University of Arkansas, Fayetteville

ScholarWorks@UARK

Graduate Theses and Dissertations

5-2017

Design and Test of a Gate Driver with Variable Drive and Self-Test Capability Implemented in a Silicon Carbide CMOS Process

Matthew Barlow
University of Arkansas, Fayetteville

Follow this and additional works at: https://scholarworks.uark.edu/etd

Part of the Electronic Devices and Semiconductor Manufacturing Commons, and the VLSI and Circuits, Embedded and Hardware Systems Commons

Citation

Barlow, M. (2017). Design and Test of a Gate Driver with Variable Drive and Self-Test Capability Implemented in a Silicon Carbide CMOS Process. *Graduate Theses and Dissertations* Retrieved from https://scholarworks.uark.edu/etd/1976

This Dissertation is brought to you for free and open access by ScholarWorks@UARK. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of ScholarWorks@UARK. For more information, please contact scholar@uark.edu, uarepos@uark.edu.

Design and Test of a Gate Driver with Variable Drive and Self-Test
Capability Implemented in a Silicon Carbide CMOS Process

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Electrical Engineering

by

Matthew W. Barlow
University of Arkansas
Bachelor of Science in Electrical Engineering, 2007
University of Arkansas
Master of Science in Electrical Engineering, 2012

May 2017
University of Arkansas

This dissertation is approved for recommendation to the Graduate Council.

____

Dr. H. Alan Mantooth
Dissertation Director

____

Dr. Simon Ang
Committee Member

____

Dr. Jia Di
Committee Member

____

Dr. Michael Glover
Committee Member

____

Dr. A. Matthew Francis
Committee Member

Abstract

Discrete silicon carbide (SiC) power devices have long demonstrated abilities that outpace those of standard silicon (Si) parts. The improved physical characteristics allow for faster switching, lower on-resistance, and temperature performance. The capabilities unleashed by these devices allow for higher efficiency switch-mode converters as well as the advance of power electronics into new high-temperature regimes previously unimaginable with silicon devices. While SiC power devices have reached a relative level of maturity, recent work has pushed the temperature boundaries of control electronics further with silicon carbide integrated circuits.

The primary requirement to ensure rapid switching of power MOSFETs was a gate drive buffer capable of taking a control signal and driving the MOSFET gate with high current required. In this work, the first integrated SiC CMOS gate driver was developed in a 1.2 μm SiC CMOS process to drive a SiC power MOSFET. The driver was designed for close integration inside a power module and exposure to high temperatures. The drive strength of the gate driver was controllable to allow for managing power MOSFET switching speed and potential drain voltage overshoot. Output transistor layouts were optimized using custom Python software in conjunction with existing design tool resources. A wafer-level test system was developed to identify yield issues in the gate driver output transistors. This method allowed for qualitative and quantitative evaluation of transistor leakage while the system was under probe. Wafer-level testing and results are presented.

The gate driver was tested under high temperature operation up to 530 °C. An integrated module was built and tested to illustrate the capability of the gate driver to control a power MOSFET under load. The adjustable drive strength feature was successfully demonstrated.

©2017 by Matthew Barlow
All Rights Reserved

Acknowledgements

My partners in the lab have been essential to my success. From helping with test fixtures and lab setups, to helping make a large project possible, to making jokes and keeping spirits high, to giving feedback on my ideas. Specifically, (and in alphabetical order), I would like to thank Nick Chiolino, Kim Cornett, Attanu Dutta, Matt Francis, Guoyuan Fu, Obidiah Kegege, Chris Lee, Javier Valle Mayorga, Ashfaqur Rahman, Sayan Seal, and Paul Shepherd.

I would also like to thank Dr. H. Alan Mantooth for providing the opportunities throughout my time at the University of Arkansas for interesting and productive research.

I would like to thank Ewan P. Ramsay, Robin F. Thompson, Robert A. R. Young, and Jennifer D. Walls from Raytheon UK for their help and support during the design and fabrication of this work.

I would also like to thank Dr. Micheal Glover and Micheal Steger, of the High Densitity Electronics Center at the University of Arkansas. Their electronics packaging expertise was essential for going from a fresh wafer to packaged and testable circuits.

This research was funded by the National Science Foundation Grant #IIP1237816. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation

Dedication

I am forever grateful to my family, for supporting me from the beginning to the end. My wonderful wife, Kathryn, who believes in me constantly. To my dear children, Abigail, Ethan, and Iriana for cheering me up with their boundless energy. To my loving mother and father, for providing me the foundation needed to even contemplate such an endeavor. I dedicate this work to all of you. Thank you so much, I could not have done it without all of you.

Table of Contents

1 Introduction 1

Organization 2

2 Silicon Carbide Overview 3

SiC in Power Electronics 6

SiC Integrated Circuits 7

Summary 10

3 Power Electronics Switching 11

Examination of switching topologies 11

Gate Drive Techniques 26

Silicon carbide challenges 30

Summary 31

4 Gate Driver Design 33

Design Goals and Specifications 33

Power Module Integration 37

Adjustable Drive Strength 38

Lessons Learned 57

5 Power FET Optimization 59

Parametric evaluation of problem space 68

Summary 83

6 Design for Test 84

Motivation 84

Characteristics of failures 87

In-circuit Fault Detection 94

Improved circuit testability 105

Wafer test results 107

Conclusions 115

Lessons Learned 116

7 Testing 117

High temperature test design 117

High temperature setup evaluation 120

Test Setup Configuration 124

Power module integration 136

Conclusions 153

Lessons Learned 154
8 Conclusions and Future Work 155
Future Work........ 156
A. Appendix: Python Code Base 162
Python Optimization of FETs........ 162
Gate Driver Test Interface 173

List of Figures

Fig. 2.1. Unipolar power device structures: vertical power MOSFET structure (a) and power Schottky diode structure (b). 4

Fig. 2.2. Example cross-section of a CMOS inverter in the Raytheon HiTSiC process 9

Fig. 3.1. Fundamental switch orientation that can be expanded into various power electronics circuits by the addition of passive components. 12

Fig. 3.2. Clamped-inductive load circuit, with optional synchronous MOSFET Q2 13

Fig. 3.3. Clamped inductive load schematic showing location of C_{GD} and C_{GS} 14

Fig. 3.4. Example MOSFET switching waveforms for a clamped-inductive load turn-on. 15

Fig. 3.5. Various load configurations for a single low-side switch. (a) shows a non-inductive resistive load, (b) shows a capacitance in parallel with the load, and (c) shows an inductive load with flyback diode 17

Fig. 3.6. Clamped inductive load configuration with charged inductor. D1 starts in conduction, and Q2 switches on, resulting in a fast rising V_{DS} on Q1. This rapid drain voltage rise has the potential to cause the V_{GS} of Q1 to rise above $V_{TH(on)}$ 17

Fig. 3.7. Gate charge and related capacitance parameters for a Cree C3M0065090J MOSFET. The gate charge parameters are linear in three separate regions as indicated by the dQ/dV line. Input capacitance is 813 pF in off region with a V_{DS} of 400 V and an I_{DS} of 20 A at 25 °C.[33] 18

Fig. 3.8. Parasitic turn-on sub-circuit with explicit internal capacitances and resistances illustrated for the purpose of deriving system behavior under a switching transient. 21

Fig. 3.9. Simplified gate drive network for identifying safe operating conditions. 22

Fig. 3.10. Example peak gate voltage after 10 ns time with an I_{CGD} of 200 mA. The C_{GS} used is 660 pF, which represents the gate voltage behavior switching 600 V in 12 ns. The critically damped boundary line is marked with circular markers. 24

Fig. 3.11. Taking the internal resistance and capacitances of devices from Fig. 3.8, parasitic interconnect inductances are added as lumped elements to complete the system model. Series inductances have been combined to simplify analysis. 26

Fig. 3.12. Various gate driver connection strategies. (a) shows a single supply connection, (b) shows a dual-rail power supply, (c) shows an active Miller clamp, and (d) shows a diode – resistor network to provide additional drive strength at turn-off. 28

Fig. 4.1 Gate driver output topologies possible with the Raytheon HiTSiC process. Part (a) shows a NPN pull-up device, (b) shows a NFET pull-up device, and (c) shows a PFET pull-up device. 35

Fig. 4.2. Voltage domains required for a NFET totem pole output. 35

Fig. 4.3. Gate drive slice strength modulation schematic. 39

Fig. 4.4. Input signal timing for multiple drive strength switching. 42

Fig. 4.5. Use of SR_OR pin to maintain drive functionality during drive strength programming. 43

Fig. 4.6. Dead-time generation and input control logic. 44

Fig. 4.7. Dead time generation logic simulation at 25 °C. The input signal on top (red) is delayed and turned into separate active high signals for the PFET pull-up (magenta) and the NFET pull-down (green). 45

Fig. 4.8. Single transistor slice drive logic for a NFET. The output connects directly to the gate of the transistor slice. 46

Fig. 4.9. Single transistor slice driver with tristate logic. 47

Fig. 4.10. Schematic for the test-mode drain voltage control and measurement. 48

Fig. 4.11. System block diagram 49

Fig. 4.12. Power transistor orientation options considered. 53

Fig. 4.13. Gate driver layout showing two NFET output transistor slices and the associated pads and buffers. 54

Fig. 4.14. Layout of the control logic for a single NFET slice. 54

Fig. 4.15. Control logic layout. 55

Fig. 4.16. Metal routing of V_{DD} and V_{SS} nets highlighted. Yellow area is metal 1 V_{SS}, and the red area is metal 1 V_{DD}. 56

Fig. 4.17. Final top-level layout for the integrated gate driver. Dimensions are 5.0 mm x 4.5 mm. 57

Fig. 5.1. Example layout output from the layout generation routine. In this design, there are five fingers stacked vertically (nf = 5), and there are four stacks of fingered transistors (m = 4). 63

Fig. 5.2. Schematic representation of SPICE testbench netlist. 65

Fig. 5.3. Comparison of effective transistor width versus the width of the main metal conductors. Transistor width is adjusted by changing the number of 60 μm wide transistor blocks (multiplicity). Designs are simulated over temperature, normalized to total layout area, and plotted. 69

Fig. 5.4 Parametric evaluation of the static current characteristics of a NFET over size and metal width. Current is presented as A/mm. Maximum current occurs when total width is 1.32 mm at 25 ℃ to 1.08 mm at 200 ℃. Spending additional area on metal width does not improve current density. 71

Fig. 5.5. Maximum current response while comparing the number of divisions in a design. 73

Fig. 5.6. Turn-on time comparing the number of divisions in a design. 74

Fig. 5.7 Turn-on speed comparing gate signal distribution sizes (wgb and wp). 76

Fig. 5.8. Current density of different gate signal distribution (gp v wgb). 77

Fig. 5.9 Turn-on time comparing poly distribution width and the number of fingers. 79

Fig. 5.10. PFET current density in A/mm^2. 81

Fig. 5.11. Maximum current in A/mm for a PFET. 82

Fig. 6.1. Bonding pad from the first pass after attempting 1 mil gold ball bonding three times. The gold coating on the pad (yellow) has been torn off where the ball bonds were made, leaving the underlying metal below. 86

Fig. 6.2. Gate oxide breakdown on a 1.2 μm channel length transistor during a V_{GS} sweep, with a V_{DS} of 0.1 V. Breakdown occurs at a V_{GS} of 10.5 V, and I_G reaches the programmed current limit of 10 mA. Subsequent runs indicate a gate-source short of 375 Ω. 88

Fig. 6.3. Topology of a drain-source short. 89

Fig. 6.4. Circuit response of an example power inverter over a range of drain-source leakage. A resistor connected to the drain and source of a NFET is varied in resistance from 0.1 Ω to 1 MΩ. The added resistance contributes to the quiescent current (green) draw up to the point where it overpowers the PFET pull-up. After that point, the output voltage (red) is affected, and current is limited by the saturation current of the PFET. 89

Fig. 6.5. Circuit response of an example power inverter over a range of gate-source leakage. A resistor is placed across the gate and source of the power NFET, and the value is varied from 0.1 Ω to 1 MΩ. The resistance contributes to increased quiescent current (green) of the system to the point where the resistance decreases the gate-source voltage (red). Smaller resistances result in a current limited by the saturation current of the output driver. 91

Fig. 6.6. Topology of a gate-source short. 91

Fig. 6.7. Topology of a gate-drain short. 93

Fig. 6.8. Circuit response of an example power inverter over a range of gate-drain leakage. A resistor is placed between the gate and drain of an output NFET and the value is swept from 0.1 Ω to 1 MΩ. The NFET gate is driven to 0 V with an expected drain voltage of 15 V. Quiescent current (blue) increases with decreasing resistance to the point where the NFET gate voltage rises above V_T. On the left side, the current becomes dominated by the NFET driven into the saturation region, drawing additional current as V_{GS} (green) rises and V_{DS} (red) falls. 93

Fig. 6.9. Weighted inverter thresholds, compared to a standard 1X inverter. The PFET-weighted inverter is shown in red, and the NFET-weighted inverter is shown in blue.......... 98

Fig. 6.10. Drain-source short characteristics for NFET (left) and PFET (right) devices. The red line indicates the larger $|V_{DS}|$ threshold from the weighted inverter pair, and the blue line indicates the lower threshold. 100

Fig. 6.11. Gate-source short characteristics for NFET (left) and PFET (right). The red curve represents the amount of time required to indicate a fault for an equivalent gate-source resistance.......... 101

Fig. 6.12. Gate-Drain short characteristics, with an initial condition of V_{GS} = 15V, and V_{DS} = 0V. The curve indicates the time required for the gate voltage to discharge to the drain voltage through the added equivalent gate-drain resistance, as measured from the gate. 102

Fig. 6.13. Gate-drain short characteristics, with an initial condition of V_{GS} = 0V, and V_{DS} = 15V. The weighted inverters measure the drain voltage as it leaks through to the gate, turning on the transistor. The red line indicates the higher $|V_{DS}|$ threshold, and the blue line indicates the lower $|V_{DS}|$ threshold.......... 102

Fig. 6.14. TVDD and TM_EN pad cell schematic. 106

Fig. 6.15. Example test configuration GUI for wafer level gate driver verification.......... 108

Fig. 6.16 R5C4 gate-drain Test Pass. The yellow trace is the test data input (TDI), the cyan trace is the test clock (TCLK), and the magenta trace is the test enable input (TEN). The green trace observes the serial data output (SDO) from the gate driver IC. The serial data output reads 0x00CFF, which indicates a passing result. 111

Fig. 6.17. R5C4 gate-source Test Pass. The yellow trace is the test data input (TDI), the cyan trace is the test clock (TCLK), and the magenta trace is the test enable input (TEN). The green trace observes the serial data output (SDO) from the gate driver IC. The serial data output reads 0xFF300, which indicates a passing result.......... 111

Fig. 6.18 R1C2 gate-drain Test with fault in 19th bit. The yellow trace is the test data input (TDI), the cyan trace is the test clock (TCLK), and the magenta trace is the test enable input

(TEN). The green trace observes the serial data output (SDO) from the gate driver IC. The serial data output reads 0x00CFE, which indicates a fault (expected 0x00CFF).... 112

Fig. 6.19 R1C2 gate-source test, with a fault in 6th bit. The yellow trace is the test data input (TDI), the cyan trace is the test clock (TCLK), and the magenta trace is the test enable input (TEN). The green trace observes the serial data output (SDO) from the gate driver IC. The serial data output reads 0xFD300, which indicates a fault (expected 0xFF300). 113

Fig. 7.1. RTD sensor configuration for hot finger thermocouple temperature correlation testing. The LDCC 68 package is identical to other packages used for testing. A blank SiC die was used to simulate the circuit under test, and the platinum RTD was epoxied to the top of the die. The RTD leads span multiple pins to allow for 4-wire Kelvin connections. Discolorations are due to the solder and high temperatures used during testing. 121

Fig. 7.2 Aluminum hot finger temperature calibration results. Hot plate used the thermocouple as feedback, resulting in temperature overshoot. 122

Fig. 7.3. Copper hot finger calibration results. Hot plate was controlled without feedback, resulting in the underdamped temperature response. Negative discontinuities were a result of intermittent thermocouple contact at 55 minute, and RTD lead shorting at 1:00:00 and 1:07:00. 123

Fig. 7.4. Temperature deviation between the thermocouple at the bottom of the LDCC 68 package and the top surface of the SiC die measured with the RTD. Measurement issues caused large discontinuities at points, major outliers have been removed. 124

Fig. 7.5. Hot plate testing using the QFP64 test configuration with the gate driver loaded with a C2M0280120D power MOSFET. Measured temperature at the bottom of the package is 455 ℃, with an input frequency of 1.0 MHz. 125

Fig. 7.6 Rise and fall times driving a C2M0280120D MOSFET at 15 V by W31 R3C2. 128

Fig. 7.7 Rise and fall times driving C2M0280120D at minimum drive strength by W31 R3C2. 128

Fig. 7.8 Comparison of rise and fall time variation over temperature while driving a C2M0280120D MOSFET by W31 R3C2. Times are from the minimum and maximum drive strengths, and are normalized to the value at 25 ℃. 129

Fig. 7.9 Propagation delay over temperature driving a C2M0280120D at 15 V by W31 R3C2. 129

Fig. 7.10. Representative driver output at 454 ℃ with a 1 MHz square wave driving C2M0280120D MOSFET by W31 R3C2... 130

Fig. 7.11. Gate driver output resistance over temperature, with a V_{DD} of 12 V. 132

Fig. 7.12. Gate driver output resistance of W31 R6C1 over temperature with a V_{DD} of 15 V... 132

Fig. 7.13. Gate driver rise and fall times over temperature at a V_{DD} of 12 V driving the gate of a C3M0065090D power MOSFET. 133

Fig. 7.14 Gate driver rise and fall times over temperature for die W31 R6C1 with V_{DD} = 15 V driving the gate of a C3M0065090D power MOSFET. 134

Fig. 7.15. Gate driver propagation delay with V_{DD} = 12 V driving the gate of a C3M0065090D power MOSFET. 134

Fig. 7.16. Gate driver propagation delay of W31 R6C1, with V_{DD} = 15 V driving the gate of a C3M0065090D power MOSFET. 135

Fig. 7.17. Representative waveform R6C1, 528 ℃, driving a C3M0065090D. 136

Fig. 7.18. Power module demonstration schematic. 137

Fig. 7.19. Assembled board with all components soldered and wirebonded. 138

Fig. 7.20. High voltage test setup with power supplies, function generator, FPGA board, oscilloscope, and module. 139

Fig. 7.21. Oscilloscope probe connection for the drain-source voltage measurement. A wire loop holds the ground portion of the oscilloscope probe and makes the connection to PVSS, and a wire target is soldered to VSW for the tip contact. The probe is taped using electrical tape (not shown) for additional mechanical stability. 141

Fig. 7.22. Representative oscilloscope waveform demonstrating the 2 pulse test configuration and turn-off waveform at a V_{BUS} of 300 V and an inductor current of 10 A. 142

