
  pimg=pimg/wgt
  window,2,xsize=aa(2)-aa(0)+1,ysize=aa(3)-aa(1)+1 $
       ,title='average of : '+series1
  tvscl,pimg
  for i=0,aa(2)-aa(0)do $
     for j=0,aa(3)-aa(1)do $
        sdimg(i,j)=stdev(all(i,j,0:wgt-1))
end

;*******************************************************
;sumimg sums images without bias information.
;*******************************************************

pro sumimg,series1,pimg
; SUMS SERIES
print,1,'\\plu\images\intel\september\'
print,2,'\\plu\april\'
print,3,'\\plu\images\intel\april2\'
print,4,'\\plu\april3\'
print,5,'\\plu\images\intel\april4\'
print,6,'D:\'
print,7,'E:\'
read,'number of appropriate directory :',dirnum
case dirnum of
1:dirname='\\plu\images\intel\september\'
2:dirname='\\plu\april\'
3:dirname='\\plu\images\intel\april2\'
4:dirname='\\plu\april3\'
5:dirname='\\plu\images\intel\april4\'
6:dirname='D:\'
7:dirname='E:\'
endcase
  fileid=bytarr(10)     ; generate lower case letters from a
                        ; to j
  fileid(*)=97+indgen(10) ; store letters a ascii byte values
  pimg=fltarr(512,512)
  pimg(*,*)=0.0
  window,0,xsize=512,ysize=512,title=series1
  for i=0,9 do begin
   filename=dirname+series1+string(fileid(i))+'.img'
   print,filename
   openr,unit,/get_lun,filename
   a=assoc(unit,bytarr(512,512))
   junk=a(0)
   tvscl,junk
   pimg=pimg+junk
   free_lun,unit
  endfor
  tvscl,pimg
end
```

```
;**************************************************************
;active_area allows the user to determine an area which will be evaluated
;and expanded by a factor of 4 later.
;**************************************************************

pro active_area, pimg,actarea
; POINT & CLICK TO FIND A RECTANGLE THAT ENCLOSES ALL DEVICES
window,0,xsize=512,ysize=512
tvscl,pimg
print,'Move cursor to bottom left edge of active area and click'
cursor,x1,y1,/device
cursor,xx,yy,/device
print,'Move cursor to top right edge of active area and click'
cursor,x2,y2,/device
cursor,xx,yy,/device
plots,[x1,x1,x2,x2,x1],[y1,y2,y2,y1,y1],/device,thick=2,color=254
actarea=[x1,y1,x2,y2]
print,'active area bounded by : ', actarea
end

;**************************************************************
;find_5_dev allows the user to find 5 devices from the active_area masked of
;in active_area.
;**************************************************************

pro find_5_dev,pimg,ixy,x,y
; FINDS 5 RECTANGLES WITHIN THE ACTIVE AREA WHERE DEVICES ARE LOCATED
ok='y'
ixy=fltarr(4,5)
;x=fltarr(11)
;y=fltarr(11)
x1=0
x2=0
y1=0
y2=0
;cnt=0
xy=intarr(4,5)
spimg=size(pimg)
print,'spimg',spimg          ;diagnostic
window,21,xsize=4*spimg(1),ysize=spimg(2)*4,title='Find devices' $
    ,xpos=720
; RESIZE THE IMAGE BY 4X TO MAKE IT EASIER TO SEE WHERE THE DEVICES ARE
bigimg=rebin(pimg,4*spimg(1),4*spimg(2),/sample)
tvscl,bigimg
for i=0,4 do begin
if total(ixy(*,i)) eq 0  then begin
 print,'Move cursor to bottom left edge of ',i+1,' device and click'
 cursor,x1,y1,/device
  xy(0,i)=x1
    ;x(cnt)=x1
  xy(1,i)=y1
    ;y(cnt)=y1
cursor,x,y,/device
    ;cnt=cnt+1
 print,'Move cursor to top right edge of',i+1,' device and click'
 cursor,x2,y2,/device
  xy(2,i)=x2
    ;x(cnt)=x2
  xy(3,i)=y2
    ;y(cnt)=y2
    ;cnt=cnt+1
```

```
 cursor,x,y,/device
endif else xy(*,i)=4*ixy(*,i)
Print, 'Please record these x and y positions.'
Print, x1,x2,y1,y2          ;diagnostic
;dev_histogram, x1,x2,y1,y2,pimg, 4*spimg(1),spimg(2)*4
window,21,xsize=4*spimg(1),ysize=spimg(2)*4,title='Find devices'$
   ,xpos=720
tvscl,bigimg
plots,[xy(0,i),xy(0,i),xy(2,i),xy(2,i),xy(0,i)],$
     [xy(1,i),xy(3,i),xy(3,i),xy(1,i),xy(1,i)],thick=2,color=240,/device
;print,'x :',fix(.5 +x/4.0),'  y :',fix(.5+y/4.0)
read,'Is this ok? ( y or n )',ok
   if ok eq 'n' then repeat begin
     print,'current xy: ',xy(*,i)
     read,'new x1,y1,x2,y2',x1,y1,x2,y2
       xy(0,i)=x1
       xy(1,i)=y1
       xy(2,i)=x2
       xy(3,i)=y2
     tvscl,bigimg
     plots,[xy(0,i),xy(0,i),xy(2,i),xy(2,i),xy(0,i)],$
      [xy(1,i),xy(3,i),xy(3,i),xy(1,i),xy(1,i)],thick=2, $
      color=240,/device
     read,'Is this ok? ( y or n )',ok
   endrep until ok eq 'y'
 endfor
;cnt=0
;print,'count',cnt                  ;diagnostic
;for j=0,4 do begin
        x(cnt)= xy(0,j)
        y(cnt)= xy(1,j)
        cnt=cnt+1
        print, 'count',cnt                                ;diagnostic
        x(cnt)= xy(2,j)
        y(cnt)= xy(3,j)
        cnt=cnt+1
        print,'count',cnt ,' j',j                         ;diagnostic
;endfor              ; ends the for loop to cycle through all the devices
tvscl,bigimg
for i=0,4 do plots,[xy(0,i),xy(0,i),xy(2,i),xy(2,i),xy(0,i)],$
     [xy(1,i),xy(3,i),xy(3,i),xy(1,i),xy(1,i)],thick=2,color=240,/device
ixy=fix(0.5+xy/4.0)
end

;*******************************************************************
;check_mask overlays the defined area over the image to check if it is positioned
;correctly
;*******************************************************************

pro check_mask,aa,dmask,img
; DISPLAY MASK SUPERIMPOSED  ON CORRECT SCALE IMAGE TO CHECK
window,20,xsize=20+2*(aa(2)-aa(0)),ysize=aa(3)-aa(1)+1,title='check mask',xpos=200,ypos=100
big=fltarr(20+2*(aa(2)-aa(0)),aa(3)-aa(1)+1)
junk=img
junk(dmask)=junk(dmask)+2*max(img)
big(0:aa(2)-aa(0),*)=junk
big(aa(2)-aa(0)+19:*,*)=img
tvscl,big
end

;*******************************************************************
;pick_dev uses the defined devices to calculate the phase
```

```
;*******************************************************************

pro pick_dev,phas,dcount,dthe,num
; PICK OUT THE DEVICE USING THE MASK AND CALCULATE THE PHASE
 dthe=fltarr(5,num)
 startindx=0
 endindx=dcount(0)-1
 for i=0,4 do begin
   for j=0,num-1 do begin
     phasj=phas(j,startindx:endindx)
     good=where(phasj lt 1999.0,ngood)
     if (ngood gt 0 ) then dthe(i,j)=avg(phasj(good))else print,'Bad area:',i,j
   endfor
   startindx=startindx+dcount(i)
   endindx=endindx+dcount(i+1 < 4)
 endfor
end