Fig. 7.23. Turn-off waveforms at a constant inductor current across multiple bus voltages. Waveforms have been synchronized to the peak voltage after turn-off. The different load conditions are based on the inductor load current at the time of turn-off. 143

Fig. 7.24. A comparison of C_{OSS} of the C3M0090065 device across applied drain voltage. By assuming a fixed inductance in a resonant circuit, a projected series resonant frequency can be projected as a function of V_{DS}. The observed resonant frequency tracks between 4-6 nH loop inductance. 145

Fig. 7.25. Turn-on waveforms at a constant inductor current across multiple bus voltages. Waveforms have been synchronized to the 50 % transition point. 146

Fig. 7.26. Turn-on waveforms with varying inductor current and a 300 V DC bus voltage. 147

Fig. 7.27. Turn-off waveforms with varying inductor current and a 300 V DC bus voltage. Waveforms are roughly synchronized to the peak voltage after turn-off. 147

Fig. 7.28. Turn-on waveforms over various drive strengths. The load inductor current is 5 A with a nominal drain voltage of 60 V. 149

Fig. 7.29. Peak turn-on dv/dt while varying the drive strength. Switching conditions were a VBUS of 60 V and an inductor current of 5 A. The maximum dv/dt corresponds to any ringing after turn-on, and is near the observed noise level in the system. The minimum dv/dt is the main slew rate during switching. 150

Fig. 7.30. Rise and fall times as a function of drive strength. All situations are performed with a bus voltage of 60 V. Turn-on corresponds to a 5 A load current and to a turn-off 10 A load current. 151

Fig. 7.31. Turn-off waveforms over various drive strengths. The load inductor current is 10 A with a nominal drain voltage of 60 V. 152

Fig. 7.32. Turn-off dv/dt as a function of drive strength. All waveforms are with a nominal 60 V bus voltage and 10 A load inductor current. The positive dv/dt corresponds to the main rising voltage from turn-off, and the negative dv/dt matches with the falling voltage after the first drain voltage overshoot. 152

Fig. 7.33. Comparison of overshoot voltages and percent overshoot as a function of drive strength. 153

List of Acronyms and Abbreviations

AC	Alternating Current
AlN	Aluminum Nitride
BJT	Bipolar Junction Transistor
DBC	Direct Bond Copper
DC	Direct Current
DFT	Design For Testability
DRC	Design Rule Checks
DUT	Device Under Test
ENIG	Electroless Nickel Immersion Gold
ESD	Electro Static Discharge
GaN	Gallium Nitride
GUI	Graphical User Interface
I/O	Input / Output
IC	Integrated Circuit
IGBT	Isolated Gate Bipolar Transistor
LVS	Layout Versus Schematic
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
NFET	N-channel Field Effect Transistor
PEX	Parasitic Extraction
PFET	P-channel Field Effect Transistor
SiC	Silicon Carbide
SJT	Super Junction Transistor
SMU	Source Measure Unit
SOI	Silicon On Insulator
TSV	Through-Silicon Via
ZCS	Zero Current Switching
ZVS	Zero Voltage Switching

1 Introduction

For decades, silicon devices have dominated the semiconductor industry. Easy manufacturing techniques and constant improvements led to a well-developed ecosystem of devices ranging in capabilities from the latest digital processors to image sensors to power switching devices. Recently, an interest in wide bandgap combinations such as silicon carbide (SiC) and gallium nitride (GaN) have led to new devices with capabilities that exceed the performance of their silicon counterparts [1]. Researchers at the University of Arkansas have been working with SiC power devices as they were introduced and reached maturity [2]–[4]. Recent developments in SiC have allowed for more intensive integrated circuit (IC) design prospects. Combining the SiC power devices along with the SiC IC parts is a logical extension of the technology.

Theoretical performance benefits from changing a power device from a silicon technology to a silicon carbide device have turned into tangible benefits. The properties of silicon carbide allow for useful Schottky diodes at higher breakdown voltages than silicon, and power MOSFETs exist at 600 V ratings and above with lower on-resistance and lower capacitance than their silicon counterparts. The large bandgap provides new opportunities, as silicon carbide demonstrates capabilities at a temperature range beyond the reach of traditional silicon devices. With this performance extension made possible by silicon carbide power devices, control electronics are required that can operate in the same demanding environment. The most fundamental control circuit for a power device such as a MOSFET is a gate driver, a device that takes a logic-level signal and amplifies it to a strength capable of rapidly turning on and off a power MOSFET. Without a suitable gate driver circuit to safely control the power MOSFET, several of the performance benefits of silicon carbide devices cannot be realized. A

high-temperature capable gate drive circuit is an essential circuit for enabling high-temperature capable power electronics.

Organization

This dissertation is organized with overview material first. Chapter 2 describes general wide bandgap phenomena with a focus on silicon carbide. Direct applications of silicon carbide to power devices and integrated circuits are covered, with an emphasis on the silicon carbide integrated circuit process used in this work. Following that, in chapter 3, will be an overview of power electronics used in this work. General power electronics concepts are presented with particular attention paid to implications for gate drive electronics.

Chapters 4-6 deal with the design of the integrated silicon carbide gate driver. As the chronological development of the driver spans two manufacturing runs and interconnected development of separate features, a purely time-oriented organization does not make sense. Instead, the core driver development is described in chapter 4. The output transistors for the gate driver were optimized, and the description of that process is given in chapter 5. An evaluation of design-for-test (DFT) issues related to the design, and active decisions to improve testability in the design are provided in chapter 6, as well as the evaluation of the DFT system as implemented and the yield tests that resulted.

The in-circuit testing of the gate driver is given in chapter 7. This chapter covers the test configurations used, and the evaluation of the test setup. Multiple samples are tested under various configurations to successfully demonstrate the functionality of the gate driver. Finally, conclusions and future work are given in chapter 8. An appendix contains the code that was written in the course of completing this work.

2 Silicon Carbide Overview

Silicon has dominated the semiconductor industry so thoroughly that all other semiconductors are judged relative to silicon to determine their merits and disadvantages. The most defining differentiation in new semiconductor materials has been the bandgap energy, E_G. Silicon carbide and gallium nitride exhibit much higher bandgap energies of 3.26 eV and 3.39 eV, respectively, compared to the nominal 1.12 eV of silicon. These so-called wide bandgap semiconductors have several different intrinsic properties that are enumerated in Table 2.1 [5]. Silicon carbide has multiple crystal polytypes, such as 3C, 4H, and 6H, but the primary focus of recent research and production devices has been on 4H SiC.

Table 2.1 Material Properties of Silicon Carbide and Gallium Nitride compared to Silicon

Parameter	Silicon	4H-SiC	GaN
Bandgap Energy	1.12 eV	3.26 eV	3.39 eV
Critical Electric Field	0.23	2.2	3.3
Relative Permittivity	11.8	9.7	9.0
Electron Mobility	1400	950	800/1700(1)
Intrinsic Carrier Concentration n_i at 300K	1e10	8e-9	2e-10
Thermal Conductance	1.5	3.8	1.3 (epi)
Baliga FOM $\varepsilon_r * \mu_n * E_C^3$	2.01x10e20	9.81x10e22	5.50x10e23 (2DEG)

(1) Significant difference between bulk and 2-dimension electron gas

Considering the impacts of Table 2.1 on typical power devices, the first choice is to compare a theoretically optimal transistor from one material to another. For the purposes of power devices, it is useful to compare the ideal on-state resistance. Unipolar power devices, such as power MOSFETs and Schottky diodes, contain a drift region, shown in Fig. 2.1, that is responsible for blocking voltage when the device is not conducting. The length of this drift region dictates the voltage that the device is capable of blocking, and lower resistance is desirable for reducing device conduction loss. The specific resistance of the ideal drift region can

be characterized by Eq. (2.1)[6]. The ideal resistance of the drift region $R_{on\text{-}ideal}$ is given as a function of the breakdown voltage BV, which increases with the square of the required voltage. In the denominator, the relative permittivity ε_r, electron mobility μ_n, and critical electric field E_C combine as material-specific constants that influence the device performance. The denominator products (2.2), also known as Baliga's figure of merit for power devices [6] is a useful number for comparing semiconductor materials. The lower mobility of silicon carbide in Table 2.1 is easily compensated by the large difference in critical electric field, especially as E_C is cubed in the equation. Comparing the Baliga figure of merit to silicon, silicon carbide has a score 491 times better, and gallium nitride scores 2,735 times better than the silicon baseline. This demonstrates the first potential improvement of wide bandgap devices: lower on-resistance at a given breakdown voltage.

S G S
N+ P P N+
N-DRIFT REGION
DRAIN
(a)

ANODE
N-DRIFT REGION
CATHODE
(b)

Fig. 2.1. Unipolar power device structures: vertical power MOSFET structure (a) and power Schottky diode structure (b).

$$R_{on-ideal} = \frac{4BV^2}{\varepsilon_R \mu_n E_C^3} \tag{2.1}$$

$$Baliga\ FOM = \varepsilon_R \mu_n E_C^3 \tag{2.2}$$

The higher Baliga figure of merit score indicates that an ideal SiC device at a given breakdown voltage can be physically smaller in both area and drift region thickness than an equivalent silicon device. A smaller die area requires a smaller effective gate area, which reduces the amount of gate-source capacitance necessary with all other effects being equal. However, the increased critical electric field E_C decreases the required drift region thickness, bringing the gate physically closer to the drain contact and increasing the gate-drain capacitance of a vertical power MOSFET. This changes the inherent ratio of gate-source to gate-drain capacitance, which has implications that will be described further in chapter 3.

The other promise of wide bandgap materials is the potential for high-temperature operation. The intrinsic carrier concentration of a semiconductor should be lower than the doped carrier concentration for proper operation. As the absolute temperature, T, rises in a semiconductor material, the intrinsic carrier concentration n_i will also increase. If the intrinsic carrier concentration approaches or exceeds the doped carrier concentration, the semiconductor will not function as intended. The properties determining the intrinsic carrier concentration in a semiconductor are the bandgap energy of the material E_G and the state densities N_C and N_V. Equation (2.3) [6] shows the relationship between these material properties, absolute temperature T, and Boltzmann's constant (1.38×10^{-23} J K^{-1}). The state density in SiC is 44% of silicon, but the major difference in intrinsic carrier concentration over temperature comes from the higher bandgap voltage of silicon carbide.

$$n_i = \sqrt{N_C N_V} e^{-\frac{E_G}{2kT}} \tag{2.3}$$

In practice, standard silicon parts operate up to 125 ℃, with enhanced power devices operating up to 175 ℃. High temperature silicon-on-insulator (SOI) extends the useful range of silicon integrated circuits up to 250 ℃, with research pushing the boundaries up to 450 ℃ [12]. Silicon carbide has a much greater operational temperature range in commercial and experimental devices, with commercially available power transistors rated for 210 ℃ [13], and experimental power converters operating at 250 ℃ [14]. On the integrated circuit front, silicon carbide has been shown to function above 400 ℃ [15], [16] and even higher than 500 ℃. The higher temperature capability fuels forays into new circuit locations and situations, such as reduced cooling requirements, engine combustion chambers, and down-hole exploration [4]. The unique capabilities of silicon carbide are opening new realms for power electronics and electronics in general.

Another benefit demonstrated in Table 2.1 is the difference in thermal conductivity. Silicon carbide has a higher coefficient of thermal conductivity that is useful for power devices. This higher coefficient value allows for more efficient heat transfer from internal structures to external heat dissipation structures such as heat sinks, cooling base plates, or radiators. More efficient heat transfer also results in more uniform die temperatures, resulting in hot spots that are less pronounced compared to silicon.

SiC in Power Electronics

In particular, silicon carbide power devices lend themselves to high voltage operation. Commercially available transistors range in maximum drain-source voltages of 600 V [13] to 1700 V [17]. Schottky barrier diodes are available in similar voltage ranges, and provide benefits over standard silicon PN junction diodes. Higher voltage devices exist, but have not reached widespread commercial distribution. Packaged single devices have current ratings up to 160 A

[18]. This range of voltages and currents lends itself to several high power applications traditionally performed by silicon insulated gate bipolar transistors (IGBT). Such applications include AC motor drives, power inverters [3], and electric car chargers [19]. The benefits of SiC transistors over typical IGBT applications are reduced device capacitance, instant turn-off instead of IGBT tail current, and better thermal conductivity. These benefits result in lower switching energy and lower energy loss.

High Temperature Gate Driver Research

Prior work towards high temperature gate drive began with an SOI drive circuit with external SiC JFET for the output drive [9]. Later SOI efforts led to a driver and module rated for 225 ℃ [20]. Integrated SiC gate drive electronics began with an attempt at a CMOS driver in 6-H SiC [21]. Further work in a Cree NFET process realized the first SiC integrated gate driver [15], [22]. Another effort with an integrated NPN process resulted in a circuit capable of driving low-voltage drive SJT devices [23]. With the advent of a high temperature silicon carbide CMOS process [24], a SiC CMOS gate driver for SiC power MOSFETs is a logical extension.

SiC Integrated Circuits

Integrated circuits are another domain where interest in high-temperature electronics has fueled investigations into silicon carbide. Initial efforts began with the 6H polytype of silicon carbide due to availability, and small device count CMOS circuits were developed and tested [21], [25]–[27]. Further work has been done with NFET enhancement and depletion mode processes [15], [16], [28], JFETs [29], [30], and BJTs [23]. Of particular interest to power electronics design are gate driver circuits. With power devices capable of operating at elevated temperatures, companion circuitry such as transistor drivers are necessary for a complete high-temperature implementation. A seemingly ideal scenario of placing a SiC gate driver on the same

die as a power MOSFET was explored in [15], but the limitations imposed by the vertical power MOSFET processing made that an elusive goal.

Overall, SiC integrated circuits face limitations not experienced by contemporary silicon devices. While the electron mobility of SiC is lower than Si, the effective transistor transconductance is lower than expected [31]. One issue with doping SiC circuits is that high-energy ion implantation is the primary doping method [31]. The doping was performed before any growth of gate oxide to allow for the high temperature annealing. This eliminates the self-aligned gate benefits used in silicon, increasing the difficulty of manufacturing. The gate must also overlap the source and drain terminals more than a self-aligned process, resulting in higher gate-source and gate-drain capacitance. These disadvantages make lateral MOSFETs lower in performance compared to silicon counterparts.

Recently, Raytheon Systems Limited in the UK has developed a CMOS integrated circuit process using 4H SiC [24], [32]. This process consists of an N-substrate for PFET devices with a P-type well for the NFETs. This is opposite of traditional silicon designs where the PFET parts are contained inside an N-type well. The minimum MOSFET channel length of the process is 1.2 μm, with two polysilicon layers for MOSFET gates and floating polysilicon-insulator-polysilicon capacitors. A single metal layer is available for interconnect, made of a refractory metal. A cross-section of a CMOS inverter is shown in Fig. 2.2.

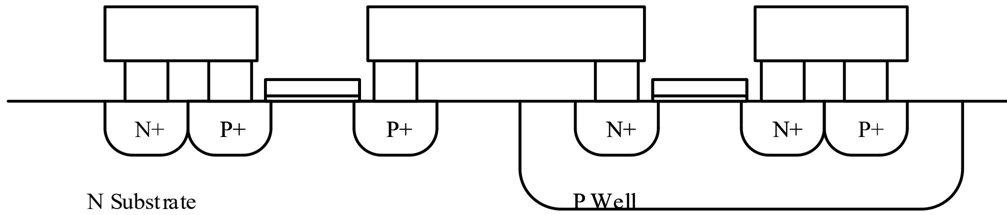


Fig. 2.2. Example cross-section of a CMOS inverter in the Raytheon HiTSiC process

The University of Arkansas began work with the HiTSiC® process in late 2012 through the National Science Foundation Building Innovation Capacity program. Upon receiving samples from experimental runs, initial efforts began with measurements of transistor test structures with the intent of developing simulation models for the NFET and PFET devices. A process development kit (PDK) was created for Cadence Virtuoso, and simulation models were created in BSIM3v3 to represent the expected behavior of the first run. Devices were measured over temperature to create separate fixed-temperature models at 25 ℃, 100 ℃, 200 ℃, and 275 ℃. Layout verification tests in Calibre were included, and consisted of design rule checking (DRC), layout versus schematic (LVS), and parasitic extraction (PEX). The first tapeout was completed in August 2013, with wafers returning in February 2014. Models were updated to measured data from run 1, and upgraded to BSIM4 to allow greater flexibility adapting a silicon model to silicon carbide. Additional model corners were generated based on burn-in behavior observed, which resulted in two model values at 200 °C and 300 °C: a "fresh" version representing a new device, and the normal version exposed to several hours of bias at temperatures above 200 °C. Layout rules were updated following some yield issues, and minor updates in the PDK were made in time for the second tapeout in September 2014. Second run wafers returned in April 2015 for testing. Gate driver circuits were fabricated on both runs, with the final version on run 2 comprising the majority of the design information presented.

Summary

On paper, silicon carbide has many compelling properties that lead to a favorable comparison to silicon. A theoretical SiC power device has many benefits over a similar silicon device, though SiC manufacturing techniques have yet to approach the level of refinement in silicon systems. With demonstrated potential from early devices, work continues towards advancing power silicon carbide further. Recent developments in SiC integrated circuit process design have produced several methods for high-temperature electronics capable of operation from 400 °C to 600 °C with good performance. These recent developments have opened the door for new technologies and circuit implementations at high temperatures that are not possible with silicon. Power and integrated electronics now can push into new regimes where electronics dared not venture before.

3 Power Electronics Switching

The application of silicon carbide power transistors and diodes requires a thorough understanding of the uses and limitations of transistors in power electronics applications. This chapter examines several switching concepts that are relevant to application circuits that use SiC power MOSFETs and gate drivers. Many concepts are universal at the broadest level of design, while a few are specific to MOSFETs or silicon carbide.

Examination of switching topologies

While not an exhaustive survey of switch-mode power converters, many power topologies can be derived from a circuit with two switches connected together, as in Fig. 3.1. Depending on the circuit functionality required, Q1 or Q2 may be omitted, though D1 and D2 functionally exist as discrete devices or as internal body diodes from the MOSFET. Frequently this structure is implemented in non-isolated converters such as buck, boost, and half-bridges. Multiple copies can be instantiated to form an H-bridge or a 3-phase inverter. Blocking is required in one voltage orientation, and diode conduction in the other direction is acceptable or necessary for proper operation. This diode operation can be augmented by activating the switch, and this is called synchronous switching. Switch mode converters add an inductive component to the V_{SW} node functioning as an energy storage device or a filter, with the on-cycle of the switching charging the inductor through the active switch, and discharging with a path through a diode in the off-cycle. This switching configuration is commonly called a clamped inductive load, and has a specific set of system dynamics. Specifically, the inductor charges when one of the switches turns on, and turning off the switch will result in the output voltage rising to forward-bias the opposing diode.

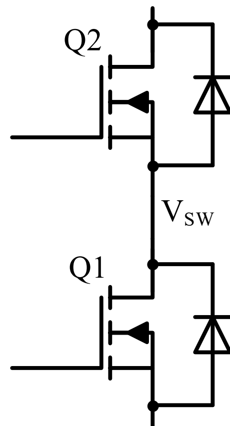


Fig. 3.1. Fundamental switch orientation that can be expanded into various power electronics circuits by the addition of passive components.

Consider an idealized switch that does not turn off instantly, but instead rapidly and steadily increases the effective resistance across the two terminals. A MOSFET representation of such a circuit is shown in Fig. 3.2. At turn-off of the power device, the inductor forces the voltage of the output node to forward-bias the high-side diode. The current during the voltage swing is the full inductor current, and only decreases once the diode is forward-biased. The voltage across the switch rises to V_{BUS} while the current is at I_L, and the current decreases from I_L to 0 A while the voltage is at V_{BUS}. At turn-on, the exact process is reversed, with the switch increasing in current from 0 to I_L while at V_{BUS}, and then the voltage decreasing once the current in the transistor reaches I_L. This type of switching involves high peak power, is frequently called “hard switching”, and is the sole type of switching examined here. Other techniques include zero voltage switching (ZVS) and zero current switching (ZCS) where resonant circuits are used to reduce the voltage or current across the switch to reduce switching losses.

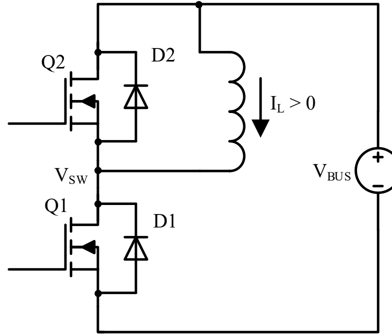


Fig. 3.2. Clamped-inductive load circuit, with optional synchronous MOSFET Q2

MOSFET Clamped-inductive load switching

Further exploration of the clamped inductive load requires replacing the theoretical switches with real components. For this discussion, a MOSFET is used to explore the deeper complexities of a hard-switched clamped inductive load. Sample idealized waveforms are shown in Fig. 3.4. There are four distinct phases of turn-on in a clamped inductive load switching event, beginning with the initial change in MOSFET gate voltage. The gate voltage increases steadily from the resting "off" voltage to the point where the MOSFET begins to turn on at V_{TH}. No other changes in the system are observed during this phase. Once the gate voltage rises above V_{TH}, the MOSFET begins to conduct current. While current starts flowing through the MOSFET, the drain voltage will remain constant. As long as the MOSFET drain current is less than the total inductor current, the remaining inductor current must have an additional path. In this circuit, the alternate path is through the diode in parallel with the inductor. As long as current is flowing through the diode, the drain voltage will stay at a diode drop above V_{BUS}. The gate voltage will continue to increase consistently in this phase until the drain current is equal to the inductor current. Once the MOSFET has turned on enough to turn off the diode, the drain voltage can start changing, which leads to the next state.

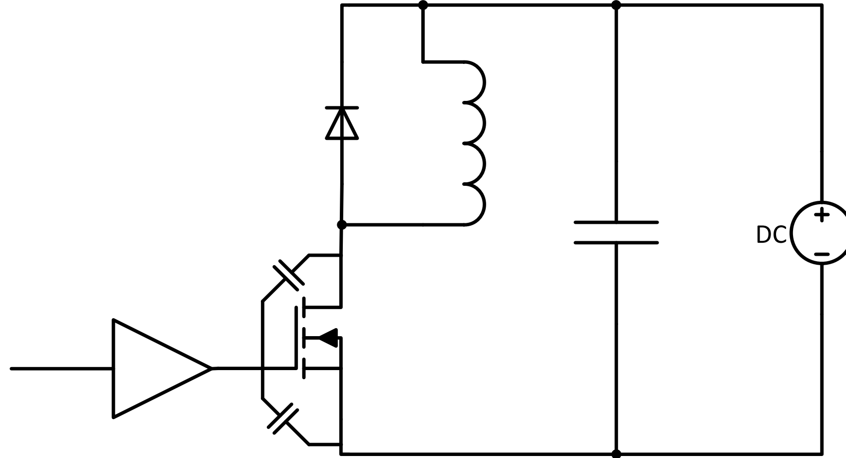


Fig. 3.3. Clamped inductive load schematic showing location of C_{GD} and C_{GS}

Once the drain current is greater than the inductor current, the drain voltage falls. The falling drain voltage provides negative feedback from the output to the gate of the MOSFET through C_{GD}. In typical MOSFETs, $C_{GD} \ll C_{GS}$, which results in a gate charge depletion that is proportional to the *dv/dt* of V_{DS} and C_{GD}. This current flow steals the gate driver charging current from charging C_{GS}, and slows the rise in gate voltage. This flatter section in the V_{GS} waveform is commonly referred to as the Miller plateau. The Miller plateau lasts until V_{DS} reaches its steady-state value of approximately 0 V.

The final switching phase consists of the MOSFET gate voltage rising to the intended drive voltage after crossing the Miller plateau. This change in gate voltage is necessary to decrease the $R_{DS(ON)}$ of the MOSFET to acceptable levels. The drain voltage may decrease slightly, but not at a rate that produces significant gate current. The end of this phase is marked by the gate voltage reaching $V_{GS(ON)}$.

MOSFET turn-off in a clamped-inductive system is functionally the same as turn-on, except the phases occur in reverse order. The gate driver will begin to decrease the gate voltage to the point where the MOSFET $R_{DS(ON)}$ begins to increase significantly. This results in the drain voltage rising as the inductor forces V_{DS} higher to accept the total current. As the drain voltage

rises, current flows into the gate through C_{GD}, resulting in the negative feedback that causes the Miller plateau. The drain voltage rises until it reaches a voltage high enough to turn on the diode. The drain current then decreases to 0 A as the gate voltage decreases below V_{TH}, and then the gate voltage decreases to $V_{GS(OFF)}$.

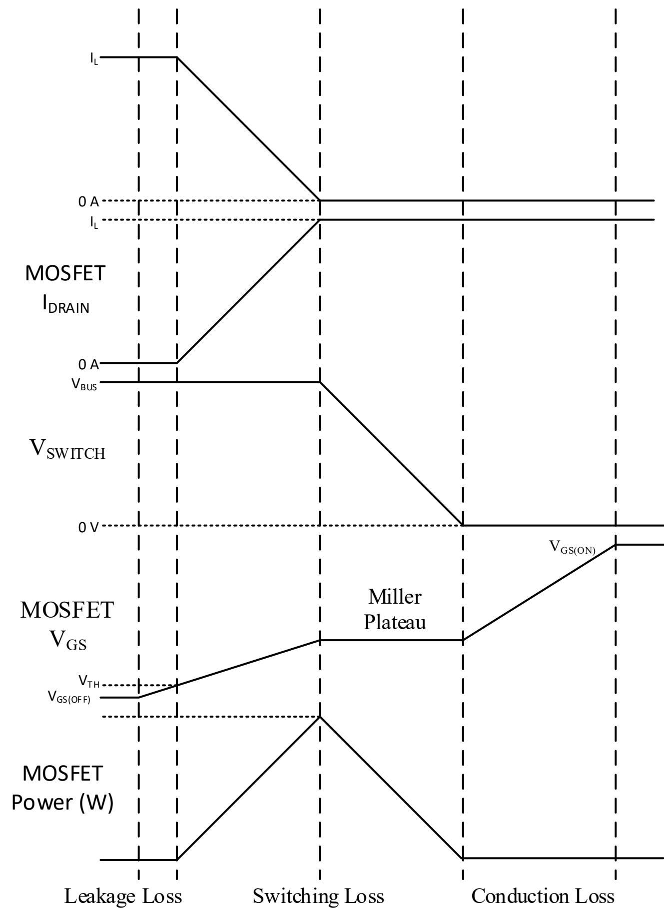


Fig. 3.4. Example MOSFET switching waveforms for a clamped-inductive load turn-on.

It is important to note that the clamped-inductive switching configuration develops several peak conditions during switching that do not exist to the same extent in other common configurations. The clamped-inductive load guarantees that the MOSFET will experience maximum *di/dt*, maximum *dV/dt*, and maximum instantaneous power in a single controlled switching event. While several performance maximums will occur, the system will constrain

peak MOSFET drain currents, and the total energy dissipated is a function of the switching speed.

Consider three other switching loads illustrated in Fig. 3.5: a resistive load, a capacitive load, and an uncharged inductive load. For the simple resistive load (a), the drain voltage and drain current both begin to change simultaneously. Peak currents, as well as drain *dV/dt* are determined by how fast the MOSFET turns on and the value of R_{LOAD}. Adding a capacitor in parallel with the MOSFET, as in (b), will slow the drain fall-time and limit the peak dV/dt of the drain voltage. The drain current contains no inherent limiting mechanism, resulting in an unconstrained peak drain current that is dependent on the MOSFET switching speed and the size of the capacitance. The same magnitude of drain current as a clamped-inductive circuit may be observed during switching if the load capacitance is large enough. The total energy dissipated by the power MOSFET in this scenario is directly proportional to the capacitance of the load, and may exceed the safe operating area of the device. An uncharged inductor will result in a small decrease in peak drain dV/dt since the drain voltage will begin to change immediately after V_{GS} crosses $V_{TH(ON)}$ instead of after the drain is charged to the load current. With no load current, the total switching energy dissipated will be less than the clamped inductive load case.

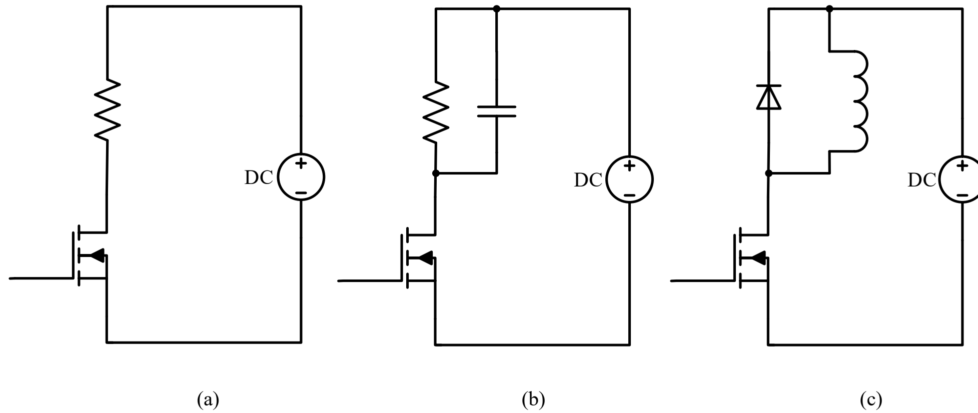


Fig. 3.5. Various load configurations for a single low-side switch. (a) shows a non-inductive resistive load, (b) shows a capacitance in parallel with the load, and (c) shows an inductive load with flyback diode

The drive strength of the gate driver is a constant influence in the switching speed. In all but the Miller plateau, the MOSFET gate behaves similar to a capacitor. As a result, the total switching time is primarily determined by the gate driver drive strength. Circuit conditions can also influence the switching rates. The magnitude of the inductor current determines how long it takes to transfer the current from the clamping diode to the power MOSFET. The magnitude of the drain voltage determines how long it takes for the MOSFET to cross the Miller plateau.

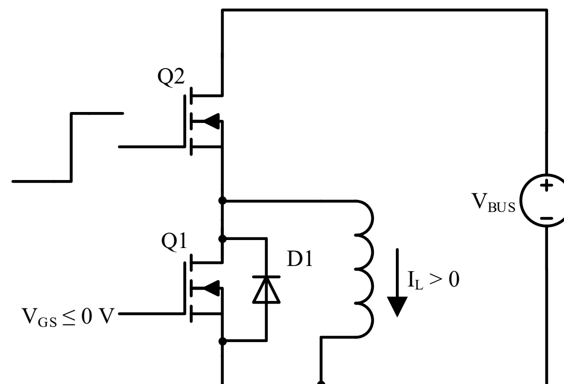


Fig. 3.6. Clamped inductive load configuration with charged inductor. D1 starts in conduction, and Q2 switches on, resulting in a fast rising V_{DS} on Q1. This rapid drain voltage rise has the potential to cause the V_{GS} of Q1 to rise above $V_{TH(on)}$

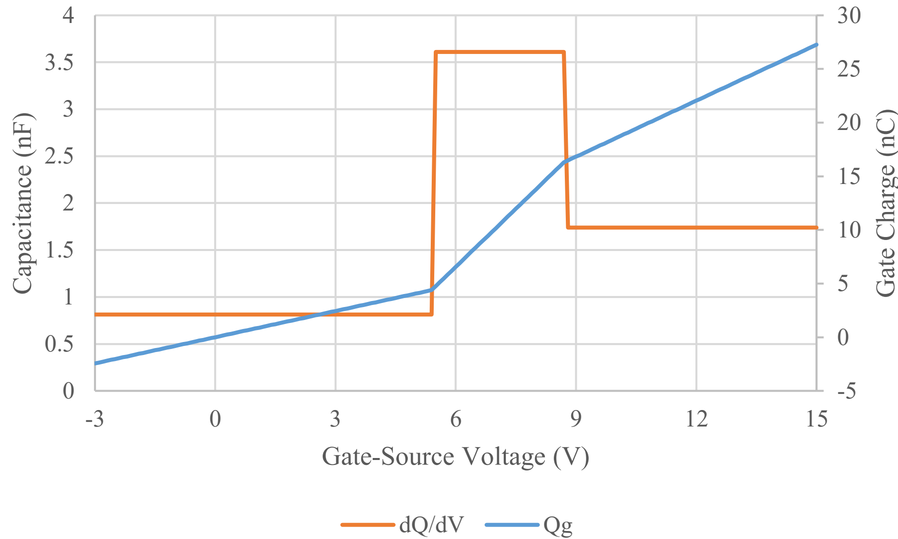


Fig. 3.7. Gate charge and related capacitance parameters for a Cree C3M0065090J MOSFET. The gate charge parameters are linear in three separate regions as indicated by the *dQ/dV* line. Input capacitance is 813 pF in off region with a V_{DS} of 400 V and an I_{DS} of 20 A at 25 ℃.[33]

While the normal switching criteria manage the switching characteristics of the MOSFET, another scenario must be considered with some topologies containing multiple transistors. Considering the system in Fig. 3.6, when the high-side MOSFET Q2 switches on, the drain voltage will experience a rapid voltage swing. Internal capacitances and resistances are shown in Fig. 3.8. This rapidly rising voltage will induce a current in the parasitic MOSFET capacitances C_{GS} and C_{GD}. Given a gate charge plot of Q_G vs. V_{GS} from a transistor datasheet, the estimated switching condition required to turn on Q1, assuming no gate drive, can be calculated. Fig. 3.7 shows the gate charge (Q_G) versus V_{GS}, and the resulting gate capacitance from the datasheet of a Cree C3M0065090J.

The simplest gate drive stability criteria is determined by assuming that the transistor gate drive provides only the initial DC operating point, and has no impact on the gate voltage during switching. For the purposes of evaluation, C_{GD} is set to a constant value, and the drain voltage is assumed to transition at a fixed *dV/dt*. This allows an estimation of the current flowing into the

gate node (3.1). A common datasheet graph for a power MOSFET includes a gate voltage plot versus gate charge, which allows for an identification of how much gate charge is required to transition from V_{GS} = 0 V to $V_{GS(th)}$. Using this value and our gate current from (3.1), the maximum time a gate current can be sustained before the gate voltage limit is reached is given in (3.2). This "safe" time can then be fed back to identify a safe V_{BUS} (3.3), and then substituting in (3.2) and (3.1) results in equation (3.4). This shows a relationship between the gate charge required to turn on the device with C_{GD}, and the assumed *dV/dt* term disappears.

$$I_{CGD} = C_{GD} \times \frac{dV_{DS}}{dt} \tag{3.1}$$

$$\frac{Q_{G(ON)}}{I_G} = t_{safe} \tag{3.2}$$

$$t_{safe} \times \frac{dV_{DS}}{dt} = V_{BUS_{safe}} \tag{3.3}$$

$$V_{BUS_{safe}} = \frac{dV_{DS}}{dt}\left(\frac{Q_{G(ON)}}{C_{GD}\frac{dV_{DS}}{dt}}\right) = \frac{Q_{G(ON)}}{C_{GD}} \tag{3.4}$$

Further refining of (3.4) can be accomplished by simplifying the $Q_{G(ON)}$ term into a capacitance multiplied by the turn-on voltage. While this approximation requires a linear relationship between charge and voltage, examining the gate charge plot in Fig. 3.7 shows a linear Q/V behavior below the Miller plateau. This refinement allows replacement of $Q_{G(ON)}$ in (3.5), and substitution into the previous safe voltage yields a safe bus voltage expressed as a ratio of gate capacitances and proportional to $V_{GS(ON)}$ in (3.6). Perhaps more intuitively, this shows the circuit operating as a capacitive voltage divider. For systems where V_{GS} is driven to a voltage below 0 V when off, the safe bus voltage can be found with equation (3.7).

$$Q_{G(ON)} = C_{GS}V_{GS(on)} \quad (3.5)$$

$$V_{BUS_{safe}} = \frac{C_{GS}}{C_{GD}}V_{GS(on)} \quad (3.6)$$

$$V_{BUS_{safe}} = \frac{C_{GS}}{C_{GD}}\left(V_{GS(on)} - V_{GS(off)}\right) \quad (3.7)$$

Table 3.1 Device properties from selected high voltage transistors

Part Number	**STF12N120K5[34]**	**APT37M100L[35]**	**IPW90R120C3[36]**	**CMF10120D[37]**	**C2M0160120D[38]**	**C3M0065090J[33]**	**SCT2160KE[39]**	**GA10JT12-247[40]**	**Units**
Material	Si	Si	Si	SiC	SiC	SiC	SiC	SiC	
$V_{DS(MAX)}$	1200	1000	900	1200	1200	900	1200	1200	V
$R_{DS(on)}$, 25 ℃	690	330	120	160	160	65	160	100	mΩ
$I_{D(MAX)}$, 25 ℃	12	37	36	24	19	35	22	25	A
$V_{TH(ON)}$, 25 ℃	3.0	3.0	2.5	2.4	2.0	1.8	1.6	3.2	V
C_{ISS}	1,370	9,835	6,800	928	525	660	1200	1403	pF
C_{RSS} @ 80% $V_{DS(MAX)}$	3.5	28	8	7	4	4	7	28	pF
C_{OSS}	110	150	60	63	47	60	45	28	pF
Q_G	44.2	305	270	90.8	34	30	62	55	nC
Max V_{BUS}, Eq. (3.6)	1174 98%	1054 105%	2125 236%	318 27%	263 22%	297 33 %	274 23%	150 13%	V
$V_{GS(OFF)}$ for 100% in (3.7)	-0.07	0	0	-6.65	-7.14	-3.65	-5.40	-20.8	V

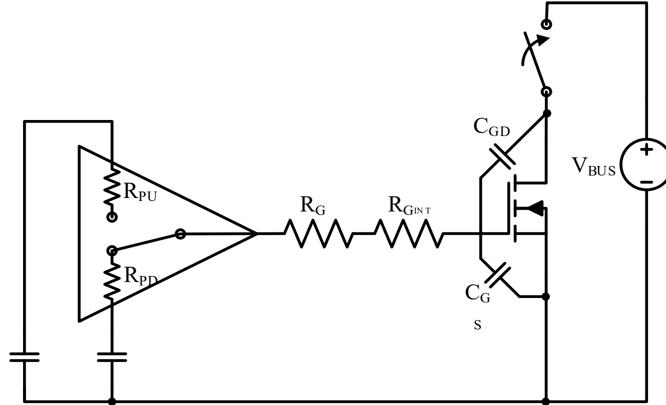


Fig. 3.8. Parasitic turn-on sub-circuit with explicit internal capacitances and resistances illustrated for the purpose of deriving system behavior under a switching transient.