;*******************************************************************
;plot_dev plots the calculated phase of the devices and plots them
;*******************************************************************

pro plot_dev,phas,dcount,avph,std,num
;pro plot_dev,phas,dcount,avph,std,num,ref,dut,xy,i0,i1,i2,i3,i4,i5
     ;the lower plot_dev is used if his_spread is used.
; PLOTS THE DEVICE USING THE MASK AND CALCULATED PHASE
 avph=fltarr(5,num)
 std=fltarr(5,num)
 ;his1=fltarr(num,33)
 ;sphas=size(phas)
 ;angle1=fltarr(num,sphas(2))
 ;his_spread,phas,ref,dut,xy,angle1, i0, i1, i2, i3, i4,i5
 ;phas=angle1
 count=0
 pgood=0
 startindx=0
 endindx=dcount(0)-1
 ;print,'pgood',pgood          ;diagnostic
 ;print,'startindex',startindx,'endindex',endindx  ;diagnostic
 good=where(phas lt 999,ngood)
 if good GT 0 then maxph=max(phas(good),min=minph) else begin
         maxph=1
         minph=-1
    endelse
 window,0,xsize=350,ysize=350,xpos=10,ypos=30,title='device 1'
 window,1,xsize=350,ysize=350,xpos=361,ypos=30,title='device 2'
 window,2,xsize=350,ysize=350,xpos=712,ypos=30,title='device 3'
 window,3,xsize=350,ysize=350,xpos=10,ypos=381,title='device 4'
 window,4,xsize=350,ysize=350,xpos=361,ypos=381,title='device 5 or 6'

 for i=0,4 do begin
   wset,i
   for j=0,num-1 do begin
     phg=phas(j,startindx:endindx )
     pgood=where(phg lt 999,count)
     if count gt 1 then begin
       ;print,'pgood',pgood          ;diagnostic
       avph(i,j)=avg( phg(pgood))/(2*!pi)  ; convert to waves
       if stdev(phg(pgood))/(2*!pi) gt 1 then std(i,j)=stdev(phg(pgood))/(2*!pi) $
         else begin
              print,'bad stdev'
             std(i,j)= 1
```

```
        end ; end of if stdev gt 1
     end ;for if statement for count gt 0
      if (j eq 0 ) then begin
       plot,phas(j,startindx:endindx), yrange=[minph,maxph], thick=3, charthick=2, background=255
       xyouts,.01,.1,/normal,string(0)+'  '+string(avph(i,j))
      end else begin
        oplot,phas(j,startindx:endindx),color=(j+2)*20 ,thick=3, charthick=2, background =255
        xyouts,.01,.1+.1*j,/normal,string(j)+'  '+string(avph(i,j)),color=(j+2)*20
      end   ;for the if else

   endfor  ;for the j incriment
   startindx=startindx+dcount(i)
   endindx=endindx+dcount(i+1 < 4)
  endfor    ; for the i incriment

print,'    image  D1       D2       D3       D4       D6'
for k=0,num-1 do print,k,avph(*,k)
end

;*****************************************************************
;all_dev generates a journal of all the data except for the graphs
;*****************************************************************

pro all_dev,avph,volt,xy,aa,jfile,std,num
journal,'c:\users\elise\'+jfile+'.txt'
print,'aa',aa
print,'xy'
print,xy
for i=1,5 do volt_phase,avph,volt,i,jfile,std,num
window,0
ymin=min(avph,max=ymax)
plot,volt(0,*),avph(0,*),yrange=[ymin,ymax], $
      thick=3, charthick=2, background=255, $
      xtitle='drain volts',title=jfile,ytitle='phase'
xyouts,.9,.1,/normal,string(1,format='(i2)')
for i=1,4 do begin
  oplot,volt(i,*),avph(i,*),color=i*40, thick=3, charthick=2,background=255
  xyouts,.9,.2+i*.1,/normal,color=i*40,string(i+1,format='(i2)')
 endfor
journal
end

;*****************************************************************
;all_dev_bin generates a journal of all the data except for the graphs
;*****************************************************************

pro all_dev_bin,avph,volt,xy,aa,jfile,std,num,nummin
journal,'c:\users\elise\'+jfile+'.txt'
print,'aa',aa
print,'xy'
print,xy
for i=1,5 do begin
     for k=0,nummin do volt_phase_bin,avph,volt,i,jfile,std,num,k
endfor
window,0
ymin=min(avph,max=ymax)
plot,volt(0,*),avph(0,*,*),yrange=[ymin,ymax], $
      thick=3, charthick=2, background=255, $
      xtitle='drain volts',title=jfile,ytitle='phase'
xyouts,.9,.1,/normal,string(1,format='(i2)')
for i=1,4 do begin
  oplot,volt(i,*),avph(i,*,*),color=i*40, thick=3, charthick=2,background=255
```

```
 xyouts,.9,.2+i*.1,/normal,color=i*40,string(i+1,format='(i2)')
 endfor
journal
end


;*****************************************************************
;dev_mask creates a device extraction mask
;*****************************************************************


pro dev_mask,aa,xy,dmask,dcount
; make device mask
print, 'The program dev_mask running now.'
dcount=intarr(5)
; CREATE A RAMPED X & Y IMAGE
yy=transpose(indgen(aa(3)-aa(1)+1,aa(2)-aa(0)+1) mod (aa(3)-aa(1)+1))
xx=transpose(findgen(aa(3)-aa(1)+1,aa(2)-aa(0)+1))/(aa(3)-aa(1)+1)
i=0
; FIND LOCATION OF 1ST DEVICE
dmask=where((xx ge xy(0,i)) and (xx le xy(2,i)) and (yy ge xy(1,i)) and $
        (yy le xy(3,i)),cnt)
dcount(i)=cnt
for i=1,4 do begin
; CONCATENATE 4 MORE DEVICES
 dmask=[ dmask ,where((xx ge xy(0,i)) and (xx le xy(2,i)) and $
                (yy ge xy(1,i)) and (yy le xy(3,i)),cnt)]
 dcount(i)=cnt
endfor
end

;********************************************************************
;This section will be used to sort out each of the five devices from the whole image
; and save it to a file to be called in later without all the future processing.
;********************************************************************

pro dev_save,xy,image,dev1,dev2,dev3,dev4,dev5,num
sdev1x=fltarr(5)
sdev1y=fltarr(5)
for i=0,4 do begin
    sdev1x(i)= (xy(2,i)-xy(0,i)+1)
    sdev1y(i)= (xy(3,i)-xy(1,i)+1)
endfor      ;ends the for loop to determine how many pixels are in each device
dev1=fltarr(10,sdev1x(0),sdev1y(0))
dev2=fltarr(10,sdev1x(1),sdev1y(1))
dev3=fltarr(10,sdev1x(2),sdev1y(2))
dev4=fltarr(10,sdev1x(3),sdev1y(3))
dev5=fltarr(10,sdev1x(4),sdev1y(4))
si0=size(image)
window,0,xsize=si0(1), ysize=si0(2)
tvscl,image
cnt=0
numx=0
numy=0
for i=0,4 do begin
    for j= xy(0,i) , xy(2,i) do begin
        for k= xy(1,i),xy(3,i) do begin
            if i eq 0 then dev1(num,(j- xy(0,i)),(k- xy(1,i))) = image(j,k)
            if i eq 1 then dev2(num,(j- xy(0,i)),(k- xy(1,i))) = image(j,k)
            if i eq 2 then dev3(num,(j- xy(0,i)),(k- xy(1,i))) = image(j,k)
            if i eq 3 then dev4(num,(j- xy(0,i)),(k- xy(1,i))) = image(j,k)
            if i eq 4 then dev5(num,(j- xy(0,i)),(k- xy(1,i))) = image(j,k)
```

```
        endfor              ;ends the k loop to do the y values of the image and sort into devices
    endfor                  ;ends the j loop to do the x values of the image and sort into the devices
endfor                      ;ends the I loop to do each of the 5 devices
end       ;ends the procedure dev_save



;*****************************************************************
;function phase calculates the phase of each device. (all at once)
;*****************************************************************

function phase,I2,Iref,idut
; CALCULATE PHASE ANGLE
test=idut*iref
nz=where(test ne 0,cnt)
ac=i2
ac(*)=-1.0*!pi
if cnt gt 0 then ac(nz)=(I2(nz)-Iref(nz)-Idut(nz))/(2*sqrt(Idut(nz)*Iref(nz)))
bad=where((ac gt 1) or (ac lt (-1)),nbad)
ac = ac < 1
ac = ac > (-1)
angle=acos(ac)
if (nbad gt 0) then angle(bad)=1000.  ; flag bad points
return,angle
end

;***************************************************************
;Procedure to find the x and y positions  of each device
;***************************************************************

pro xy_positions, x,y
y=fltarr(11)
x=fltarr(11)
print, 'Please input the 10 X positions of the 5 devices'
for j=1,10 do begin
    read, temp
    x(j)=NINT(temp/4)
endfor
Print, 'Please input the 10 Y positions of the 5 devices'
for j=1,10 do begin
    read, temp
    y(j)=NINT(temp/4)
endfor      ;ends j loop
end ; ends xy_position procedure