At this point, an examination of existing commercial devices provides insight on the problems of silicon carbide. Table 3.1 shows several transistors with V_{DS} ratings between 900 V and 1200 V, and the calculated $V_{BUS(safe)}$. The silicon devices have safe bus voltages near or exceeding the rated $V_{DS(MAX)}$, while the silicon carbide devices fall below 1/3 of $V_{DS(MAX)}$. The primary culprit for this stability discrepancy is the C_{GS}/C_{GD} ratio, which is significantly lower in silicon carbide devices. To complicate matters further, SiC MOSFETs have lower minimum $V_{GS(th)}$ values than comparable silicon devices, which reduces the $V_{BUS(MAX)}$ margin even further. The typical method to counter the poor $V_{BUS(MAX)}$ figure used in SiC is to drive the gate voltage negative when the device should be off. While equations (3.6) and (3.7) provide a switching stability criteria, the scenario is excessively conservative and unrealistic: MOSFET gates are never disconnected intentionally. In order to improve the accuracy of the testing scenario, a gate-source impedance is added, consisting of a resistor and inductor in series as shown in Fig. 3.9. This circuit approximates the switching voltage on the drain with the expression from equation (3.1).

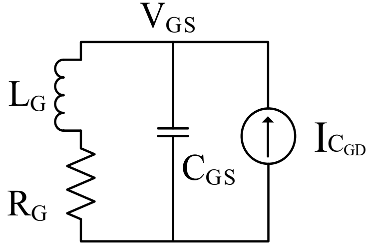


Fig. 3.9. Simplified gate drive network for identifying safe operating conditions.

The factor R_G is the sum of all resistances in the gate drive loop shown in Fig. 3.8, and L_G is the sum of all parasitic layout inductances in the gate drive loop. The system was solved in the Laplace domain, and converted back into the time domain for further analysis. The term *b* is calculated separately in equation (3.8), and determines whether the response of the system is under-damped, critically damped, or over-damped. The other term, τ_L, is the RL time constant formed by R_G in series with L_G. The critically-damped and over-damped responses are given by equation (3.10), and occur when *b* is non-negative. The under-damped response occurs when *b* is negative, giving complex values as seen in (3.11).

$$b = \frac{R_G^2}{L_G^2} - \frac{4}{L_G C_{GS}} \tag{3.8}$$

$$\tau_L = \frac{L_G}{R_G} \tag{3.9}$$

$$V_{GS}(t) = u(t)\left(I_{CGD}R_G - \left(\frac{I_{CGD}}{C_{GS}}\right)\left(\frac{\tau_L^{-1} + \sqrt{b}}{\sqrt{b}\left(\tau_L^{-1} - \sqrt{b}\right)}\right)e^{-\frac{1}{2}\left(\tau_L^{-1} - \sqrt{b}\right)t} + \left(\frac{I_{CGD}}{C_{GS}}\right)\left(\frac{\tau_L^{-1} - \sqrt{b}}{\sqrt{b}\left(\tau_L^{-1} + \sqrt{b}\right)}\right)e^{-\frac{1}{2}\left(\tau_L^{-1} + \sqrt{b}\right)t}\right), b \geq 0 \tag{3.10}$$

$$V_{GS}(t) = u(t)\Bigg(I_{CGD}R_G - \frac{2I_{CGD}}{C_{GS}\sqrt{-b}}e^{-\frac{1}{2}\tau_L^{-1}t}\cos\left(\frac{1}{2}\sqrt{-b}t + arctan\left(\frac{\sqrt{-b}}{2\tau_L^{-1}} - \frac{\tau_L^{-1}}{2\sqrt{-b}}\right)\right)\Bigg), b < 0 \tag{3.11}$$

Expected system values are taken for a C3M0065090D device, and then applied to the above equations. The peak gate voltage is shown in Fig. 3.10 for various combinations of R_G and L_G. This peak gate voltage should remain below the threshold voltage of the MOSFET for safe operation, which is 1.6 V typical at 100 °C[33]. The C_{DS} of the device is 4 pF, and an example *dV/dt* of 50 V/ns is chosen for the figure, giving an I_{CGD} of 200 mA. Equations (3.10) and (3.11) are evaluated over time from 0 to 12 ns, representing a 600 V equivalent bus. The gate capacitance C_{GS} used was 660 pF, and values of L_G and R_G are near the expected range of values for an optimized layout to a poorly optimized layout. The effect of increased inductance is a higher peak gate voltage. If the system is underdamped, which appears above the circled line, higher inductance will result in a faster rate of increase in peak voltage. The other effect is that as τ_L increases, the peak voltage decreases further from the steady-state voltage of $I_{CGD}R_G$. The smaller the value of R_G, the lower the peak gate voltage, and the lower the risk of accidental turn-on. However, the ability to reduce R_G is limited by the internal gate resistance of the power MOSFET, which is typically 4.7 Ω for the C3M0065090D. Another important observation about equations (3.10) and (3.11) is that the voltage is entirely proportional to I_{CGD}, or the drain voltage *dV/dt*. By reducing the *dV/dt* of the switching device, the safety margin can be extended.

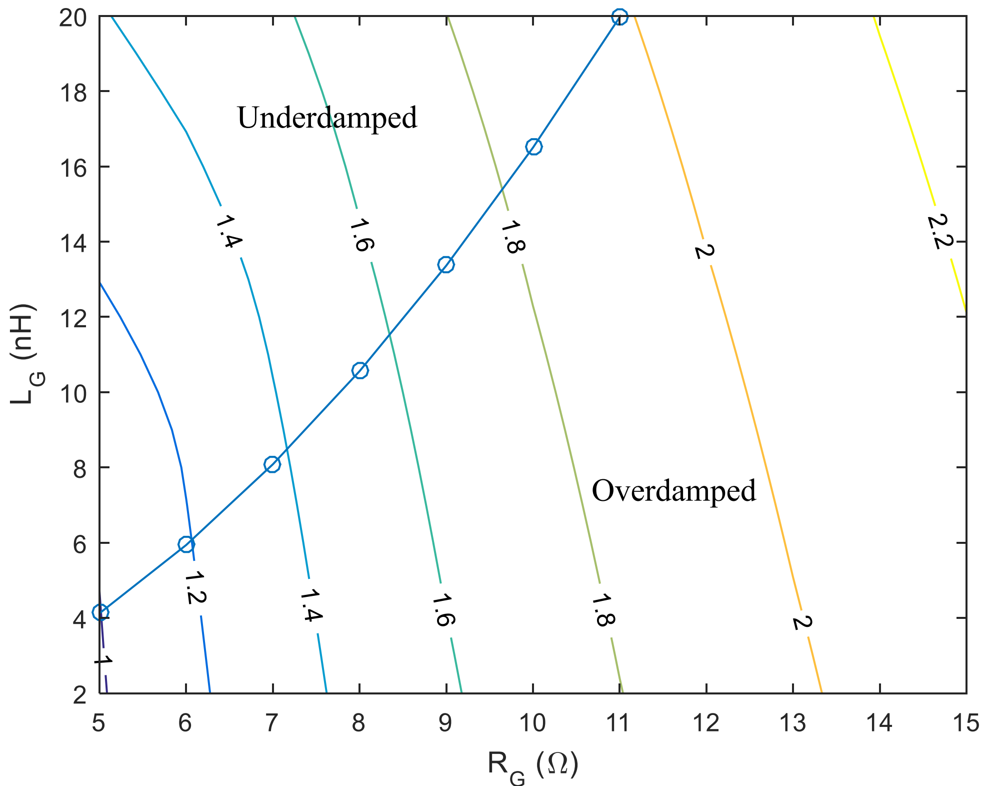


Fig. 3.10. Example peak gate voltage after 10 ns time with an I_{CGD} of 200 mA. The C_{GS} used is 660 pF, which represents the gate voltage behavior switching 600 V in 12 ns. The critically damped boundary line is marked with circular markers.

If the gate control system fails to secure the gate voltage below $V_{GS(TH)}$ while the transistor is commanded off, then the transistor will begin to turn on. If the transistor is weakly activated, then the effect may simply be a slower turn-on time with additional power dissipated through the expected off device. However, if the gate voltage reaches a high enough level, the drain voltage will begin to fall. With the falling drain voltage, the gate voltage can begin to fall as the current injection from C_{GD} has changed polarities. With a similar weakness in the pull-up device, an oscillation can result in the destruction of the power devices. As the previous statement suggests, this effect is also possible with a single device during a switching event, if the MOSFET can switch fast enough to overpower the gate driver in the Miller plateau.

There are two other parasitic inductance effects that play a significant role in electronics switching. Referencing Fig. 3.11, the common source inductance L_S is a significant hazard to safe switching. This parasitic inductance causes a negative feedback effect on the gate drive signal when the MOSFET current is switching. Turn-on and turn-off *di/dt* can easily exceed 1 A/ns in silicon carbide power MOSFETs, which will manifest in the circuit as 1 V for every nH contained in L_S. Physically, if the gate drive current loop intersects with the power loop outside of the MOSFET source bonding pads, the common wire bond inductance alone will likely exceed 2 nH, and large through-hole packages may have inductances up to 10 nH.

The other effect is a ringing effect across the switch when the switch is turned off. A series LC circuit is formed through the C_{DS} of the power MOSFET, and the sum of the inductances in the power path (L_{BUS} + L_{DIODE} + L_D + L_S). During the Miller plateau portion of turn-off, the series inductances L_{BUS} + L_D + L_S are charged to the primary inductor current, I_L. Once the end of the Miller plateau is reached with V_{SW} above V_{BUS}, the parasitic inductances are charged without a significant resistive element from Q1. The charged inductors discharge into the C_{DS} of Q1, which nominally forms an underdamped response. This underdamped response occurs on the V_{DS} of Q1, which may result in a voltage overshoot beyond the voltage ratings of the device and possibly damage Q1. Reduction of the series inductances will affect the frequency of the oscillation, but will not significantly change the magnitude of the overshoot [41].

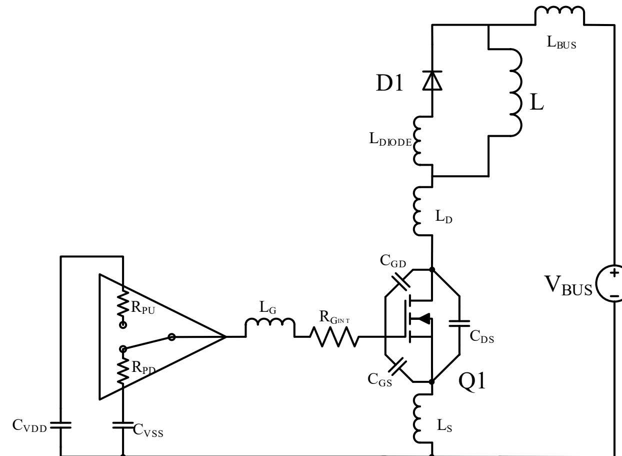


Fig. 3.11. Taking the internal resistance and capacitances of devices from Fig. 3.8, parasitic interconnect inductances are added as lumped elements to complete the system model. Series inductances have been combined to simplify analysis.

Gate Drive Techniques

Fundamentally, a gate driver is the last stage of circuitry between the control logic determining the state of the power MOSFET and the gate of the power MOSFET. With such a broad definition, many circuits fall under this category. The simplest method for driving a power MOSFET is the absence of a separate drive circuit, or a direct drive from a logic signal. The typical drive from a logic gate or microcontroller has low current drive and low voltage magnitude. The low current, frequently near 20 mA, means that only the smallest power transistors will switch quickly. The low drive voltage of 5.0 V or less causes issues with typical power MOSFETs that require 10 to 20 V to achieve the advertised performance. So-called "logic level" power MOSFETs exist, with acceptable $R_{DS(on)}$ at typical voltages for a digital system.

Gate drivers built out of discrete components are perhaps the simplest circuit to drive a MOSFET gate with higher current. A complementary pair of MOSFETs or even bipolar junction transistors (BJT) can be combined to build a push-pull pair of transistors to increase the drive strength significantly. Further simplification of the implementation involves using a gate driver IC, which combines one or more levels of drive signal magnification. Additional features may be

incorporated, including level translation, level shifting, and circuit protection. Indeed, a gate driver is not a unique circuit by itself, having hundreds if not thousands of implementations in commercially available catalog parts. For the purpose of discussion, a gate driver will be abstractly represented using the "buffer" schematic symbol, with implicit connections to a gate driver power supply.

Power supply connections are one source of design differentiation. The simplest configuration consists of a single positive supply sharing the negative terminal with the source of the power MOSFET. Specific implementations of the power supply source are outside the scope of this discussion. A single-ended supply allows the gate to be driven to V_{DD} and 0 V. An additional negative supply permits the gate voltage to be held negative, and increases the inherent noise immunity of the transistor by increasing the noise margin required to unintentionally turn on the transistor. However, multiple supplies increase the complexity and parasitic layout inductance of the design. Additional supplies of 3 or more use multiple drive level circuits for silicon carbide superjunction transistors (SJT) [42].

Another source of design surrounding a power MOSFET is the connection of the gate driver to the power MOSFET. The simplest connection is a direct connection from the output of the gate driver to the gate of the power MOSFET, equivalent to Fig. 3.12 (a) with R_G set to 0 Ω. This has the lowest additional component count and potentially the highest drive strength to the MOSFET gate. By omitting any current-limiting circuitry, the only limit to gate current is the inherent resistance of the gate driver. This arrangement requires verification that the driver will provide sufficient yet not excessive drive over all operating conditions. Excessive drive strength can violate a maximum drain *dV/dt* specification, or increase drain voltage overshoot beyond

device tolerances. Fundamentally, the performance of the system depends on the gate driver characteristics over temperature and operating conditions

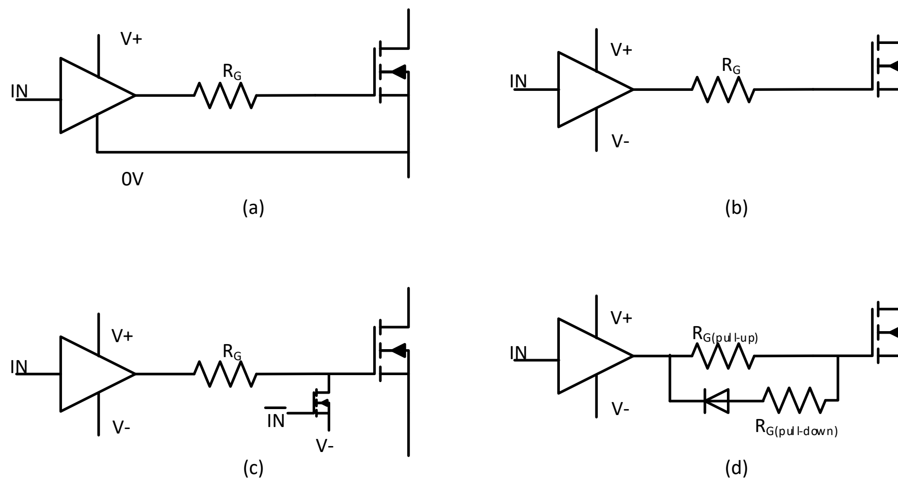


Fig. 3.12. Various gate driver connection strategies. (a) shows a single supply connection, (b) shows a dual-rail power supply, (c) shows an active Miller clamp, and (d) shows a diode – resistor network to provide additional drive strength at turn-off.

The next variant in drive connections involves the addition of a single resistor in series between the gate driver output and the MOSFET gate. This resistor accomplishes a consistent reduction in drive strength for both the turn-on and turn-off condition. The low additional complexity is more stable over temperature, and provides consistent gate current limiting. The current limiting reduces impact from variation in gate driver strength due to part-part variation or temperature effects. The resistor value determines the turn-on and turn-off speed, and can be adjusted as needed to tune the final system. However, with a single resistor, the turn-off drive strength cannot be isolated, resulting in lower noise immunity.

The limitation of a single drive strength for turn-on and turn-off can be mitigated by adding a diode in parallel with the drive resistor, Fig. 3.12 (d). This additional path provides a method for short-circuiting the gate resistor, typically allowing for a lower impedance turn-off path. A resistor in series with the diode is one variant that allows for decreased turn-off

impedance without reducing it to near 0 Ω. The main cost is complexity and increased part count, but results in higher noise immunity.

Another way of attacking the noise immunity of the power devices is with active Miller clamping, Fig. 3.12 (c). Fundamentally, this technique uses an additional transistor to short the gate directly to the negative drive voltage or the source of the power transistor. This clamping transistor can be activated independently or with the same gate driver signal. The clamp counteracts any current flowing into the gate node through the gate-drain parasitic capacitance C_{GD}, such as during turn-off, or when another device in series activates. The low impedance path provided by the transistor directly bypasses any gate drive resistor. This allows for lower drive strength during turn-on and turn-off, while ensuring a strong clamping force while holding the transistor off.

Current-mode drivers are another strategy, using a charged inductor as the primary current source for charging and discharging the gate capacitance. Two basic modes of operation exist depending on the magnitude of energy and current stored in the drive inductance. Resonant drivers match the energy stored in the inductor closely to the energy required to fully drive the MOSFET gate to a satisfactory voltage [43]–[45]. Current source drivers have a large inductor that is charged to the desired charge or discharge current. This constant current does not change significantly during turn-on or turn-off, resulting in a controlled switching event [46]. Both mechanisms provide good current drive near the Miller plateau and offer mechanisms for recovering the energy stored in the power MOSFET gate capacitance. These large inductances require charging before use, and potentially discharging before turning off the power MOSFET, resulting in increased control latency and more restrictions on minimum on-time for the power switch.

Another method suggests switching the gate driver power supply connections [47]. The single gate driver power supply can be reconnected to provide a negative V_{GS} at the same magnitude of $V_{GS(ON)}$. In a SiC MOSFET, the steady-state condition would result in a V_{GS} below the absolute minimum V_{GS}. The internal gate resistance R_G, internal gate capacitance C_{GS}, and expected current from C_{GD} during a switching event are used to determine a safe exposure time for the gate to be exposed to the negative voltage, maintaining the internal gate voltage within datasheet limits. This method shows potential power savings, but the complexity and risk of such a circuit prevent an easily integrated implementation.

Silicon carbide challenges

Wide bandgap devices offer several tangible benefits over silicon devices. The combination of low $R_{DS(ON)}$ and low device capacitances result in faster switching speeds than comparable silicon parts. The end result is that lower C_{GS} allows the transistor to turn on and off faster, and a smaller C_{DS} takes less time to fully charge and discharge.

Minor concerns in silicon MOSFET designs such as short path inductances and gate drive stability are elevated to a high priority in silicon carbide modules. The switching speed of SiC brings high drain voltage and drain current derivatives. This high current change (*dI/dt*) and voltage change (*dV/dt*) produce significant impulses in the circuit that adversely affect circuit performance and stability. High *dV/dt* couples directly through capacitances, generating large current impulses over the duration of the switching event. Peak *dV/dt* of 10 V/ns or higher, commonly achieved in SiC power designs, generates a current of 10 mA through a parasitic capacitance of 1 pF. Such current magnitudes approach the drive limits of digital and analog electronics, potentially causing data corruption or unintended operation. Similarly, fast current switching of 1 A/ns or greater results in a voltage of 1 V across an inductance of only 1 nH.

Source inductances that are common to the gate drive loop and the drain-source current loop have a negative feedback effect on the gate, forcing the gate drive voltage down based on the parasitic inductance.

A typical problem with existing silicon carbide transistors is the ratio of C_{GS} to C_{GD} is relatively low compared to silicon devices, as seen in Table 3.1. This poor ratio combined with the high voltages typically used for bus voltages results in potentially dangerous operation in the Miller plateau. The standard SiC mitigation strategy involves using a negative turn-off supply to improve the inherent stability of the system. However, the maximum negative gate voltage that SiC MOSFETs tolerate is less than the maximum positive voltage, and not sufficiently low to provide an insurmountable obstacle to unintended turn-on, as also shown in Table 3.1.

Silicon carbide MOSFETs also have different DC characteristics compared to silicon devices, as well as different voltage ranges. Typical high-voltage silicon power MOSFETs have a wide gate voltage range, often +/- 20 V. Full drive strength occurs at 10 V or less, and there are large V_{GS} regions where the device is fully on or off. The gate voltage margins for silicon carbide are minimal, requiring full voltage for the minimum $R_{DS(on)}$, and a minimum voltage of -5 to -10 V. This increases the difficulty of driving the gate of a SiC MOSFET by removing the safety margins.

Summary

The drive strength of a gate driver needs to be controlled in order to guarantee system performance and stability. The drive strength controls essential system switching parameters, as it directly affects the switching speed of the power devices. As a result, adjusting the drive strength allows for direct control of the output *di/dt* and *dV/dt*, as well as overshoot and ringing.

Even systems with no explicit circuitry to control drive strength, such as a direct connection of a gate driver to the power MOSFET, require consistent system performance.

4 Gate Driver Design

While the decision to design a gate driver was made, the target system implementation was left broad: design a high temperature, CMOS gate driver for SiC power MOSFETs with a variable drive strength. A more detailed set of core specifications were developed in order facilitate design decisions, based on prior experiences using silicon gate drivers driving SiC power MOSFETs.

Design Goals and Specifications

A list of gate driver design goals is presented in Table 4.1. These began with the target output voltage from the gate driver. With a nominal 0 – 15 V range for normal CMOS logic, the 0 – 15 V range was selected as the target output voltage. This voltage underdrives 1200 V SiC MOSFETs, but provides a satisfactory drive level for 900 V devices. While a larger voltage range was desirable, increasing this beyond the SiC fabrication process limits was beyond the scope of this dissertation.

A major concern early on was the drive strength capability of the output transistors. Gate driver current ratings are given in terms of peak short-circuit current, essentially identifying the highest possible sinking and sourcing current possible. This peak current rating is rarely, if ever observed in practice as the pull-up and pull-down networks gradually ramp up to full strength. Nonetheless, a target of 4 A was set to match common gate driver IC ratings. Due to the large observed mismatch between PFET and NFET drive strength and the lack of a negative $V_{GS(off)}$, an asymmetric pull-down strength of 8 A was proposed. In order to provide a more realistic drive strength condition, a target transition time of 30 ns was given for the gate driver output when loaded.

A switching frequency of 500 kHz was selected as a target minimum frequency for power electronics. This higher frequency allows the faster switching speed of silicon carbide MOSFETs to be realized, and was a reasonable frequency goal considering the target rise and fall times for the output. Finally, the operating temperature range was set to the same range as the simulation models available, which was 25 – 300 ℃.

Table 4.1 Initial Design Goals

Specification	Value	Unit
Output Voltage Range	0 – 15	V
Peak Gate Drive Current	+4 / -8	A
Output Rise/Fall with load	<30	ns
Target Switching Frequency	>500	kHz
Operating Temperature Range	25 – 300	℃

Basic Topology

Starting with the Raytheon HiTSiC process, a topology analysis was required before delving into further design decisions. The most important decision in a gate driver is the output stage, the transistors that directly drive the gate of the power MOSFET. At the time of the design, three devices were considered as possible output drivers: the standard PFET, the standard NFET, and a simplistic NPN. Based on the limitations of the devices, Fig. 4.1 shows the possible configurations of output devices.

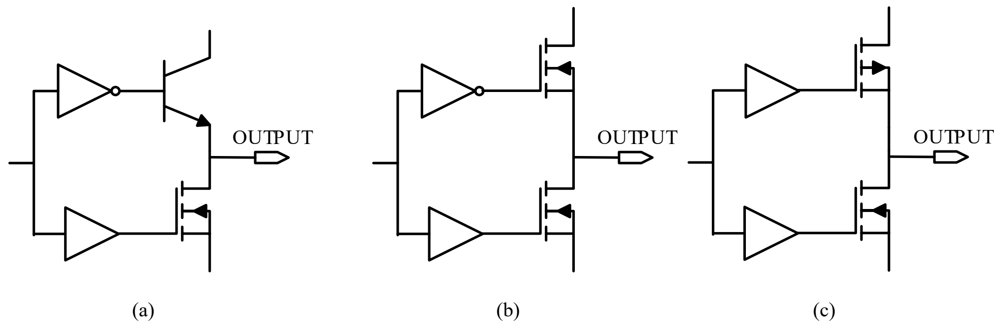


Fig. 4.1 Gate driver output topologies possible with the Raytheon HiTSiC process. Part (a) shows a NPN pull-up device, (b) shows a NFET pull-up device, and (c) shows a PFET pull-up device.

The NPN available at the time required the collector to be connected to VDD. This limited the NPN to a pull-up device only, as the collector must be connected to the output for a pull-down configuration. The topology using an NPN device, Fig. 4.1 (a), has a high current density but has multiple disadvantages. Since the NPN is a pull-up device, it is operating as a common-collector or emitter follower configuration. This inherently limits the strong pull-up voltage to be one diode drop below VDD. Additionally, this does not eliminate any reliance on a PFET, as a PFET is required for optimal drive of the NPN. Combined with the lack of data available for the NPN, the bipolar transistor was avoided.

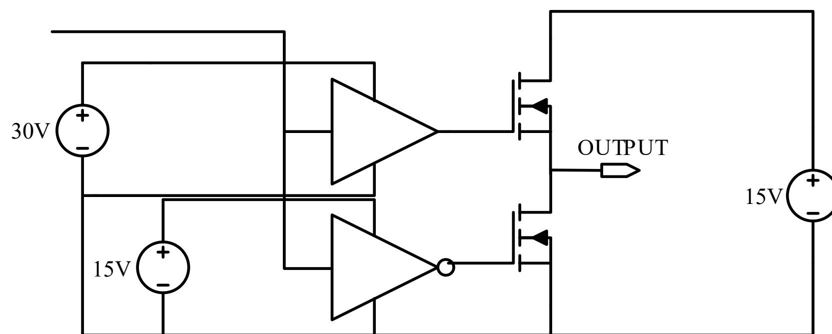


Fig. 4.2. Voltage domains required for a NFET totem pole output.

Another option was to exclusively use NFET devices in the output stage (Fig. 4.1 (b)) in a totem pole orientation, shown in greater detail in Fig. 4.2. With the initial measured die, the drive strength of the NFET was between 5 and 10 times greater than the drive strength of the

PFET. A NFET pull-up device would require substantially less area to achieve the same drive strength as a PFET. In order to drive the pull-up NFET with a 15 V signal, significant machinations were required due to the P-well structure of the Raytheon HiTSiC process. The gate oxide had a rating of 15 V, while the internal diode junctions had a rating of 30 V. This allows both the pull-up and pull-down NFET to float relative to the substrate voltage. The N-type substrate must be the highest voltage in the system, meaning that in order to drive the pull-up NFET with a +15 V gate bias when driven high, the highest voltage would need to be 30 V. The pull-down device would be driven by a level-shifter in the 0 – 15 V domain. The required dual supplies also implied that standard 15 V circuits would have a power supply between the 15 V and 30 V rail, resulting in additional interfacing complications.

The final option was to drive the power transistor with a CMOS output, Fig. 4.1 (c). While the PFET lacked the drive strength of the NFET, a 0 – 15 V output swing was realized with minimal complexity. The pull-up drive strength did not decrease near the positive voltage rail, nor did it require an additional stacked 15 V supply in order to achieve acceptable drive strength. The various devices available are shown in Table 4.2, with the complexity and drive strength qualitatively estimated. Considering the drive complexity as a risk factor, the CMOS output topology provided good drive characteristics with the lowest risk, and was selected for the gate driver design.

Table 4.2. Summary of Drive Methods

	NFET	**PFET**	**NPN**
Pull-down strength	Good	Bad	Moderate
Pull-up strength	Good	Good	Good
Pull-down drive complexity	Low	High	Moderate
Pull-up drive complexity	High	Low	Moderate

Power Module Integration

When using a power module, existing techniques for high temperature operation isolate the sensitive control electronics from high temperatures found near power electronics. Modules with internal drivers must limit the internal temperature of the module to the temperature limit of the driver, which is often lower than the maximum temperature of the power devices.

A major impact in driving large power MOSFETs is the parasitic inductances present in the gate drive loop. Typical power module characterization involves accurate characterization of parasitic layout properties inside the power module, but external bus wiring and gate drive connections are neglected. As a result, large series inductances in the gate drive loop alter the anticipated stability of the switching system. These inductances inhibit peak switching speeds and limit performance. Within the constraints of a wire-bonded power module, the most direct wire-bond connection is directly from the gate pad of the MOSFET to the output pad of the gate driver, and the source of the MOSFET to the ground of the gate driver. The next best option for gate connection has a pad for the gate inside the module that is wire-bonded to both the gate driver and power MOSFET gate. However, there are many concerns upon moving the gate driver into a power module:

- A common substrate is Direct Bond Copper (DBC): thick copper on alumina or aluminum nitride (AlN). The thick copper has poor trace resolution which also leads to poor connectivity options. Typical applications limit routing to a single layer with no holes. This limits the number and complexity of I/O for a system in a power module;
- No soldermask and poor routability limit ability to place supporting components into module;

- Coarse layout features are preferred to reduce the required precision in DBC etching. However, fine traces are typical in chip on board (COB) applications.

Considering the limitations that exist when moving the gate driver into the power module, a set of goals formed around making the system integration feasible and easier:

- Minimize routing complexity of system,
- Minimize number of must-connect pads, and
- Minimize number of external components required.

A direct connection of the gate driver to the power MOSFET eliminates many options of drive strength control discussed in chapter 3. On-chip resistors are not consistent over temperature or process variation for a fixed drive current limit, and they limit peak system performance. Previously mentioned gate drive techniques function by modifying an ideal driver by adding external restrictions to the drive current. With an integrated driver, the strength can be modulated internally to create the same effect without requiring external components to restrict the system. The method for modulating the drive strength must be chosen to complement the capabilities of the process, or system losses may result in failure to meet performance metrics. Placing devices in series wastes drive strength and depends on the ability to adequately drive a floating transistor. Output transistor calculations indicate low initial drive strength, and floating transistors cannot be driven well in the HiTSiC process.

Adjustable Drive Strength

In the process of optimizing the transistor dimensions described in chapter 5, it was observed that the output transistors should be subdivided into "slices" for adjustable drive strength performance. Further investigation was undertaken to determine what the advantages of

such a system would be. By separately enabling each transistor slice, several drive profiles are possible. Device slices may also be bypassed in the event of some failures, providing a level of redundancy.

Selectively enabling different numbers of output slices, shown in Fig. 4.3, results in an adjustable drive current output. Control logic passes the gate driver command signal through to enabled transistor slices, and disabled transistor slices are driven to a V_{GS} of 0 V. For the proposed system of eight transistor slices, a range of eight different drive strengths for pull-up and pull-down are available.

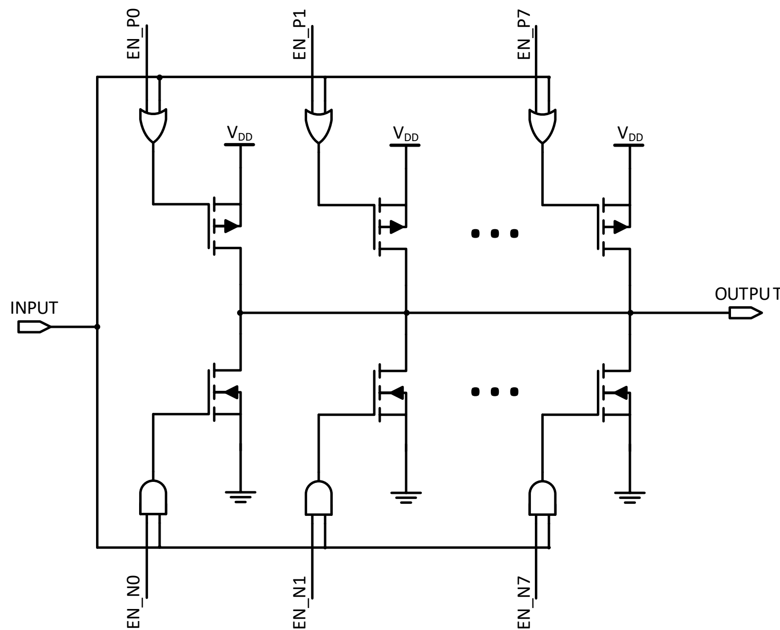


Fig. 4.3. Gate drive slice strength modulation schematic.

Two main constraints limited the number of slices implemented: control logic size and routing complexity. The size of the control logic for each transistor slice as implemented was significant. Additional slices reduced the available active area for the output transistors, an undesirable tradeoff. The other constraint was that as the number of transistor slices increased, the area required to distribute drive signals increased. This increase in signal routing area came at

a direct cost to active output transistor area. Recall that the HiTSiC process only has one level of metal for on-chip interconnect routing. Eight slices for each pull-up and pull-down provided an acceptable tradeoff of the available area with the desired system performance, and matched well with the transistor optimizations in Chapter 5. A major consideration was the available area for logic gates, as each transistor slice contains many gates. Two rows of logic gates per transistor slice with an enhanced power and ground routing results in a row height of 276 µm, for a maximum of 18 slices without any spare area. The number of PFET slices and NFET slices were set to an equal number, and this resulted in 16 transistor slices.

With equally sized slices, the number of drive strengths available was equal to the number of slices. With each slice using the same layout, the electrical difference between slices was expected to be minimal, which resulted in a linear relationship between the number of slices enabled and the drive strength of the output. Additional drive strength variation range could be realized by making the slices different sizes. For example, a binary weighting of slice sizing would increase the drive strength dynamic range significantly, and could be combined with a linearly weighted segment. Replacing one of the existing slices with a ½ strength and a ¼ strength slice while maintaining the rest at the original strength would increase both the dynamic range and resolution by a factor of four. However, the cost in terms of control electronics area and potential reliability concerns from additional design complexity resulted in uniform slice sizes for the final run.

The slice enable configuration is stored in a shift register, and provides a logic signal to AND with the gate driver input signal. If a slice's configuration register is storing a '1', then the slice will be driven with the gate driver input signal, otherwise the slice will be held to a V_{GS} of 0 V. A discrete register for each slice was chosen to allow greater flexibility with selecting active

slices. An alternative to storing individual slice programming is to store the number of enabled slices in a register, and decode the number of transistor slices to activate. This method saves area due to needing fewer bits of storage (3 versus 8). However, a defective transistor slice cannot be selectively deactivated in a decoded drive strength setting, resulting in single slice failure that ruins the entire device. The slice yield issues previously experienced in the first HiTSiC run were near 50%, so individual slice registers were selected to provide increased yield at a cost of more die area.

An additional feature was designed to provide an alternative to the adjustable drive strength: an override signal that ignores the state of the configuration registers and enables all the output transistor slices to be driven by the gate driver input. This shift register override pin (SR_OR) can also be used to create a second drive strength level that can be accessed by asserting the shift register override pin.

With the SR_OR pin, three modes of operation are available to the power module designer. The simplest mode is a full drive strength gate driver that can be realized by tying the SR_OR pin to V_{DD} and completely ignoring the configuration shift register. This mode is useful when the number of pins available inside a power module is limited, or the configuration register cannot be programmed in-system. The outputs in this case will be driven with all transistor slices enabled. While many pins must still be connected for proper operation, all but the gate driver input pin and the output pins can be tied to either V_{DD} or V_{SS}.

An intermediate complexity case involves two additional control pins. In this case, the shift register is controlled, and the SR_OR pin is tied to V_{SS}. This mode allows for the adjustable drive strength functionality, with independent control of the pull-up and pull-down drive strength. The independent control allows for the adjustment of the rising and falling switching

speed by activating only a few of the output transistor slices. The shift register directly controls which transistor slices are activated, so shifting new configurations through the shift register will immediately change the drive state of each transistor. Depending on the existing state and the new pattern, loading in new data may reduce or completely deactivate the drive to the power MOSFET. As a result, this mode offers limited support for dynamic drive strength switching. Instead, the drive strength should be changed at power-up to a specific value. This drive strength value could be previously measured during initial module characterization, or adjusted following a fault that required a system reset.

For full module control as well as the ability to change settings on the fly, a third input is needed for controlling the SR_OR pin. Using the SR_OR allows for a third control mode that adjusts the drive strength during a holding state, shown in Fig. 4.4. This modulation allows for the optimization of the turn-on and turn-off transitions, while allowing for maximum drive strength to hold the gate state to mitigate parasitic turn-on and turn-off.

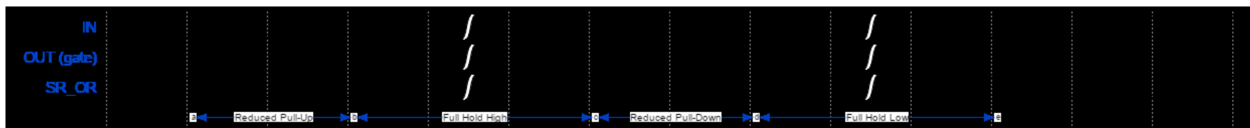


Fig. 4.4. Input signal timing for multiple drive strength switching.