;******************************************************************************
;This will process the phases and return a final phase values for both the plus and minus delta
;phases
;******************************************************************************

pro all_parts, x,y, i1, php0,phm0,phas
    si1=size(i1)
    n=si1(1)
    m=si1(2)
    sphp0=size(php0)
    a=sphp0(1)
    b=sphp0(2)
    print, 'php0 dimension 1',a,'php0 dimension 2',b
    temp=fltarr(n,m)
    temp(*,*)=0
```

```
temp=i1
tempr=fltarr(n,m)
tempr(*,*)=0
sum=intarr(n,m)
phplus=fltarr(n,m)
phmin=fltarr(n,m)
mask=fltarr(n,m)
mask(*,*)=0
;his1=fltarr(256)
answer=''
value=0
number=0
count=0
;THIS SECTION IS PRODUCING A MASK FOR THE DEVICES.  THE MASK WILL
;CONTAIN ONLY VALUES OF 1.
for i=1,9,2 do begin
    cnt=0
    mask(x(i):x(i+1),y(i):y(i+1))=1
    roi_indices=where ((mask eq 1),cnt)
    ;his1(*)=(i1(roi_indices))
    if cnt gt 0 then sum(roi_indices)=i1(roi_indices)+sum(*)
endfor      ;ends the for loop to fill in a mask of the device areas.
Window,31,xsize=n,ysize=m,title='Mask of the devices'
tvscl,sum

;THIS SECTION DOES THE HISTOGRAM ON THE DELTA PHASES (BOTH PLUS AND MINUS
; THE UNBIASED IMAGES)
byte_phase_p=byte(php0*256/!pi)  ;converts php0 to a byte value between 0 and 255
byte_phase_m=byte(phm0*256/!pi); converts phm0 to a byte value between 0 and 255
phase_his_p=bytarr(256)
phase_his_m=bytarr(256)
for i=0,a-1 do begin
    for j=0,b-1 do begin
        phase_his_p(byte_phase_p(i,j))= phase_his_p(byte_phase_p(i,j))+1
        phase_his_m(byte_phase_m(i,j))= phase_his_m(byte_phase_m(i,j))+1
            ;this section is filling the byte array where the byte_phase_* has the same value through out the array
            ;Thus this is the histogram of the added and subtracted unbiased image from the biased images.
    endfor          ;ends the j for loop to get the y positions of the image
endfor              ;ends the I for loop to get the x positions of the image
window,    17,xsize=1024, ysize=256, title = 'Histogram of adjusted phases.'
plot, phase_his_p, xrange=[0,256], xtitle='Phase value', ytitle='# of pixels in each histogram value', $
linestyle=1,color=50, thick=3, title='Delta Phase Histograms'
    oplot, phase_his_m, linestyle=2, color=125, thick=3
print,'Are there any ranges which may be a device (for the added phase values)? (y or n)'
read,answer
if answer eq 'y' then begin
    print,'How many ranges are there?'
    read,value
    his_r=fltarr(2*value+2)
    his_r(0)=0
    his_r(2*value+1)=255
    maskdev=bytarr(value)
    print,'Please input the edges of each range.'
    for i=1,2*value do begin
        read, number
        his_r(i)=number                 ; gets the ranges for the historgram
    endfor      ;ends the for loop which gets the ranges of each possible device
    window,10,xsize=si1(1)*4,ysize=si1(2)*4,title='Possible devices'
    for j=0,2*value do begin
            print,'Histogram range',j
            bdppl=where ((byte_phase_p(*,*) GT his_r(j)) and (byte_phase_p(*,*) LT his_r(j+1)),count)
            if count GT 0 then begin
```

```
                    maskdev(bdppl)=1
                    smaskdev=size(maskdev)
                    rangemult=rebin(maskdev,smaskdev(1)*4,smaskdev(2)*4,/sample)
                    tvscl,rangemult
                    answ=' '
                    Print,'Does this appear to match devices?'
                    read,answ
                    if answ eq 'y' then begin
                        for k=0,a-1 do begin
                            for l=0,b-1 do begin
                                if ((byte_phase_p(k,l) GT his_r(j)) and (byte_phase_p(k,l) LT his_r(j+1))) then  begin
                                    phplus(k,l)=php0(k,l)
                                endif else   phplus(k,l)=1000
                            endfor              ;ends the l for loop to get the y positions of the image
                            endfor                  ;ends the k for loop to get the x positions of the image
                        endif         else print, 'There is no devices.'      ;ends the if-then statement for if a device match is found
                    endif     else count=0         ;ends the if-then statement for if count GT 0
            endfor                       ;ends the for j loop for the ranges between 0 and 2*value
        endif                            ;ends the if-then statement for answer eq 'y' for ranges on the histogram for the added phases
    print,'Are there any ranges from the Histogram which may be a device (for the subtracted phase values)? (y or n)'
    read,answer
    if answer eq 'y' then begin
        print,'How many ranges are there?'
        read,value
        his_r=fltarr(2*value+2)
        his_r(0)=0
        his_r(2*value+1)=255
        maskdev=bytarr(value)
        print,'Please input the edges of each range.'
        for i=1,2*value  do begin
            read, number
            his_r(i)=number
        endfor       ;ends the for loop which gets the ranges of each possible device
        window,10,xsize=si1(1)*4,ysize=si1(2)*4,title='Possible devices'
        count=0
        for j=0,2*value   do begin
            print, 'The subtracted unbiased images.'
            if ((phase_his_m(i) GT his_r(j)) and (phase_his_m(I) LT his_r(j+1))) then begin
                bdpm=where (((phase_his_m(i) GT his_r(j)) and (phase_his_m(I) LT his_r(j+1))),count)
                if count GT 0 then begin
                    maskdev(bdpm)=1
                    smaskdev=size(maskdev)
                    rangemult=rebin(maskdev,smaskdev(1)*4,smaskdev(2)*4,/sample)
                    tvscl, rangemult
                    answ=' '
                    Print,'Does this appear to match devices?'
                    read,answ
                    if answ eq 'y' then begin
                        for k=0,a-1 do begin
                        for l=0,b-1 do begin
                            if ((byte_phase_m(k,l) GT his_r(j)) and (byte_phase_m(k,l) LT his_r(j+1))) then begin
                                phmin(k,l)=phm0(k,l)
                            endif else phmin(k,l)=1000
                        endfor             ;ends the l for loop to get the y positions of the image
                    endfor                 ;ends the k for loop to get the x positions of the image
                endif    else print, 'There is not a device.'  ;ends the if-then statement for if a device match is found
                endif         else count = 0   ;ends the if-then statement for if count GT 0
            endif                  ;end the if-then for the histogram condition
            endfor                 ;ends the j for loop
        endif       ;ends the if-then statement for answer eq 'y' for ranges on the histogram of the minus condition

        window,11,xsize=n,ysize=m, title='Added phases'
```

```
    image1=phplus
    image1(*,*)=0
    pp=(where (phplus NE 1000))
    image1(pp)=1
    tvscl, image1
    window,12,xsize=n,ysize=m, xpos=100, ypos=100,title='Subtracted phases'
    image2=phmin
    image2(*,*)=0
    pm=(where (phmin NE 1000))
    image2(pm)=1
    tvscl, image2
    Print,'Do these appear to be a device? (y or n)'
    reply=' '
    read,reply
    if reply EQ 'y' then begin
        php0=phplus
        phm0=phmin
    endif                           ;ends the if statement which changes php0 and phm0 to there final values.
end                                 ;ends the procedure all_parts

pro stuff, num,i1,phas,x,y,volt,jfile
phm0=phas
php0=phas
num1=num-1
for i=0,num1 do begin & $
  bad=where((phas(i,*) gt 999.0) or (phas(0,*) gt 999.0),nbad) & $
  phm0(i,*)=phas(i,*)-phas(0,*) & $   ; subtract 0 phase case
  php0(i,*)=phas(i,*)+phas(0,*) & $   ; add 0 phase case
  if (nbad gt 0) then begin & phm0(i,bad)=2000.0 & php0(i,*)=2000.0 & $
   end & $
endfor  ;ends for i loop to add and subtract and add the unbiased phase
    ;from a biased phase
all_parts, x,y, i1, php0,phm0,phas

plot_dev,php0,dcount,avph,std,num

device,/cursor_original
Print,'This section is the phase of a biased image set plus the unbiased image set.'
all_dev,avph,volt,xy,aa,jfile,std,num
window,0
v4=fltarr(num)
d40=fltarr(num)
st4=fltarr(num)
v4(*)=volt(3,*)
d40(*)=avph(3,*)-avph(3,0)
st4(*)=std(3,*)

plot,v4,d40,xtitle='Drain Voltage',$
ytitle='Phase angle (wave)',charsize=1.3,$
    thick=2,title='Response of 20/20 FET ',yrange=[0,.2]
oploterr,v4,d40,st4,num1
window,2
;v1=fltarr(num)
;d10=fltarr(num)
;st1=fltarr(num)
;v1(*)=volt(0,*)
;d10(*)=avph(0,*)-avph(0,0)
;st1(*)=std(0,*)

;plot,v1,d10,xtitle='Drain Voltage',$
;    ytitle='Phase angle (wave)',charsize=1.3,$
 ;   thick=2,title='Response of 1344/0.7 FET ',yrange=[0,.2]
```

```
;oploterr,v1,d10,st1,num1
;!psym=0

wait,60

plot_dev,phm0,dcount,avph,std,num
Print,'This section is the phase of a biased image set minus the unbiased image set.'
all_dev,avph,volt,xy,aa,jfile,std,num
v4=fltarr(num)
d40=fltarr(num)
st4=fltarr(num)
v4(*)=volt(3,*)
d40(*)=avph(3,*)-avph(3,0)
st4(*)=std(3,*)
plot,v4,d40,xtitle='Drain Voltage',$
ytitle='Phase angle (wave)',charsize=1.3,$
thick=2,title='Response of 20/20 FET ',yrange=[0,.2]
oploterr,v4,d40,st4,num1
!psym=0
end                              ;ends procedure stuff
```

```
;Program to analyze the data from 11-17-96 set b
;Program Name: d1117b.pro
;reference is a wedge. DUT is a Good chip
;This program uses the procedures: sumimg, sumwgt, active_area, find_5_dev,
            check_mask, pick_dev, dev_mask, dev_save, and xy_positions.


device,/cursor_crosshair
loadct,13
already='no'
;already='yes'
if ( already eq 'no' ) then begin & $
sumimg,'tb1117',tb1117            & $     ; read DUT to find active area
tb1117b=shift(tb1117,0,100)         & $; shifts the image up 100 pixels
active_area,tb1117b,aa           & $      ; find active area = aa
print,' active area : ',aa     & $
sumwgt,'tb1117',idut,sdidut,aa     & $      ; Idut
sumwgt,'rb1117',ref,sdref,aa     & $      ; Iref
sumwgt,'rtb17',i0,sdi0,aa  & $      ; I2(v=0)
sumwgt,'gndb17',i1,sdi1,aa     & $     ;vd1,2,3,4,6=0 v vg=2V
sumwgt,'gdbd17',i2,sdi2,aa     & $     ;vd1,3,6=2v vg=2v vd2,4=4v
sumwgt,'gdbc17',i3,sddrc,aa     & $     ;vd1=2v vg=2v vd4=4v
;sumwgt,'ngdbd',i4,sddrd,aa     & $     ;vd1, 3, 6=2v vg=0v vd2,4=4v
;sumwgt,'ngdbc',i5,sddre,aa     & $     ;vd1, =2v vg=0v vd4=4v
sumwgt,'gdba17',i4,sddrf,aa     & $     ;vd1=2v vg=2v
sumwgt,'gdbb17',i5,sddrf,aa     & $     ;vd4=4v vg=2v
endif
num=8
volt=fltarr(5,num)
volt(*,0)=0 +.1 ; gate voltage
volt(*,1)=0 ; voltage
volt(*,2)=[2,4,2,4,2] ; voltage
volt(*,3)=[2,0,0,4,0] ; voltage
;volt(*,4)=[2,4,2,4,2] ; voltage
;volt(*,5)=[2,0,0,4,0] ; voltage
volt(*,4)=[2,0,0,0,0] ;
volt(*,5)=[0,0,0,4,0];
imagenum=6
extra=0
num1=imagenum-1
answer=' '
nummin=0
sdev1x=fltarr(5)
sdev1y=fltarr(5)
dcount=intarr(5)
xy=fltarr(4,5)
;x=fltarr(11)
;y=fltarr(11)
find_5_dev,idut,xy                  ; point & click on 5 devices
print,' xy :',xy
for i=0,4 do begin
    sdev1x(i)= (xy(2,i)-xy(0,i)+1)
    sdev1y(i)= (xy(3,i)-xy(1,i)+1)
    print, 'sdev1x',sdev1x,'sdev1y',sdev1y                ;diagnostic
endfor      ;ends the for loop to determine how many pixels are in each device
dev1=fltarr(num,sdev1x(0),sdev1y(0))
dev2=fltarr(num,sdev1x(1),sdev1y(1))
dev3=fltarr(num,sdev1x(2),sdev1y(2))
dev4=fltarr(num,sdev1x(3),sdev1y(3))
dev5=fltarr(num,sdev1x(4),sdev1y(4))
dev1(*,*,*)=0
dev2(*,*,*)=0
```

```
dev3(*,*,*)=0
dev4(*,*,*)=0
dev5(*,*,*)=0
sdev1=size(dev1)
sdev2=size(dev2)
sdev3=size(dev3)
sdev4=size(dev4)
sdev5=size(dev5)
TJ1=fltarr(1,sdev1 (2),sdev1 (3))          ;temporary junk array #1
TJ2=fltarr(1,sdev2(2),sdev2(3))         ;temporary junk array #2
TJ3=fltarr(1,sdev3(2),sdev3(3))         ;temporary junk array #3
TJ4=fltarr(1,sdev4(2),sdev4(3))         ;temporary junk array #4
TJ5=fltarr(1,sdev5(2),sdev5(3))         ;temporary junk array #5
print, 'device 1 dimensions',sdev1(1),sdev1(2),sdev1(3)      ;diagnostic
print, 'device 2 dimensions',sdev2(1),sdev2(2),sdev2(3)      ;diagnostic
print, 'device 3 dimensions',sdev3(1),sdev3(2),sdev3(3)      ;diagnostic
print, 'device 4 dimensions',sdev4(1),sdev4(2),sdev4(3)      ;diagnostic
print, 'device 5 dimensions',sdev5(1),sdev5(2),sdev5(3)      ;diagnostic
dev_mask,aa,xy,dmask,dcount             ; create device extraction mask
check_mask,aa,dmask,idut                ; check mask against image
dev_save,xy,idut,TJ1,TJ2,TJ3,TJ4,TJ5  ;dut alone image
dev1(0,*,*)=TJ1(0,*,*)
dev2(0,*,*)=TJ2(0,*,*)
dev3(0,*,*)=TJ3(0,*,*)
dev4(0,*,*)=TJ4(0,*,*)
dev5(0,*,*)=TJ5(0,*,*)
dev_save,xy,ref, TJ1,TJ2,TJ3,TJ4,TJ5  ;reference alone image
dev1(1,*,*)=TJ1(0,*,*)
dev2(1,*,*)=TJ2(0,*,*)
dev3(1,*,*)=TJ3(0,*,*)
dev4(1,*,*)=TJ4(0,*,*)
dev5(1,*,*)=TJ5(0,*,*)
dev_save,xy,i0, TJ1,TJ2,TJ3,TJ4,TJ5   ;unbiased image
dev1(2,*,*)=TJ1(0,*,*)
dev2(2,*,*)=TJ2(0,*,*)
dev3(2,*,*)=TJ3(0,*,*)
dev4(2,*,*)=TJ4(0,*,*)
dev5(2,*,*)=TJ5(0,*,*)
dev_save,xy,i1, TJ1,TJ2,TJ3,TJ4,TJ5   ;biased image 1
dev1(3,*,*)=TJ1(0,*,*)
dev2(3,*,*)=TJ2(0,*,*)
dev3(3,*,*)=TJ3(0,*,*)
dev4(3,*,*)=TJ4(0,*,*)
dev5(3,*,*)=TJ5(0,*,*)
dev_save,xy,i2, TJ1,TJ2,TJ3,TJ4,TJ5   ;biased image 2
dev1(4,*,*)=TJ1(0,*,*)
dev2(4,*,*)=TJ2(0,*,*)
dev3(4,*,*)=TJ3(0,*,*)
dev4(4,*,*)=TJ4(0,*,*)
dev5(4,*,*)=TJ5(0,*,*)
dev_save,xy,i3, TJ1,TJ2,TJ3,TJ4,TJ5   ;biased image 3
dev1(5,*,*)=TJ1(0,*,*)
dev2(5,*,*)=TJ2(0,*,*)
dev3(5,*,*)=TJ3(0,*,*)
dev4(5,*,*)=TJ4(0,*,*)
dev5(5,*,*)=TJ5(0,*,*)
dev_save,xy,i4,TJ1,TJ2,TJ3,TJ4,TJ5    ;biased image 4
dev1(6,*,*)=TJ1(0,*,*)
dev2(6,*,*)=TJ2(0,*,*)
dev3(6,*,*)=TJ3(0,*,*)
dev4(6,*,*)=TJ4(0,*,*)
dev5(6,*,*)=TJ5(0,*,*)
```