With the ability to override the control shift register state, the hazard of eliminating or reducing drive strength while loading new drive strength values is eliminated. The SR_OR pin can be asserted, as shown in Fig. 4.5, while the new values are loaded into the shift register. When a configuration update is performed while the gate driver is holding a state, the impact to switching behavior can be minimized. The shift register can operate at a higher frequency than the gate driver. However, with 20 bits of data to load, the drive strength update will overlap multiple switching cycles of faster gate driver switching frequencies.

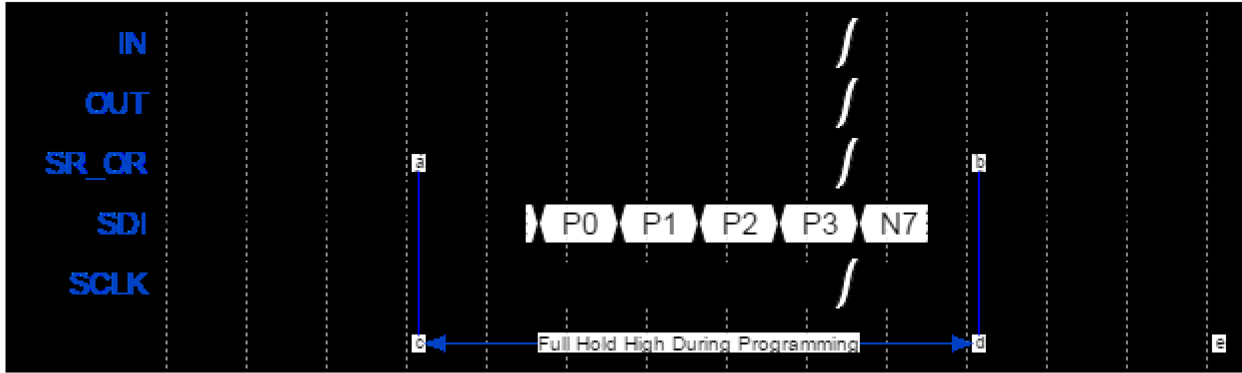


Fig. 4.5. Use of SR_OR pin to maintain drive functionality during drive strength programming.

Dynamic drive strength modulation allows for the drive strength to be adjusted as part of a control loop to maintain desirable power MOSFET switching properties while a module undergoes load variation, module aging, temperature changes, V_{BUS} swings or other changes that impact performance. Critical performance metrics, such as *dv/dt* or peak drain voltage overshoot/undershoot, can be monitored by a supervisory circuit. A maximum overshoot limit, or a target *dv/dt* can be set, and the drive strength directly influences each parameter.

Logic Implementation

With the system goals set, the digital logic was implemented to realize the gate driver functionality. A standard logic library was used to implement the digital logic. In addition to the mentioned drive slice logic, there is also a top-level control block responsible for converting the input pin values into control signals for all of the drive slices and the test mode logic. While the test mode logic is a significant portion of both the bit slices and the core logic, the test interface will be discussed in detail in a later chapter. The test mode uses a different set of input pins, with the normal pins intentionally left floating. This requires logic to select the desired input pin for each function, shown in Eqs. (4.1) and (4.2). The signal TM_EN is derived from the TVDD pin, and generates a logic ‘1’ when power is applied to TVDD during testing. In normal operation, TVDD is disconnected, resulting in a logic ‘0’ on the TM_EN line. The other logic functions

performed with the TM_EN signal are the enabling and disabling of features based on the test mode status. Eqs. (4.3) and (4.4) enable alternate functions toggled by the TEN pin, and Eq. (4.5) shows the SR_OR feature disabled by the TM_EN signal.

$$CLK_INT = CLK \cdot \overline{TM_EN} + TCLK \cdot TM_EN \tag{4.1}$$

$$SDI_INT = SDI \cdot \overline{TM_EN} + TDI \cdot TM_EN \tag{4.2}$$

$$READ_IN = TM_EN \cdot TEN \tag{4.3}$$

$$DRIVE_EN = \overline{TM_EN \cdot TEN} \tag{4.4}$$

$$SR_OR_INT = \overline{SR_OR \cdot \overline{TM_EN}} \tag{4.5}$$

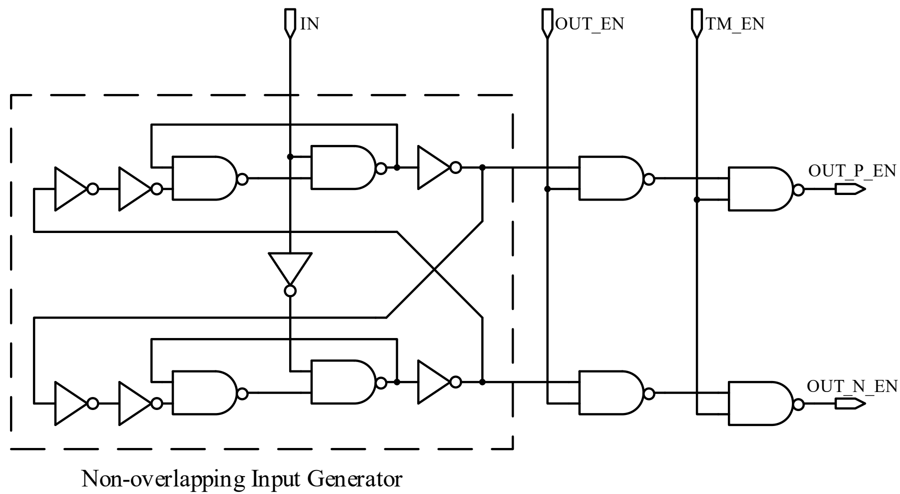


Fig. 4.6. Dead-time generation and input control logic.

Table 4.3 Simulated Control Block Timing

Temperature	Rising Delay	Rising Dead time	Falling Delay	Falling Dead time
25 ℃	56.9 ns	28.4 ns	54.0 ns	28.5 ns
100 ℃	41.8 ns	20.8 ns	39.7 ns	20.9 ns
200 ℃	40.2 ns	20.3 ns	38.2 ns	20.3 ns
300 ℃	53.8 ns	27.2 ns	51.6 ns	27.2 ns

The gate driver also needs protection from activating both the pull-up and pull-down transistors at the same time when changing output drive states. In order to mitigate this short-circuit condition, a delay must be inserted from the time that the pull-up/pull-down output is deactivated to when the pull-down/pull-up network is activated, also known as dead time. A block of logic was added to ensure that the pull-up and pull-down signals are non-overlapping with a minimum amount of delay. Fig. 4.6 shows the configuration of a standard non-overlapping signal generator, and Table 4.3 demonstrates the simulated timing of the control block over temperature. Representative waveforms of the timing block are shown in Fig. 4.7, with the input signal (red) translated into drive enable signals for the high side (purple) and low side (green).

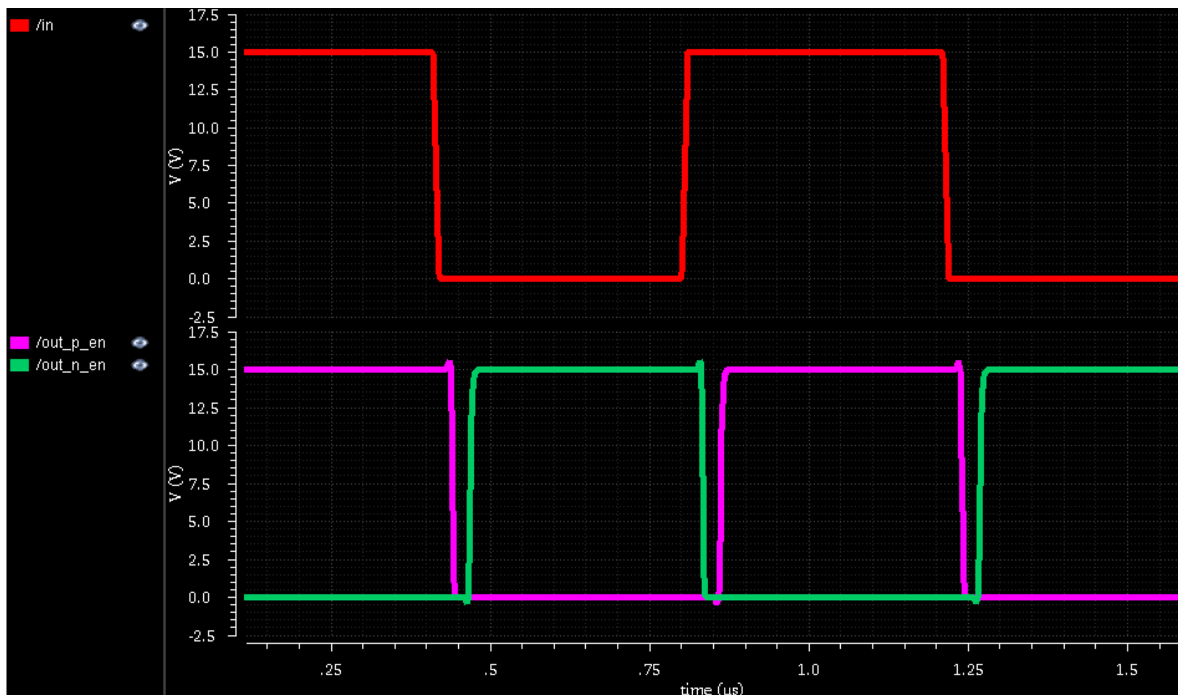


Fig. 4.7. Dead time generation logic simulation at 25 °C. The input signal on top (red) is delayed and turned into separate active high signals for the PFET pull-up (magenta) and the NFET pull-down (green).

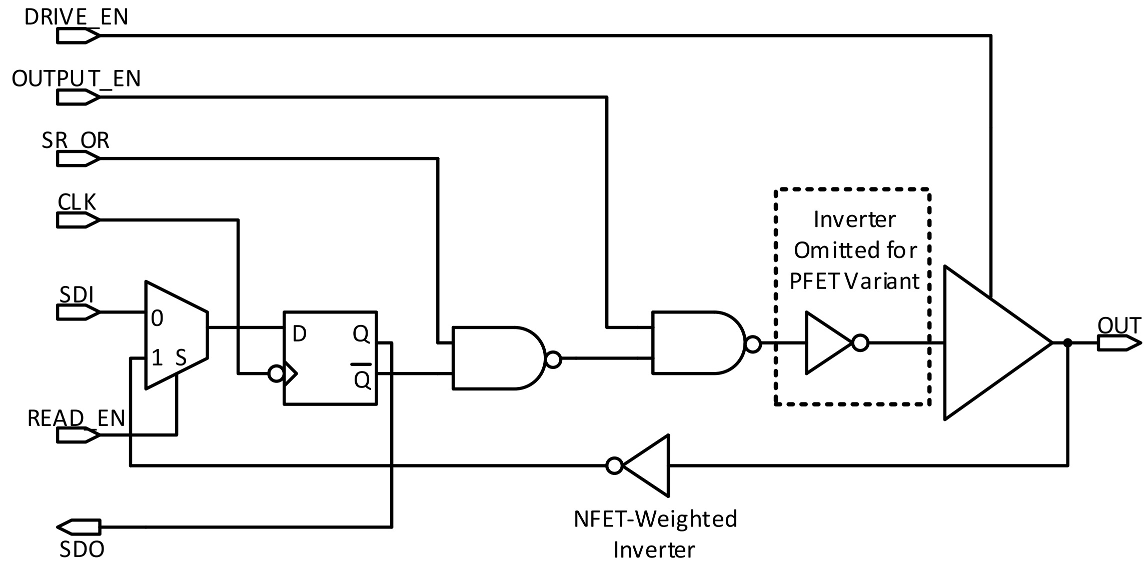


Fig. 4.8. Single transistor slice drive logic for a NFET. The output connects directly to the gate of the transistor slice.

The drive slice logic, shown in Fig. 4.8, is the other major logic block. The logic allows for many control signals to decide whether the slice transistor is driven to a high or low value, or left floating. The logical functions of the NFET and PFET control slices are shown in Eq. (4.6) and (4.7), respectively. The flip-flop takes serial data from the serial data input pin (SDI) and outputs it on the serial data output (SDO) pin. The floating transistor slice is important to the test mode operation. The control logic is largely identical between the PFET slice and the NFET slice, with the exception of one inverter before the output tristate buffer. This inverter was chosen to maintain the logic polarity between the NFET and PFET side to allow all control signals to have the same sense across the entire design. The other difference is that the weighted inverter is PFET-weighted for the PFET control slice. At the output of the drive slice is a tristate buffer that is capable of driving one transistor slice. In normal operation, these are locked into a high impedance state, as previously mentioned with Eq. (4.4).

$$OUT_{NFET} = OUTPUT_EN \cdot (\overline{SR_OR} + SDO) \tag{4.6}$$

$$OUT_{PFET} = \overline{OUTPUT_EN \cdot (\overline{SR_OR} + SDO)} \tag{4.7}$$

The tristate buffer must drive the gate of one of the power transistor slices, and also provide a high-impedance output state. There are multiple methods to make a tristate driver, but with drive strength concerns the tristate enable was moved to the low power input side as shown in Fig. 4.9. The buffer consists of separate drive paths for the pull-up and pull-down network. The gain stages begin immediately after the NAND gate, with transistor dimensions listed in Table 4.4. Inverter sizes were selected to be 3 to 5 times of the previous stage to avoid overloading each stage. The buffer was simulated over temperature driving both the final PFET slice as well as the separate NFET slice. In order to maximize design reuse, the same output driver was used for the PFET and NFET slices, as well as the blocks driving the FET array drains. Simulation results over temperature are shown in Table 4.5.

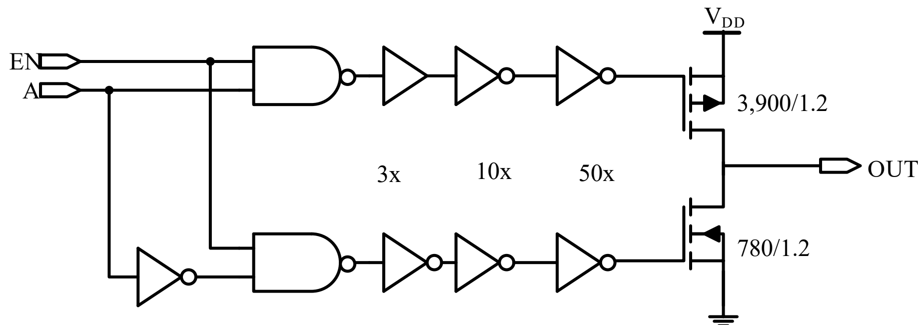


Fig. 4.9. Single transistor slice driver with tristate logic.

Table 4.4 Tri-State Buffer Transistor Dimensions

Stage	NFET W/L	PFET W/L
BUFx3 / INVx3	12.0 µm / 1.2 µm	60 µm / 1.2 µm
INVx10	40 µm / 1.2 µm	200 µm / 1.2 µm
INVx50	200 µm / 1.2 µm	1000 µm / 1.2 µm
FINAL	780 µm / 1.2 µm	3900 µm / 1.2 µm

Table 4.5 Transistor Slice Driver Simulation Results

Temperature	PFET Slice		NFET Slice		Rising Delay	Falling Delay
	Rise	Fall	Rise	Fall		
25 ℃	31.3 ns	23.4 ns	27.5 ns	19.0 ns	63.8 ns	52.7 ns
100 ℃	21.2 ns	19.7 ns	17.9 ns	15.4 ns	45.5 ns	40.6 ns
200 ℃	19.4 ns	18.7 ns	15.5 ns	13.7 ns	42.4 ns	38.0 ns
300 ℃	30.9 ns	19.8 ns	25.3 ns	15.0 ns	61.8 ns	47.8 ns

The final major block is another modification of the PFET and NFET control slice, and is used in test mode to set and observe the MOSFET drain voltage. Continuing the theme of block reuse, the existing logic found in the NFET and PFET control slice is used and augmented. The DRIVE_EN signal is the only dynamic input, controlling the tristate value of the driver, and eliminating the effect of the shift register when not in test mode. Combined with the control logic and the DRIVE_EN signal, this block is deactivated during normal operation, appearing as unused bits in the configuration shift register. An additional weighted inverter and shift register bit is added to provide two comparison thresholds for evaluation of the drain voltage. The modified schematic, shown in Fig. 4.10, is used for both the NFET and PFET drain nodes. Further discussion on the use of this block during test mode will be presented in Chapter 6.

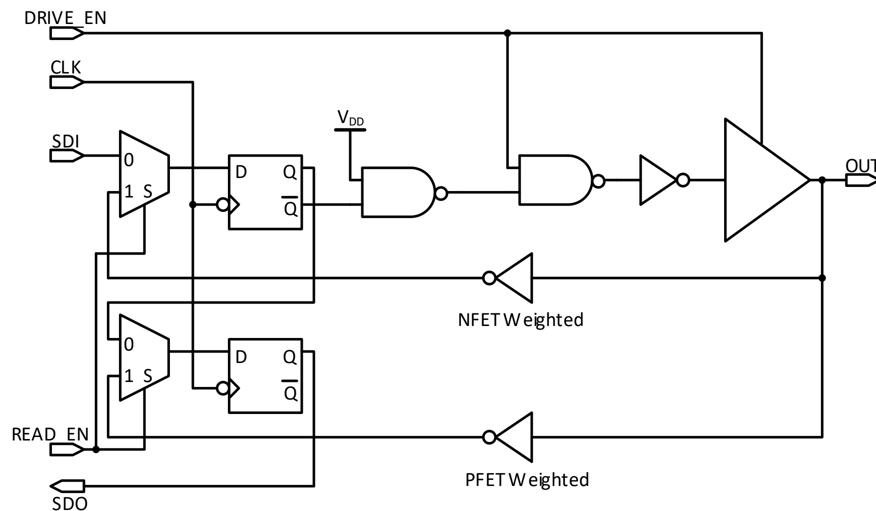


Fig. 4.10. Schematic for the test-mode drain voltage control and measurement.

Fig. 4.11. System block diagram

The control signals are tied together as shown in Fig. 4.11. All of the flip-flops from each control slice in the design are tied together in series into a configuration shift register that runs through the entire design. The shift register serves a dual purpose: holding the drive slice configuration, as well as providing a mechanism to output the test mode results. The shift register input begins with a multiplexor in the control block that selects between the D_IN and TDI pins. The shift register then flows through the pull-down control blocks, the test mode control blocks, and then through the pull-up control blocks. The last bit of the shift register in the pull-up block is connected to the SDO pad, where it can be probed. The input to each bit can either be the preceding bit in the register, or the test mode feedback from the same control block. This scan mode allows each transistor slice to be evaluated separately and efficiently. The bit usage for each bit is shown in Table 4.6.