```
dev_save,xy,i5, TJ1,TJ2,TJ3,TJ4,TJ5    ;biased image 5
dev1(7,*,*)=TJ1(0,*,*)
dev2(7,*,*)=TJ2(0,*,*)
dev3(7,*,*)=TJ3(0,*,*)
dev4(7,*,*)=TJ4(0,*,*)
dev5(7,*,*)=TJ5(0,*,*)
window,13,xsize=50,ysize=50,xpos=140,ypos=300,title='mainpro'
tvscl,dev1(0,*,*)
;xy_positions, x, y        ;This gets the individual positions of x and y of the devices
save,dev1,dev2,dev3,dev4,dev5,aa,dmask,sdev1x,sdev1y,volt,num,filename='d:/d1117bdevices.dat'
save,ref,idut,i0,i1,i2,i3,i4,i5,xy,sdev1,sdev2,sdev3,sdev4,sdev5,filename='d:\aaraw1117b.dat'

end
```

```
;This program does the analysis on the saved devices.
;Written by Elise Michaels
;File name: main121b1.pro
;Procedures contained within this program file: plot_dev. volt_phase. all_dev. all_dev_bin.
            all_parts. stuff. dev_op,. And the function phase.



;*******************************************************************
;plot_dev plots the calculated phase of the devices and plots them
;*******************************************************************

pro plot_dev,phas1,phas2,phas3,phas4,phas5, avph, std, num,hisans
;PLOTS THE DEVICE USING THE MASK AND CALCULATED PHASE
avph=fltarr(5,num-2)        ;avph(# of devices,# of biased images)
std=fltarr(5,num-2)
count=0
pgood1=0
pgood2=0
pgood3=0
pgood4=0
pgood5=0
sphas=size(phas1)  ;Does the first device, the other devices will be cycled through later.
startx=0
endx=sphas(2)-1    ;takes the 2nd array dim. And subtracts 1 to account for the zero.
starty=0
endy=sphas(3)-1         ;takes the 3rd array dim and subtracts 1 to account for the zero.
;print,'pgood',pgood          ;diagnostic
;print,'startindex',startindx,'endindex',endindx    ;diagnostic

    good1=where(phas1 lt 999,ngood1)      ;this selects the good points with in the phase array
    ;print,good1                          ;diagnostic
    good2=where(phas2 lt 999,ngood2)  ;the ngood's are scalars and the goods are arrays
    good3=where(phas3 lt 999,ngood3)
    good4=where(phas4 lt 999,ngood4)
    good5=where(phas5 lt 999,ngood5)
    maxph=fltarr(5)
    minph=fltarr(5)
  if (ngood1 GT 0) then begin  ;this checks to see if there is at least one good value which is less then 999.
        maxph(0)=max(phas1(good1))  ;This section get the max and min of the phase of each device for all bias
        minph(0)=min(phas1(good1))        ;conditions.
    endif else begin
        maxph(0)=1
        minph(0)=-1
    endelse
if (ngood2 GT 0) then begin
        maxph(1)=max(phas2(good2))
        minph(1)=min(phas2(good2))
    endif else begin
        maxph(1)=1
        minph(1)=-1
    endelse
if (ngood3 GT 0) then begin
    maxph(2)=max(phas3(good3))
    minph(2)= min(phas3(good3))
endif else begin
        maxph(2)=1
        minph(2)=-1
    endelse
if (ngood4 GT 0) then begin
        maxph(3)=max(phas4(good4))
```

```
        minph(3)= min(phas4(good4))
endif else begin
        maxph(3)=1
        minph(3)=-1
    endelse
if (ngood5 GT 0) then  begin
        maxph(4)=max(phas5(good5))
        minph(4)= min(phas5(good5))
    endif else begin
        maxph(4)=1
        minph(4)=-1
    endelse
    print, 'Device maximums', maxph, 'Device minimums', minph             ;diagnostic
 window,0,xsize=350,ysize=350,xpos=10,ypos=30,title='device 1'
 window,1,xsize=350,ysize=350,xpos=361,ypos=30,title='device 2'
 window,2,xsize=350,ysize=350,xpos=712,ypos=30,title='device 3'
 window,3,xsize=350,ysize=350,xpos=10,ypos=381,title='device 4'
 window,4,xsize=350,ysize=350,xpos=361,ypos=381,title='device 5 or 6'
    phas=phas1      ;the phas function will allow us to cycle through all of the devices
 for i=0,4 do begin                                   ;does all 5 devices
  wset,i                            ;sets the window for the data to be plotted in
  for j=0+hisans,(num-3+hisans) do begin
        print,'Condition',j,'start x & y',startx,starty,'end x & y',endx,endy     ,'Device',i+1          ;diagnostic
    phg=phas(j+hisans,startx:endx,starty:endy )
    pgood=where(phg lt 999,count)
    if (count gt 1) then begin
      ;print,'pgood',pgood          ;diagnostic
      avph(i,j)=avg( phg(pgood))/(2*!pi)  ; convert to waves
      if (stdev(phg(pgood))/(2*!pi) gt 1) then std(i,j)=stdev(phg(pgood))/(2*!pi) else begin
                 print,'bad stdev'
                 std(i,j)= 1
        end ; end of if stdev gt 1
        endif ;for if  statement for count gt 1
    plot,phas(0+hisans,startx:endx,starty:endy), yrange=[minph(i),maxph(i)], thick=3, charthick=2, background=255
    xyouts,.01,.1,/normal,string(0)+'  '+string(avph(i,j))                    ;plots the unbiased image
        for k=0+hisans,(num-3+hisans)  do begin
                oplot,phas(k,startx:endx, starty:endy),color=(k)*20 ,thick=3, charthick=2, background =255
      xyouts,.01,.1+.1*(k),/normal,string(k)+'  '+string(avph(i,k)),color=(k)*20
                       ;plots the biased images over the unbiased image
     endfor   ;for the k loop
   endfor  ;for the j increment
    if i eq 1 then begin            ;changes for device 2
            phas=phas2
            sphas=size(phas2)
    endif
    if i eq 2 then  begin           ;changes for device 3
            phas=phas3
            sphas=size(phas3)
    endif
    if i eq 3 then begin                 ;changes for device 4
            phas=phas4
            sphas=size(phas4)
    endif
    if  i eq 4 then begin           ;changes for device 5 or 6
        phas=phas5
        sphas=size(phas5)
    endif
    endx=sphas(2)-1          ;gets the x size of the next array
    endy=sphas(3)-1          ;gets the y size of the next array
 endfor    ; for the i incriment

print,'     image  D1        D2        D3        D4        D6'
```

```
for k=0+hisans,(num-3+hisans) do print,k,avph(*,k)
end

;*******************************************************************
;volt_phase generates the voltage information from each set of data and prints
;it out in a usable format.
;*******************************************************************

pro volt_phase,avph,volt,devnum,jfile,std,num
;vo=fltarr(5,num)
;av=fltarr(5,num)
;st=fltarr(5,num)
num1=num-1
print,'Device :',devnum
print,'V(gate)   V(drain)     Phase'
for i=2,num1 do print,volt(0,i),volt(devnum-1,i-2),avph(devnum-1,i-2)

;window,2*devnum,xsize=400,ysize=400,ypos=300,xpos=800-100*devnum
;plot,avph(devnum-1,*),ytitle='Phase Angle (Radians)',psym=5,title=jfile+' FET # :'+string(devnum)
;for i=0,num1 do xyouts,i,.2*max(avph(devnum-1,*)),string(volt(devnum-1,i),format='(F3.1)')
;for i=0,num1 do xyouts,i,.1*max(avph(devnum-1,*)),string(volt(0,i),format='(F3.1)')

window,devnum+4,xsize=400,ysize=400,ypos=40,xpos=800-100*devnum
plot,volt(0,*),avph(0,*), thick=3, charthick=2, background=255, $
  ytitle='Phase Angle (waves)',$
  psym=5,title=jfile+' FET # :'+string(devnum,format='(i3)'),xtitle='Drain Voltage'
;vo(*,*)=volt(*,*)
;av(*,*)=avph(*,*)
;st(*,*)=std(*,*)
oploterr,volt,avph,std,num1
!psym=0
end

;*******************************************************************
;all_dev generates a journal of all the data except for the graphs
;*******************************************************************

pro all_dev,avph,volt,xy,aa,jfile,std,num
journal,'c:\users\elise\'+jfile+'.txt'
print,'aa',aa            ;prints the active area array
print,'xy'                      ;prints the xy values of each device
print,xy
for i=1,5 do volt_phase,avph,volt,i,jfile,std,num
window,0
ymin=min(avph,max=ymax)
plot,volt(0,*),avph(0,*),yrange=[ymin,ymax], $
      thick=3, charthick=2, background=255, $
      xtitle='drain volts',title=jfile,ytitle='phase'
xyouts,.9,.1,/normal,string(1,format='(i2)')
for i=1,4 do begin
  oplot,volt(i,*),avph(i,*),color=i*40, thick=3, charthick=2,background=255
  xyouts,.9,.2+i*.1,/normal,color=i*40,string(i+1,format='(i2)')
 endfor
journal
end

;*******************************************************************
;all_dev_bin generates a journal of all the data except for the graphs
;*******************************************************************

pro all_dev_bin,avph,volt,xy,aa,jfile,std,num,nummin
journal,'c:\users\elise\'+jfile+'.txt'
```

```
print,'aa',aa
print,'xy'
print,xy
for i=1,5 do begin
     for k=0,nummin do volt_phase_bin,avph,volt,i,jfile,std,num,k
endfor
window,0
ymin=min(avph,max=ymax)
plot,volt(0,*),avph(0,*,*),yrange=[ymin,ymax], $
      thick=3, charthick=2, background=255, $
      xtitle='drain volts',title=jfile,ytitle='phase'
xyouts,.9,.1,/normal,string(1,format='(i2)')
for i=1,4 do begin
  oplot,volt(i,*),avph(i,*,*),color=i*40, thick=3, charthick=2,background=255
  xyouts,.9,.2+i*.1,/normal,color=i*40,string(i+1,format='(i2)')
 endfor
journal
end


;*******************************************************************
;function phase calculates the phase of each device. (all at once)
;*******************************************************************

function phase,dev1,sd
; CALCULATE PHASE ANGLE
test=dev1(0,*,*)*dev1(1,*,*)
nz=where(test ne 0,cnt)
ac=dev1(*,*,*)
angle=dev1(*,*,*)
angle(*,*,*)=-1.0*!pi
ac(*,*,*)=-1.0*!pi
if cnt gt 0 then begin
     for k=2,(sd(1)-1) do begin                ;for bias condition
          for i=0,(sd(2)-1) do begin           ;for the x coordinate
               for j=0,(sd(3)-1) do begin      ;for the y coordinate
                    ac(k,i,j)=(dev1(k,i,j)-dev1(1,i,j)-dev1(0,i,j))/(2*sqrt(dev1(1,i,j)*dev1(0,i,j)))
                    ;print, 'Bias condition', k, 'X position' ,I,'Y Position', j            ;diagnostic
               endfor        ;ends the j loop for the y positions of the image
          endfor             ;ends the i loop for the x positions of the image
     endfor                  ;ends the k loop for the images
endif                           ;ends the if condition (cnt gt 0).
bad=where((ac gt 1) or (ac lt (-1)),nbad)
ac = ac < 1
ac = ac > (-1)
angle(0,*,*)=1000
angle(1,*,*)=1000
for k=2,sd(1)-1 do begin
angle(k,*,*)=acos(ac(k,*,*))
if (nbad gt 0) then angle(bad)=1000.  ; flag bad points
end
return,angle
end

;*****************************************************************************
;This will process the phases and return a final phase values for both the plus and minus delta
;phases
;*****************************************************************************


pro all_parts, xy, dev, php0
    si1=size(dev)
```

```
    n=si1(1)         ;number of bias conditions
    m=si1(2)         ;x positions
    o=si1(3)         ;y positions
    sphp0=size(php0)
    a=sphp0(1)       ;number of bias conditions
    b=sphp0(2)       ;x positions
    c=sphp0(3)       ;y positions
    print, 'php0 dimension 1',a,' dimension 2',b,' dimension 3',c
    temp=fltarr(n,m,o)
    temp(*,*,*)=0         ;sets a new array to zero
    temp=dev                 ;fills the new array
    sum=intarr(n,m,o)
    phplus=fltarr(a,b,c)     ;Biased phase plus unbiased phase
    ;his1=fltarr(256)
    answer=''
    value=0
    number=0
    count=0

    ;THIS SECTION DOES THE HISTOGRAM ON THE DELTA PHASES (BOTH PLUS AND MINUS
    ; THE UNBIASED IMAGES)
byte_phase_p=byte(php0*256/!pi)  ;converts php0 to a byte value between 0 - 255, 3D array
phase_his_p=bytarr(256)
;for k=0,a-1 do begin              ;bias condition of a device
    for i=0,b-1 do begin           ; x position of device
        for j=0,c-1 do begin       ; y position of device
            phase_his_p(byte_phase_p(0,i,j))= phase_his_p(byte_phase_p(0,i,j))+1
                ;this section is filling the byte array where the byte_phase_p has the same value thru the array
                ;Thus this is the histogram of the added and subtracted unbiased image from the biased images.
        endfor            ;ends the j for loop to get the y positions of the image
    endfor                ;ends the I for loop to get the x positions of the image
;   endfor                ;ends the k for loop to get the bias condition of each device
window,    17,xsize=924, ysize=256, title = 'Histogram of adjusted phases.'
plot, phase_his_p, xrange=[0,275], xtitle='Phase value', ytitle='# of pixels', $
linestyle=1, thick=3,background=256 , title='Delta Phase Histograms'
    ;oplot, phase_his_m, linestyle=2, thick=4,color=195
print,'Are there any ranges which may be a device (for the added phase values)? (y or n)'
read,answer
if answer eq 'y' then begin
    print,'How many ranges are there?'
    read,value
    his_r=fltarr(2*value+2)
    his_r(0)=0
    his_r(2*value+1)=255
    maskdev=fltarr(a,b,c)
    maskdev(*,*,*)=0                        ;maskdev(bias condition, xposition, yposition)
    print,'Please input the edges of each range.'
    for i=1,2*value do begin
        read, number
        his_r(i)=number                     ; gets the ranges for the historgram
    endfor       ;ends the for loop which gets the ranges of each possible device
    window,10,xsize=m*10,ysize=o*10,xpos=100,ypos=50,title='Possible devices'
    answ = ''
  fcntp=0
    for j=0,2*value do begin
            print,'Histogram range',j           ;diagnostic
   if (answ ne 'y') OR ((fcntp) GT 0) then begin
            for t= 0,a-1  do begin
                                ;This section will help the user locate a mask within a histogram
                               ;range and save that mask to apply to all of the other bias
                             ;conditions for one device
             for y=0,b-1   do begin
```