Table 4.6 Configuration Register Functionality

Bit	**Normal**	**Test Mode**
0	PFET 0	V_{GS} PFET 0
1	PFET 1	V_{GS} PFET 1
2	PFET 2	V_{GS} PFET 2
3	PFET 3	V_{GS} PFET 3
4	PFET 4	V_{GS} PFET 4
5	PFET 5	V_{GS} PFET 5
6	PFET 6	V_{GS} PFET 6
7	PFET 7	V_{GS} PFET 7
8	Not used	V_{DS} PFET, High Threshold
9	Not used	V_{DS} PFET, Low Threshold
10	Not used	V_{DS} NFET, High Threshold
11	Not used	V_{DS} NFET, Low Threshold
12	NFET 0	V_{GS} NFET 0
13	NFET 1	V_{GS} NFET 1
14	NFET 2	V_{GS} NFET 2
15	NFET 3	V_{GS} NFET 3
16	NFET 4	V_{GS} NFET 4
17	NFET 5	V_{GS} NFET 5
18	NFET 6	V_{GS} NFET 6
19	NFET 7	V_{GS} NFET 7

The logic was arranged to reduce the number of gates and the number of stacked PFETs in a system. The PFETs, with the shared substrate, show a significant body effect, reducing their effectiveness in a stacked configuration. To exasperate the issue, the NFET to PFET strength ratio from earlier runs was 5:1, requiring five times more PFET gate width for equivalent drive strength. A common practice in static CMOS logic is to multiply the width of transistors in series by the number of transistors in series to maintain the original drive strength. A two input NOR gate contains two stacked PFETs, resulting in PFETs that are 10 times larger than the minimum NFET size used. Compared to a two input NAND gate, where the NFETs are stacked instead, the total gate area for one NAND input is seven equivalent units of gate area, while the NOR is 11 units of gate area. Extending the comparison to three input gates, a NAND input is eight units of gate area, when the NOR is 16 of gate area. These reasons led to a strong aversion to relying on stacked PFETs in the design. Thus, most logic was refactored to make use of NAND gates and inverters.

Physical Design

The layout of the gate driver revolves around the layout of the output transistors. A method was developed to evaluate the optimal transistor size for the arrangement. This method, and the results from it will be further discussed in Chapter 5; the output transistors will be treated as already optimized for the purposes of this discussion. The digital logic library by Ozark Integrated Circuits includes layouts for all common logic gates, as well as flip flops. These layouts are used as building blocks for the design.

A few general layout strategies were employed to enhance the design. First, all nets that should be connected (e.g. VDD, VDDIO, PVDD) are connected on-chip as opposed to off-chip. Multiple pads are included for optimal performance, but are not necessary for functional testing.

This reduces the number of connections to validate functionality, and provides redundancy in case of a packaging failure.

Another general strategy was to minimize the need for signals to cross. With only one metal layer, signal crossings must use polysilicon. The contacts and polysilicon resistance offer significant performance degradation if used extensively. In a design as complicated as this one, cross-overs are unavoidable. Priority was given first to modifications necessary to avoid a break in metal paths for V_{DD} and V_{SS}, and then to high current signal paths.

One of the goals for the gate driver is to allow for direct connection of the gate driver output to the power MOSFET gate. In order for this to be possible, the gate driver output transistors must be aligned in a way to allow for easy wire-bond access. As shown in Fig. 4.12 below, there are four primary transistor orientations that were considered. Two-layer wire-bonds are considered only if the resulting bonds do not cross heights. Placing the power pads near the boundary of the die, as in (a) and (c), allow for short bonds down to the substrate. The left-right orientation of (c) and (d) limits the maximum path length of the gate driver transistors, and also causes transistors further from the control block to have a longer drive path. Managing the wire-bond directions, options (b) and (d) require sideways connections to the power nets, reducing the benefits of a large pad. Two-layer wire-bonds can be used with both (a) and (c) to allow a power connection with a short down-bond, and a longer bond to the output pads, while maintaining the ability to use the full pad width. The chosen direction, option (a), allows for less risky wire-bonds, unconstrained power transistor length, and simple proximal placement of output slice drivers and their power supplies.

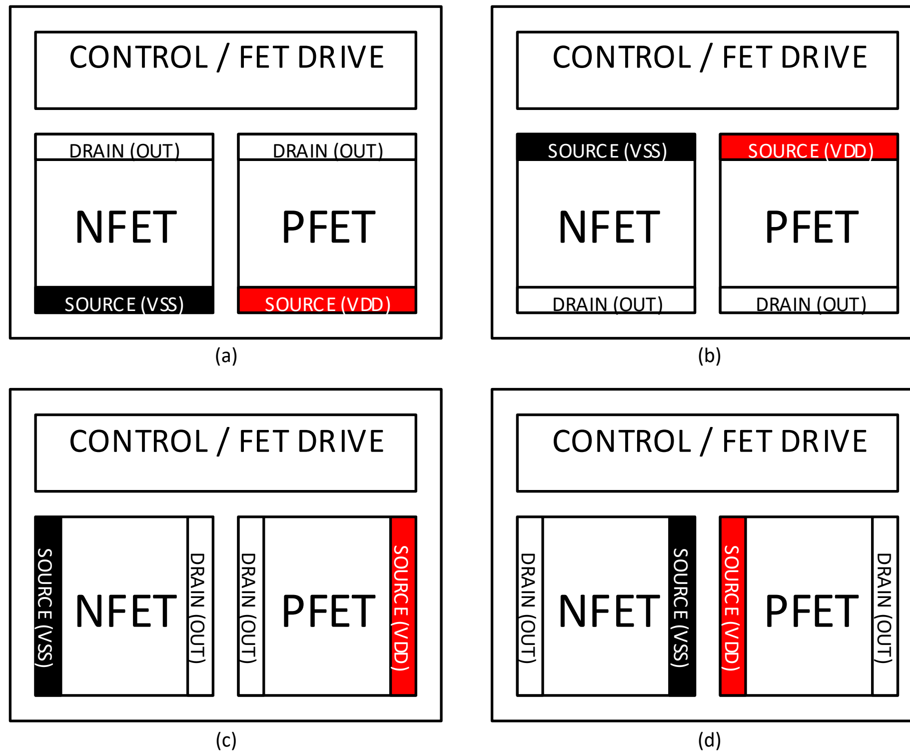


Fig. 4.12. Power transistor orientation options considered.

While the width of the output transistor slice was parametrically identified, the height was less critical to proper operation. As a result, the height was chosen to be large enough for two rows of logic and some signal routing. This size provided enough room for eight PFET and eight NFET slices, as well as sufficient room for routing V_{DD} and V_{SS}. Large metal pads were added at each end of the output transistor slice array, with two output slices sharing a pad opening. The pads were tied together, maintaining a single net for V_{DD}, V_{SS}, PULL_UP, and PULL_DOWN. The gate signal for the output transistors was carried on polysilicon beneath the metal for the output. The output buffer was then placed immediately after the pad. A slice of the layout showing two NFET slices with pads and buffers is shown in Fig. 4.13.

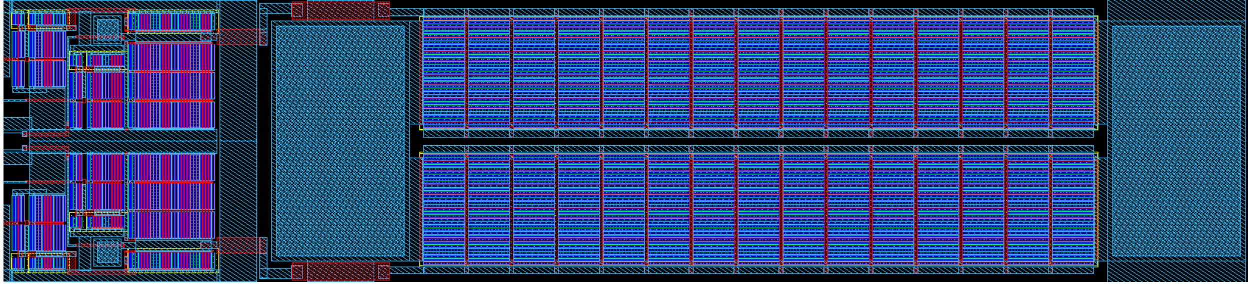


Fig. 4.13. Gate driver layout showing two NFET output transistor slices and the associated pads and buffers.

The NFET control slice was designed first, as it had the most logic of the two control blocks. Converting the NFET slice to a PFET slice involved switching the feedback inverter, and removing one inverter in the signal chain. Keeping the designs similar allowed significant effort savings in layout and verification. The final implemented NFET control slice is shown in Fig. 4.14.

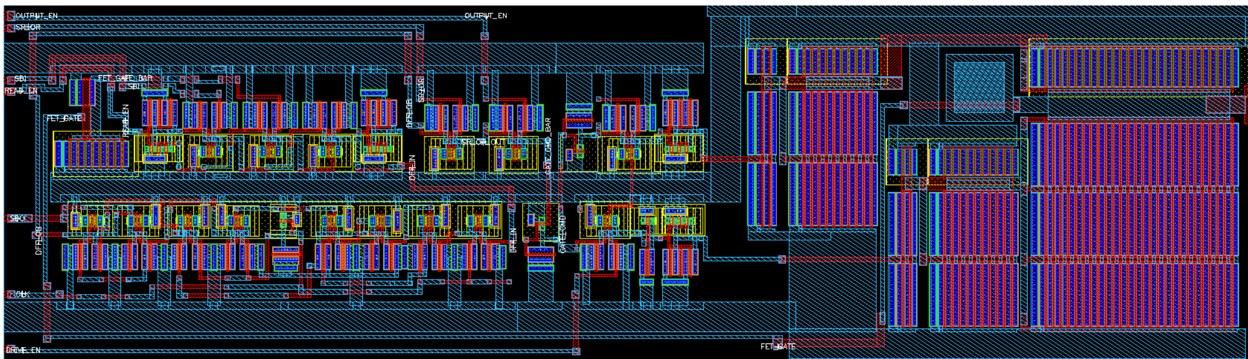


Fig. 4.14. Layout of the control logic for a single NFET slice.

The additional slices used to drive the PULL_UP and PULL_DOWN pads are very similar to the control slices, with slight modifications. The buffer originally designed to drive the output transistors was reused to drive the drains of the output transistor. The size difference between the pull-up and pull-down transistors provides a convenient opening in the layout for placing these test mode circuits, as well as the core logic for the system. The core logic layout is shown in Fig. 4.15.

Fig. 4.15. Control logic layout.

The circuit, as designed, targeted placement in a wire bonded module. This led to the usage of the standard pad library for all digital inputs, outputs, as well as power, and ground connections for the I/O pads and the logic core. The standard pad library, by Ozark Integrated Circuits, provided an easy-to-bond pad suitable for 1 mil gold ball bonding with a pad passivation opening of 90 µm by 90 µm. These pads were placed at a fixed pitch of 208 µm center-to-center, with some pads skipped due to the design having fewer pads than could be fit on a side. The pad library pads contained ESD protection diodes and digital buffers for the input and output pads. The power pads were sized larger to allow for greater flexibility with the type of wire used on the pad, enabling the use of multiple 1 mil gold wire bonds, 3 mil gold wire, or gold ribbon bonding. These pads were designed without ESD, and were 240 µm by 400 µm.

As previously mentioned, a goal for the layout was to have a single, contiguous layout shape for V_{DD}, V_{SS}, PULL_UP and PULL_DOWN. The routing for PULL_UP and

PULL_DOWN was simple, as the pads were simply bridged together with metal without considerable effort. The routing of the V_{DD} and V_{SS} net required more finesse, as the core logic required complicated distribution. Fig. 4.16 below shows the main power net routing used in the system. The final top-level layout is shown in Fig. 4.17.

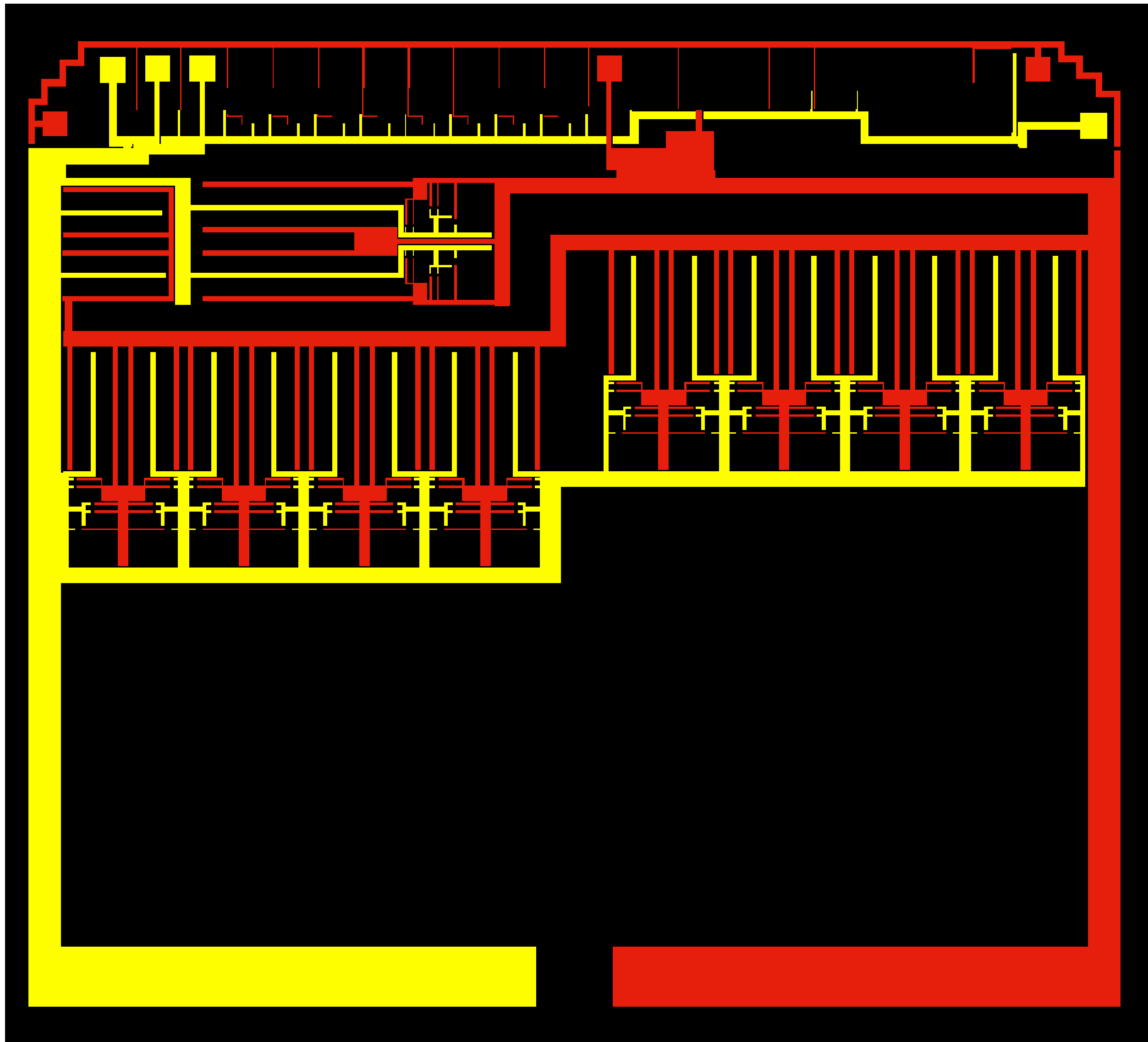


Fig. 4.16. Metal routing of V_{DD} and V_{SS} nets highlighted. Yellow area is metal 1 V_{SS}, and the red area is metal 1 V_{DD}.

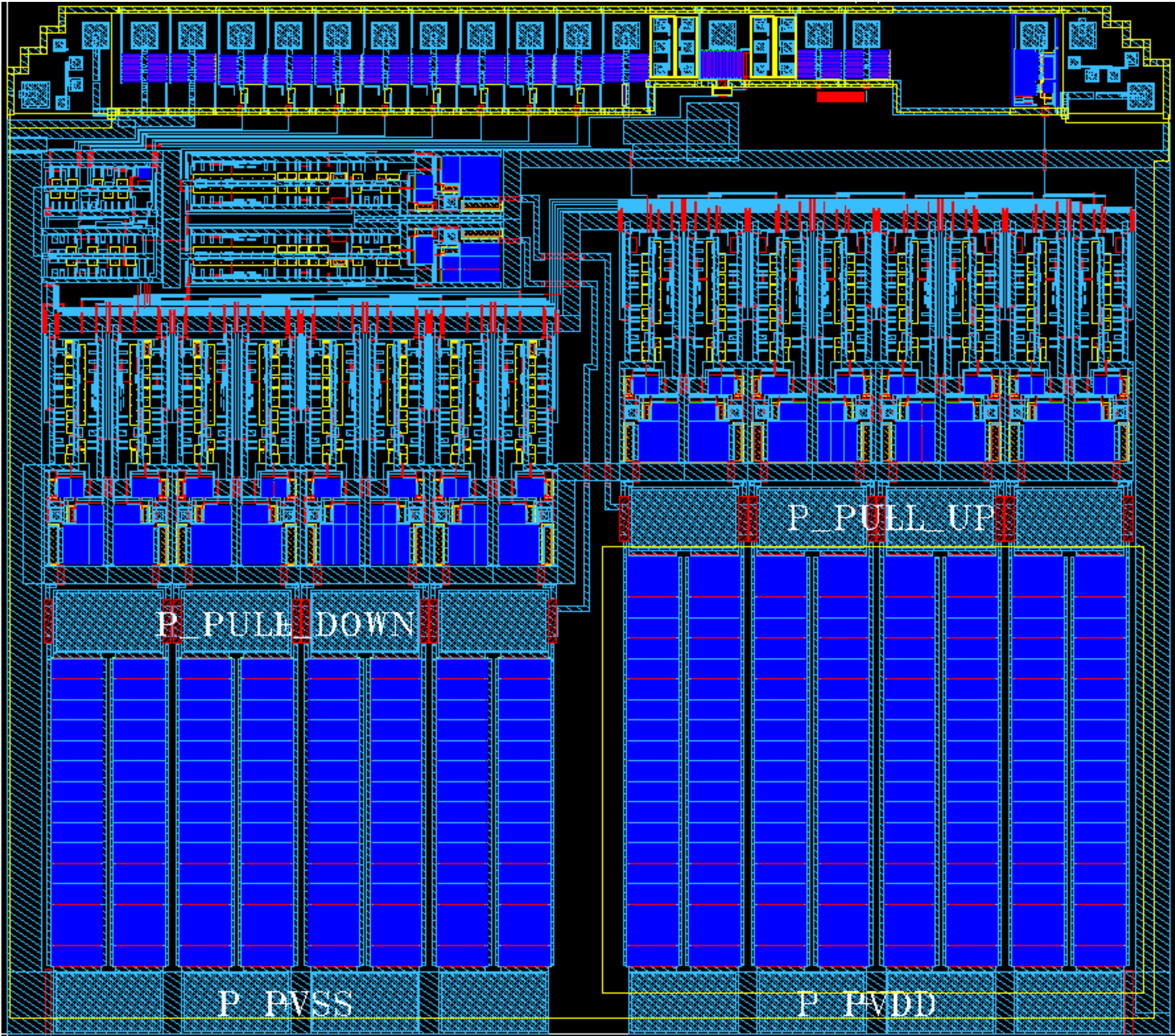


Fig. 4.17. Final top-level layout for the integrated gate driver. Dimensions are 5.0 mm x 4.5 mm.