```
            for z=0, c-1 do begin
             If ((byte_phase_p(t,y,z) GT his_r(j)) and (byte_phase_p(t,y,z) LT his_r(j+1))) then begin
                 maskdev(t,y,z)=1
        FOR x=0,a-1 DO maskdev(x,y,z)=maskdev(t,y,z)
            ENDIF
             endfor            ;ends the z for loop
            endfor                   ;ends the y for loop
          rangemult=rebin(maskdev,a,b*10,c*10./sample)
            tvscl,rangemult(t,*,*)
            Print,'Does this appear to match device(s)?'
            read,answ
            if answ eq 'y' then begin
        fcntp=1
            for x=0 , a-1 do begin
             for y= 0, b-1 do begin
                    for z= 0, c-1 do begin
                         if maskdev(x,y,z) eq 1 then begin
                              phplus(x,y,z)=php0(x,y,z)
                         endif else phplus(x,y,z)=1000
                    endfor           ;ends the z loop
                endfor               ;ends the y loop
            endfor                   ;ends the x loop
          ENDIF      else begin      ;ends the if-then statement : a device match is found
                    print, 'There is no device.'
          maskdev(*,*)=0
                    wdelete
                    window,10,xsize=m*10,ysize=o*10,xpos=100,ypos=50,title='Possible devices'
                endelse                        ;ends the if not yes then-else
    endfor                     ;ends the t for loop

        endif       ;ends the if-then statement for if answ ne 'y'
       endfor                  ;ends the for j loop for the ranges between 0 and 2*value
   endif                              ;ends the if-then statement for answer eq 'y' for ranges
                                             ;on the histogram for the added phases

   window,11,xsize=b*10,ysize=c*10,xpos=200,ypos=100, title = 'Added phases'
   image1=phplus
   image1(*,*,*)=0
   ppcnt=0
   pp= where((phplus LT 1000),ppcnt)
   if ppcnt gt 0 then image1(pp)=1
  imager=rebin(image1,a,b*10,c*10./sample)
  tvscl,imager(1,*,*)
   Print,'Does this appear to be possible devices? (y or n)'
   reply=''
   read,reply
   if reply EQ 'y' then php0=phplus     else php0=php0
   ;ends the if statement which changes php0 to the final values.
end                             ;ends the procedure all_parts

;****************************************************************************
;
;****************************************************************************

pro stuff, num,i1,phas,x,y,volt,jfile
phm0=phas
php0=phas
num1=num-1
for i=0,num1 do begin & $
  bad=where((phas(i,*) gt 999.0) or (phas(0,*) gt 999.0),nbad) & $
  phm0(i,*)=phas(i,*)-phas(0,*) & $   ; subtract 0 phase case
  php0(i,*)=phas(i,*)+phas(0,*) & $   ; add 0 phase case
```

```
  if (nbad gt 0) then begin & phm0(i,bad)=2000.0 & php0(i,*)=2000.0 & $
   end & $
endfor  ;ends for i loop to add and subtract and add the unbiased phase
     ;from a biased phase
all_parts, x,y, i1, php0,phm0,phas

plot_dev,php0,dcount,avph,std,num

device,/cursor_original
Print,'This section is the phase of a biased image set plus the unbiased image set.'
all_dev,avph,volt,xy,aa,jfile,std,num
window,0
v4=fltarr(num)
d40=fltarr(num)
st4=fltarr(num)
v4(*)=volt(3,*)
d40(*)=avph(3,*)-avph(3,0)
st4(*)=std(3,*)

plot,v4,d40,xtitle='Drain Voltage',$
ytitle='Phase angle (wave)',charsize=1.3,$
     thick=2,title='Response of 20/20 FET ',yrange=[0,.2]
oploterr,v4,d40,st4,num1
window,2
;v1=fltarr(num)
;d10=fltarr(num)
;st1=fltarr(num)
;v1(*)=volt(0,*)
;d10(*)=avph(0,*)-avph(0,0)
;st1(*)=std(0,*)

;plot,v1,d10,xtitle='Drain Voltage',$
;     ytitle='Phase angle (wave)',charsize=1.3,$
 ;    thick=2,title='Response of 1344/0.7 FET ',yrange=[0,.2]
;oploterr,v1,d10,st1,num1
;!psym=0

wait,60

plot_dev,phm0,dcount,avph,std,num
Print,'This section is the phase of a biased image set minus the unbiased image set.'
all_dev,avph,volt,xy,aa,jfile,std,num
v4=fltarr(num)
d40=fltarr(num)
st4=fltarr(num)
v4(*)=volt(3,*)
d40(*)=avph(3,*)-avph(3,0)
st4(*)=std(3,*)
plot,v4,d40,xtitle='Drain Voltage',$
ytitle='Phase angle (wave)',charsize=1.3,$
thick=2,title='Response of 20/20 FET ',yrange=[0,.2]
oploterr,v4,d40,st4,num1
!psym=0
end                              ;ends procedure stuff

;******************************************************************************
;Program to do the histogram and determine device operation
;******************************************************************************

pro dev_op, dev,ph0,phas

sdev=size(dev)
```

```
n=sdev(2)
m=sdev(3)
o=sdev(1)
print, 'X dimension of  the device',n, 'Y dimension of the device',m
temp=fltarr(1,n,m)
temp(0,*,*)=0
temp(0,*,*)=dev(0,*,*)
tempr=fltarr(1,n,m)
tempr(0,*,*)=0
bdppl =dev(*,*,*)
bdpp1(*,*,*)=0
sumn=intarr(1,n,m)
phplus=fltarr(o,n,m)
mask=fltarr(o,n,m)
mask(*,*,*)=0
answer=''
value=0
number=0
count=0
phase_p=bytarr(256)
bpp=byte(ph0*256/!pi)            ;converts ph0 to a byte value between 0 and 255
for i=0,n-1 do begin
     for j=0,m-1 do begin
          phase_p(bpp(0,i,j))= phase_p(bpp(0,i,j))+1 ;fills the phase_p with a number that will be charted later
                                                                                            ; in order to
determine the device activity
     endfor                 ;ends the j for loop
endfor                      ;ends the i for loop
window,    17,xsize=1024, ysize=256, title = 'Histogram of adjusted phases.'
plot, phase_p , xrange=[0,256], xtitle='Phase value', ytitle='# of pixels in each histogram value', $
linestyle=1,color=50, thick=3, title='Delta Phase Histograms'
print, 'Are there any ranges which maybe a device? (y or n)'
read, answer
if answer eq 'y' then begin
     print, 'How many ranges are there?'
     read, value
     his_r=fltarr(2*value+2)
     his_r(0)=0
     his_r(2*value+1)=255
maskdev=bytarr(256)
     print,'Please input the edges of each range.'
     for i=1,2*value  do begin
          read, number
          his_r(i)=number
     endfor      ;ends the for loop which gets the ranges of each possible device
     window,10,xsize=sdev(2)*20,ysize=sdev(3)*20, xpos=100,ypos=400,title='Possible image'
for j=0,2*value do begin
               print,'Histogram range',j
               bdppl=where ((bpp(0,*,*) GT his_r(j)) and (bpp(0,*,*) LT his_r(j+1)),count)
               if count GT 0 then begin
                    maskdev(bdppl)=1
                    smaskdev=size(maskdev)
                    rangemult=rebin(maskdev,smaskdev(2)*4,smaskdev(3)*4,/sample)
                    tvscl,rangemult
                    answ=''
                    Print,'Does this appear to match devices?'
                    read,answ
                    if answ eq 'y' then begin
                         for z=0,o-1 do begin
                              for k=0,n-1 do begin
                                   for l=0,m-1 do begin
                                        if ((bpp(z,k,l) GT his_r(j)) and (bpp(z,k,l) LT his_r(j+1))) then  begin
```

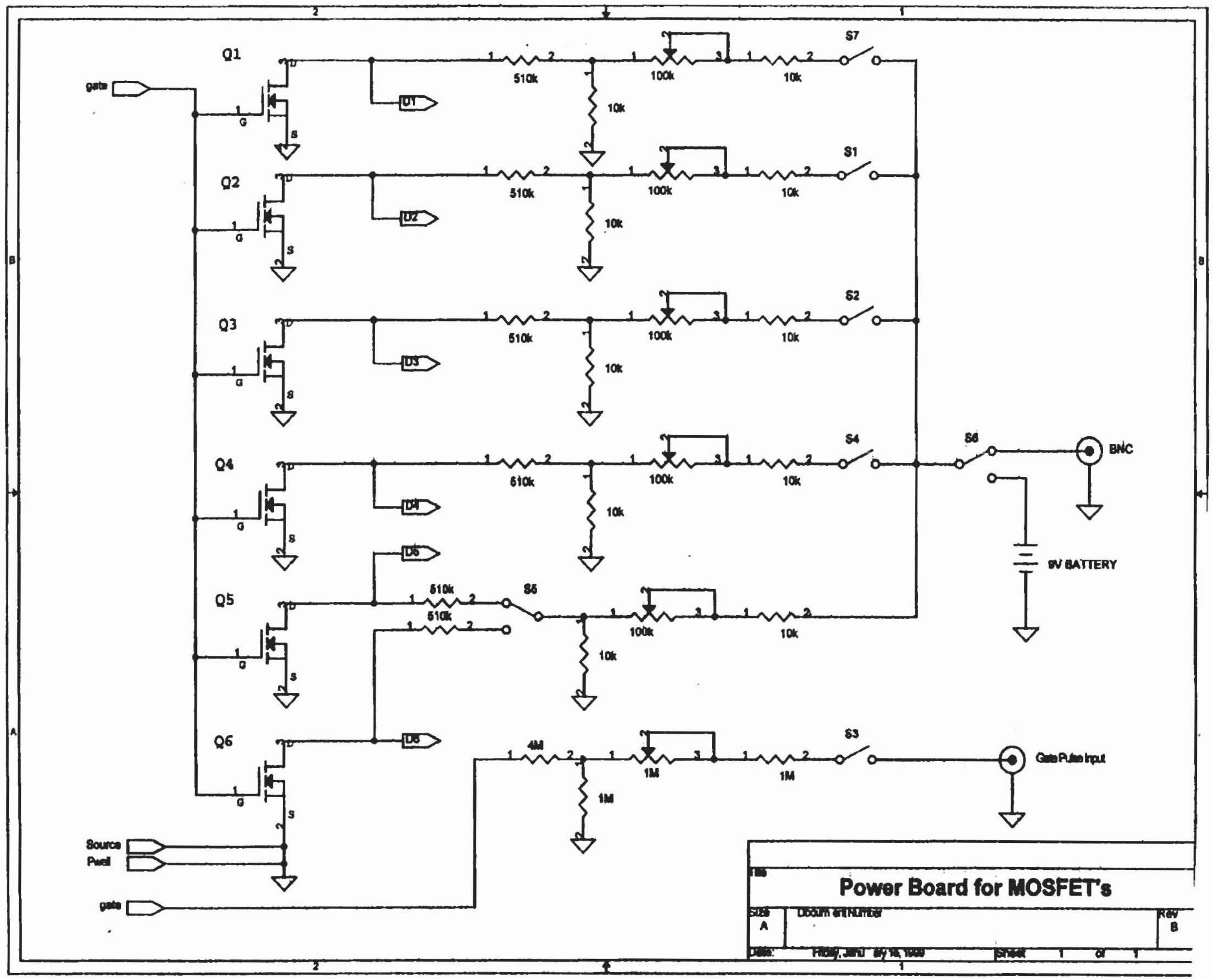
```
                        phplus(z,k,l)=dev(z,k,l)
                        mask(z,k,l)=1
                    endif else   phplus(k,l)=0
                endfor          ;ends the l for loop to get the y positions of the image
            endfor              ;ends the k for loop to get the x positions of the image
        endfor                  ;ends the z for loop to cycle through the same device of several images
    endif       else print, 'There is no devices.'     ;ends the if-then statement for if a device match is
found
  endif    else count=0       ;ends the if-then statement for if count GT 0
 endfor                  ;ends the for j loop for the ranges between 0 and 2*value
endif                        ;ends the if there are possible devices
for i=0,sdev(1) do ph0(i,*,*)=ph0(i,*,*)*mask(i,*,*)

end             ;ends the procedure dev_op
```

```
;This program will process any of the data sets after the devices have been separated out.
; Written by: Elise Michaels
;File name:danalysis.pro
; This program uses the procedures: plot_dev. volt_phase. all_dev. all_dev_bin.
;    all_parts. stuff. dev_op. In addition  the function Phase is used within this program.

;num=0
;print,'How many data sets are there (include the reference and DUT alone)?'
;read,num
phas1=fltarr(num,sdev1(2),sdev1(3))          ; create array for phase angles of dev1
phas2=fltarr(num,sdev2(2),sdev2(3))          ; create array for phase angles of dev2
phas3=fltarr( num,sdev3(2),sdev3(3))         ; create array for phase angles of dev3
phas4=fltarr(num,sdev4(2),sdev4(3))          ; create array for phase angles of dev4
phas5=fltarr(num,sdev5(2),sdev5(3))          ; create array for phase angles of dev5
;angle1=fltarr(num,total(dcount))            ; read phase at each voltage state
phas1(*,*,*)=phase(dev1,sdev1)              ;device 1 -all biased and unbiased images
phas2(*,*,*)=phase(dev2,sdev2)              ;device 2 -all biased and unbiased images
phas3(*,*,*)=phase(dev3,sdev3)              ;device 3 -all biased and unbiased images
phas4(*,*,*)=phase(dev4,sdev4)              ;device 4 -all biased and unbiased images
phas5(*,*,*)=phase(dev5,sdev5)              ;device 5 -all biased and unbiased images
jfile='dat11-17-96b'
php0a=fltarr(num-2,sdev1(2),sdev1(3))
php0b=fltarr(num-2,sdev2(2),sdev2(3))
php0c=fltarr(num-2,sdev3(2),sdev3(3))
php0d=fltarr(num-2,sdev4(2),sdev4(3))
php0e=fltarr(num-2,sdev5(2),sdev5(3))
for i=3,num-1 do begin
    php0a(i-3,*,*)=phas1((i),*,*)+phas1(2,*,*)
    php0b(i-3,*,*)=phas2((i),*,*)+phas2(2,*,*)
    php0c(i-3,*,*)=phas3((i),*,*)+phas3(2,*,*)
    php0d(i-3,*,*)=phas4((i),*,*)+phas4(2,*,*)
    php0e(i-3,*,*)=phas5((i),*,*)+phas5(2,*,*)
endfor      ;ends the adding of the unbiased phase to the biased phases
phm0a=fltarr(num-2,sdev1(2),sdev1(3))
phm0b=fltarr(num-2,sdev2(2),sdev2(3))
phm0c=fltarr(num-2,sdev3(2),sdev3(3))
phm0d=fltarr(num-2,sdev4(2),sdev4(3))
phm0e=fltarr(num-2,sdev5(2),sdev5(3))
for i=3,num-1 do begin
    phm0a(i-3,*,*)=phas1((i),*,*)-phas1(2,*,*)
    phm0b(i-3,*,*)=phas2((i),*,*)-phas2(2,*,*)
    phm0c(i-3,*,*)=phas3((i),*,*)-phas3(2,*,*)
    phm0d(i-3,*,*)=phas4((i),*,*)-phas4(2,*,*)
    phm0e(i-3,*,*)=phas5((i),*,*)-phas5(2,*,*)
endfor
avph=fltarr(5,num)
std=fltarr(5,num)
hisans=0
all_parts, xy, dev1, php0a
;all_parts, xy, dev1, phm0a
all_parts, xy, dev2, php0b
;all_parts, xy, dev2, phm0b
all_parts, xy, dev3, php0c
;all_parts, xy, dev3, phm0c
all_parts, xy, dev4, php0d
;all_parts, xy, dev4, phm0d
all_parts, xy, dev5, php0e
;all_parts, xy, dev5, phm0e
plot_dev,php0a,php0b,php0c,php0d,php0e,avph,std,num,hisans
print, 'Reference =wedge and DUT=good Chip.'
Print, 'The phase of a biased image set plus the unbiased image set.'
all_dev,avph,volt,xy,aa,jfile,std,num
```

```
window,23
v4=fltarr(5,num)
d40=fltarr(5,num-2)
st4=fltarr(5,num-2)
v4(3,*)=volt(3,*)
d40(3,*)=avph(3,*)-avph(3,0)
st4(3,*)=std(3,*)
plot,v4(3,*),d40(3,*),xtitle='Drain Voltage',$
ytitle='Phase angle (wave)',charsize=1.3,$
     thick=2,title='Response of 20/20 FET ',yrange=[0,.2]
oploterr,v4(3,*),d40(3,*),st4(3,*)
!psym=1

end
```

Appendix C:

The Cadstar drawing to follow is the circuitry which drives the Intel Chip under test.



Power Board for MOSFET's
Q1
Q2
Q3
Q4
Q5
Q6
gate
Source
Pwell
gate
D1
D2
D3
D4
D5
D6
510k
100k
10k
1M
4M
S1
S2
S3
S4
S5
S6
S7
BNC
9V BATTERY
Gate Pulse Input
Size
A
Rev
B
Sheet 1 of 1