Lessons Learned

Several lessons were learned through the process of designing a gate driver for both run 1 and run 2. Analog design in an unknown process is tenuous at best. A primarily digital approach was taken to avoid many potential pitfalls from biasing networks and level shifters required to implement alternative output drive schemes.

Care must be taken when designing the integrated circuit that the complexity required from the off-chip circuitry does not exceed the capabilities of the target environment. With an

explicit target of a power module with thick and coarse metal features, pin and external circuitry requirements must be limited to the bare minimum. This design achieves that by allowing for as few as one control input, and only a single capacitor for power and ground. The full control method requires four inputs, which is achievable even with the coarse feature size.

Minimizing layout and sub-cell variants results in being able to focus on larger design issues such as optimal transistor layout size and design-for-test. Block designs were reused when possible, and minimal variations allowed validation efforts to be shared by large portions of the design.

Fragmentation of major nets such as V_{DD} and V_{SS} can cause major issues for probe level testing and even system integration. With a single metal layer, interconnect and power compete for the low resistance metal over a polysilicon or substrate jumper. Prior efforts allowed high power signals to route on metal, isolating major power nets. Probe station testing was impossible due to the large number of pads required for testing, which was partially caused by segmented nets. By ensuring that all nets are internally connected, the design becomes more robust for probe station testing and in-module connections.

Despite the lower required pad count, later developments identified possible flip-chip options for SiC die using sintered nano-silver. However, the patterning capabilities of sintered nano-silver for flip-chip are restricted to coarser features, limiting the ability to apply this technique to the SiC gate driver.

5 Power FET Optimization

Early in the design process, the output transistor design was identified as a potential bottleneck for the overall design of the gate driver. Given the design goals of the gate driver, an approximate size could be immediately estimated for the size of the power transistor as well as the intricacies. For transistors handling large currents, the intricacies required to route and design a power transistor meant that a manually designed layout would result in a significant amount of design time and effort for each layout iteration, with low likelihood of design reuse.

Large lateral transistor layouts have many applications outside of gate drivers. Integrated power switches are also used for switch-mode converters, low-noise amplifiers, and power amplifiers. Existing design tools provide poor design aids for optimizing large manual layouts where parasitic layout effects are dominant. Various methodologies of creating a layout exist, though they end up specialized to the specific technology used [48]–[51]. Automation has been applied to transistor layout optimization to identify the best layout, but lacks the ability to inform the designer about possible tradeoffs the optimizer is making [51], [52]. With a large amount of uncertainty in design, and a wide range of design scenarios to consider, a blind or one-dimensional optimization is troublesome and removes the ability of the designer to apply design intuition and verification of results.

Automation Motivations

At the time of design in run 1, significant variability existed in the process. Wafer to wafer variation and site to site variation saw major differences. Initial die received for characterization lacked consistent devices between wafers and provided merely a hint of the anticipated final device behavior. Subsequent to the first fabrication run site-to-site variation was

large and device variation over temperature and after aging caused significant differences in drive strength.

After identifying the active gate area required to manage the 4A performance target, a large transistor array was required. While simple to implement in the schematic, large transistor arrays require significant layout effort when designed manually. Adjusting many parameters, such as individual finger width, finger-to-finger spacing, or transistor separation requires design iteration of the entire transistor array. Manual layout of each transistor scheme limits the ability of the designer to iterate through designs. Under these conditions, the limited number of iterations results in a pass/fail evaluation of each design compared to the expected specification. Once the specification is met, design iteration may stop without guaranteeing the best design.

An additional factor in the decision to automate the process was the relatively late introduction of parasitic extraction to the design kit. The time consuming task of iterating through designs was hobbled by the lack of time for evaluation. In the end, a design based on a manual array layout would result in a design based on intuition.

Design Flow Overview

An existing design flow for large transistor layout generation and validation did not exist at the beginning of the process. The first goal of the system was to automatically generate the output transistor layouts parametrically. The layout geometry needed to use the minimum design rules as often as possible so that the densest layout could be generated. Validation of the layout was the next logical step, as a layout is not useful unless it can be manufactured and matches the original schematic. Further evaluation of the layout comes in the form of parasitic extraction. Simulation of the parasitic extracted design is the easiest way to evaluate performance. Once the simulations are completed, performance metrics are extracted and plotted. The general design

and execution share similarities with [52], though layout generation and result presentation is handled differently.

Parametric Layout Generation

The foundation of any layout generation routine is the ability to create layout geometry in a format that can be processed and understood by the existing layout tools. For this design, the layout tools included Cadence Virtuoso and Mentor Graphics Calibre. Inside Virtuoso, it is possible to create layouts using parametric designs, called P-cells, and scripting is possible using the SKILL language. Alternatively, the Calibre layout verification tools are capable of using a native GDSII layout, and Virtuoso is capable of importing a layout from GDSII. A program or script external to Virtuoso can be used if GDSII is used as a universal interchange format. A Python library for generating and handling GDSII layout data, gdspy [53], was identified, and the framework for a transistor layout generation method was developed.

The Python function design for the power transistor used a few basic principles to simplify programming and allow for better flexibility in later programming efforts. The first goal was to break the problem space into smaller sub-layouts. For the transistor layout, vertical columns of MOSFETs, polysilicon tie bars, and contact arrays were generated as separate layouts and combined into a top-level assembly. Another goal was to express all parameters in terms of design rules. Specific design rules were described using the foundry rule terminology to allow easy identification of critical values and the justification for each element size and spacing. The final goal was to reduce the number of variables to focus the efforts on critical electrical parameters. Variables dictating spacing related to design rules were eliminated with the assumption that minimum design rules would be used or the design rules would be de-rated inside the MOSFET function. While each variable has a predictable effect on MOSFET

performance, the complete interaction between the variables is difficult to predict over the entire design space. These variables, listed and characterized in Table 5.1, were used to create a multi-dimension problem space for exploration under multiple models and conditions. These layout elements are visually demonstrated in Fig. 5.1. The function returns a viable layout in GDSII, a SPICE netlist for processing in future steps, and a unique identifier based on the function parameters.

Table 5.1 List of MOSFET Variables Used to Design the Output Drive

Variable	**Description**	**Impact of Increasing Variable Value:**
w	Individual FET width	• Increases total transistor width • Increases effective gate resistance • Increases active area • Decreases MOSFET contribution to RDSON
l	Individual FET channel length	• Decreases the transistor W/L ratio • Decreases drive strength
nf	Number of fingers in a vertical FET stack	• Increases gate resistance, capacitance • Increases active area
m	Number of vertical FET stacks	• Increases active area
wm	Horizontal transistor metal width	• Decreases metal contribution to $R_{DS(on)}$ • Decreases active area
wp	Vertical gate polysilicon width	• Decreases gate resistance • Decreases active area • Increases gate capacitance
wgb	Horizontal gate bus metal width	• Decreases gate resistance • Decreases active area
wc	Width of the end metal tabs	• Increases current spreading area at the drain and source terminals

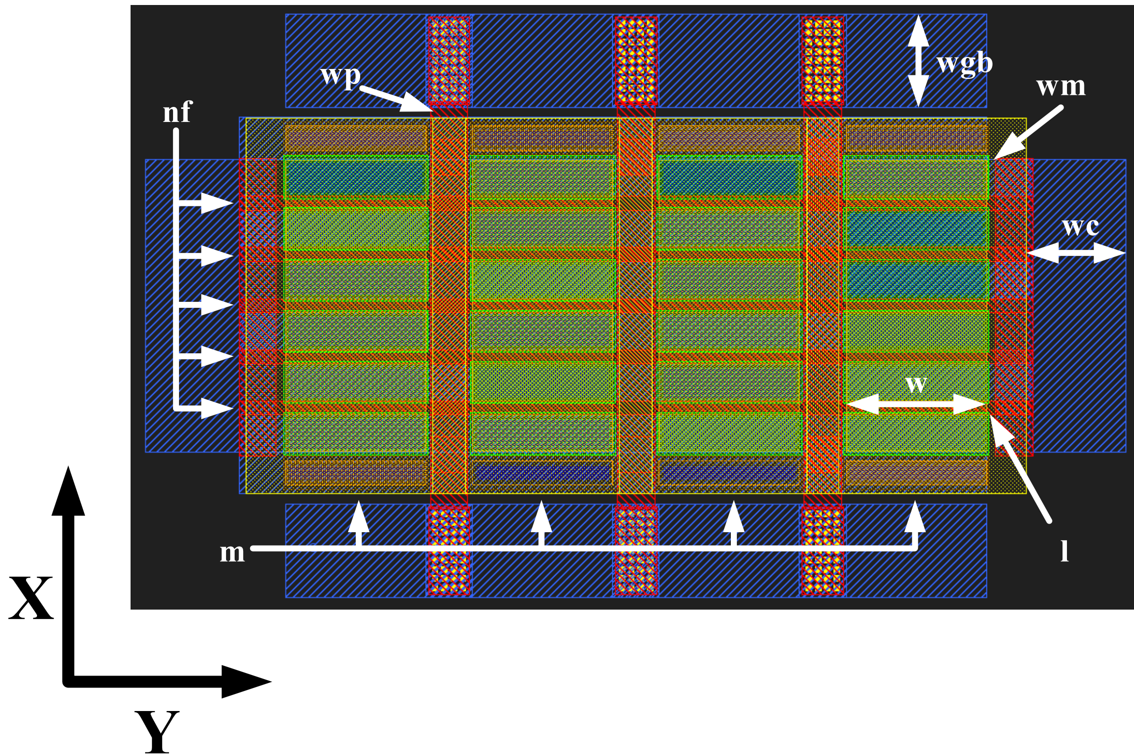


Fig. 5.1. Example layout output from the layout generation routine. In this design, there are five fingers stacked vertically (nf = 5), and there are four stacks of fingered transistors (m = 4).

Calibre Integration

With the layout and netlist generated, the raw inputs needed for Calibre [54] were generated. There are three Calibre tools used in the validation of layouts and designs – Design Rule Checking (DRC), Layout Versus Schematic (LVS), and Parasitic Extraction (PEX). Each of these is necessary to ensure that the design achieves the goals explicitly stated. For DRC, the result is important to make sure that each data point represents a legitimate, manufacturing-ready design. Additionally, failure to meet design rules indicates an algorithmic issue with the layout generation that must be addressed. The Calibre software does not offer any direct scripting interface for Python, so a command-line based interface was developed using Python. The executable, “calibre" uses a run file that specifies the source GDSII parameters and rules file while using input flags to set the type of checking performed. A source file is generated based on

the known parameters of the design rules and the design, and then the Python “subprocess” module is used to run the Calibre process with the necessary command line arguments. The text output of Calibre is directed to a file, and is parsed to read the number of design rule violations.

After the layout design rules are validated, the layout is checked for connectivity. While the schematic is as simple as possible for the device as a single MOSFET with the body and source terminals connected, the validation is important since the layout is manually generated. It is very easy to generate a netlist that Calibre understands, allowing matching of transistor width and length parameters. Additionally, running an LVS check ensures that there are no additional floating nets or transistors that are connected improperly. In addition to the layout validation concerns, layout verification is a necessary precursor to the running of parasitic element extraction. Similar to the DRC checking, a run file is built using design knowledge and known LVS settings. This file and necessary flags are used to run LVS verification between the layout GDSII and the generated SPICE netlist. The resulting output is directed to a text file, which is then parsed to count the number of errors identified.

The final PEX Calibre run is similar to the LVS run in that it requires a netlist as a source material. In the process of running, resistances are inserted into the netlist for the parasitic resistances of the polysilicon and metal layers. Two kinds of capacitances, net to net and net to substrate, are calculated, and appended to the resistor netlist. The end result is a large netlist with small resistances and capacitances distributed throughout the system. The output is logged, but the system will generate a parasitic-enhanced netlist as long as the settings are correct and the layout previously passed the LVS verification. As such, only the return value is checked for proper operation.

HSPICE Integration

With some parasitic netlists, it is possible to evaluate the effectiveness of a given layout strategy by examining the netlist directly for linked capacitances and resistances. However, as netlist complexity increases and the parasitic path includes resistances and capacitances that do not simply add together, manual evaluation of the parasitic network becomes too complicated for human or script-driven processing. As a result, the only practical evaluation method that can be automated is direct SPICE simulation.

In order to use the parasitic network, a simple simulation testbench (Fig. 5.2) is created for each layout variation. There are three basic parameters that can be evaluated through simulation: saturation current, linear region resistance, and turn-on/turn-off time. On-current and resistance are static DC parameters, while turn-on time is a dynamic characteristic. A DC voltage source is used for the drain-source supply, and a pulse voltage source is used for the gate. Two simulations are run in the testbench, a DC sweep of the drain-source voltage while V_{GS} is held to V_{DD}, and a transient simulation is run where the drain-source voltage is held to 0.1 V while the gate-source voltage is pulsed to $|V_{DD}|$. The device parameters are combined with an existing hand-crafted testbench file with all necessary simulator and model settings that remain constant for every device iteration.

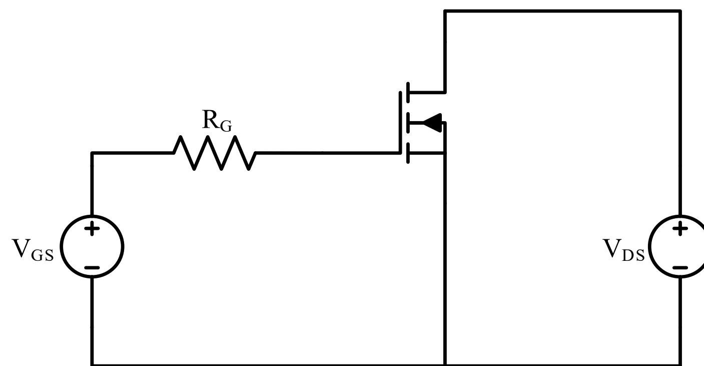


Fig. 5.2. Schematic representation of SPICE testbench netlist.

The simulation results are identified by using HSPICE [55] ".MEAS" statements to evaluate the current through the transistor during the two main operating points of interest. Direct parameters are used to measure the behavior of the circuit under conditions similar to a datasheet evaluation. Turn-on time presented a challenge for measurement as a fixed capacitive load would bias the results in favor of larger devices. Because the goal was to find an efficient layout regardless of absolute transistor size, the metric was set to measure the time required for the device under test to reach 90% of the maximum on-current with a fixed V_{DS}. While this approach neglects any Miller capacitance, it allows for the evaluation of gate resistance and capacitance simultaneously while maintaining a performance-based metric.

The other issue remaining is that normally a single simulation is performed for each testbench. However, as previously mentioned, the device parameters change dramatically over temperature and age. As a result, each simulation corner must be considered in order to properly specify the device behavior in the worst case. Another built-in HSPICE directive is used to allow the simulator to iterate through various simulator model configurations. The processed testbench file consists of six ".ALTER" configurations, with the ability of easily adding more as new models become available. The simulations ran included the four temperature nodes available (25 °C, 100 °C, 200 °C, and 300 °C), as well as the "fresh" models at 200 °C and 300 °C. As a result, the output consists of a comma separated value file with each ".MEAS" result for each model configuration.

As with the Calibre simulations, once the simulation testbench was compiled, the HSPICE simulator executable was called with the proper runtime arguments. The runtime transcript of HSPICE was recorded for off-line troubleshooting and, once the simulation completed, the results file was parsed into a Python Numpy [56] array.

Collection and plotting of data

With a basic framework of evaluating a single layout, focus shifted towards using the framework to evaluate the effects of several variables in a larger problem space. While the computer is capable of processing a multi-dimensional analysis, it is difficult to coherently present results from more than two dimensions of parametric analysis. Two related parameters were chosen to provide a set of data that can be clearly evaluated over all different corners. The Python library matplotlib [57] was used for plotting of data structures and generation of plots. The two variables changed are selected for the x and y axes, and a color contour plot was used to indicate performance across ranges used for the variables. A subplot was generated for each simulation condition tested. For the figures in this chapter, six conditions were tested according to the models available: 27 ℃ nominal, 100 ℃ nominal, 200 ℃ nominal, 300 ℃ nominal, 200 ℃ aged, and 300 ℃ aged. A common color index was used across all plots so that individual cases were directly comparable. After the plots were generated, they were saved to an image file. The minimum or maximum value on each subplot was annotated for later reference.

One side effect of the method of layout generation is that the layouts are not generated to fit inside a specified bounding box. In other words, sweeping across a given parameter will result in a constantly changing layout size. A form of normalization is required in order to compare the layouts on the same graph. This includes normalization based on total area, as well as normalization by layout height. The reason for normalizing by height is that some measured parameters such as current density indicate a choice with optimal current density, but the linear current density is too low to meet the current drive requirements with the given layout topology. Three primary plots are generated: Current density (A/mm^2), Linear current density (A/mm), and Turn-on time (ns).

Parametric evaluation of problem space

The order of optimization for the transistors was to begin with identifying important characteristics and settings for static parameters, and then focus on transient issues. To that end, the form factor is an important starting point for evaluation. Given a 5 mm x 5 mm area for design, and the topology selected from Fig. 4.12, an active area is limited to 2.5 mm in the X direction, and up to 3.0 mm in the Y direction. The Y direction is limited due to space requirements for the drive and control logic as well as bonding pads. Initial evaluation of the problem space focused on the highest current density achievable, which used a Y dimension between 400 and 600 μm. However, extrapolating the size of the active area of the highest current density configuration shown in Fig. 5.3 to meet the current requirement of 8 A results in an X dimension of 3.3 mm, clearly exceeding the X dimension boundary of 2.5 mm set earlier. The current density metric of A / mm^2 in Fig. 5.3 establishes a lower bound for the size of the transistor structure. The peak current density width occurs at an "m" value of eight, for an active transistor width of 480 μm.

While the area efficiency is not optimal, the maximum current of the layout can still be increased. In order to consider a larger transistor capable of higher current, a second normalization of the current data is performed, normalizing the current to the layout's Y dimension. This normalization gives the effective current if the transistor had a Y dimension of 1 mm. Evaluating this in A / mm, the total X dimension can be extended further than the current density per unit area metric indicated, depending on the temperature of interest. A new maximum current is then identified, where increasing the size of the device no longer increases current capacity, and is the upper limit for an X dimension. The total width providing the maximum current decreases with increasing temperature, while the transistor conductance increases at

higher temperatures. As a result, peak currents occur at a total width of 1.32 mm at 25 °C and 1.08 mm at 200 °C, setting a maximum upper bound on total width at 1.32 mm. The effective width chosen was 1.20 mm, which provides insurance against poor transistor drive strength at low temperature. A greater width would result in less dependence on the transistor drive strength as metal losses would consist of a larger portion of the effective resistance, at the cost of increased die area and gate capacitance.

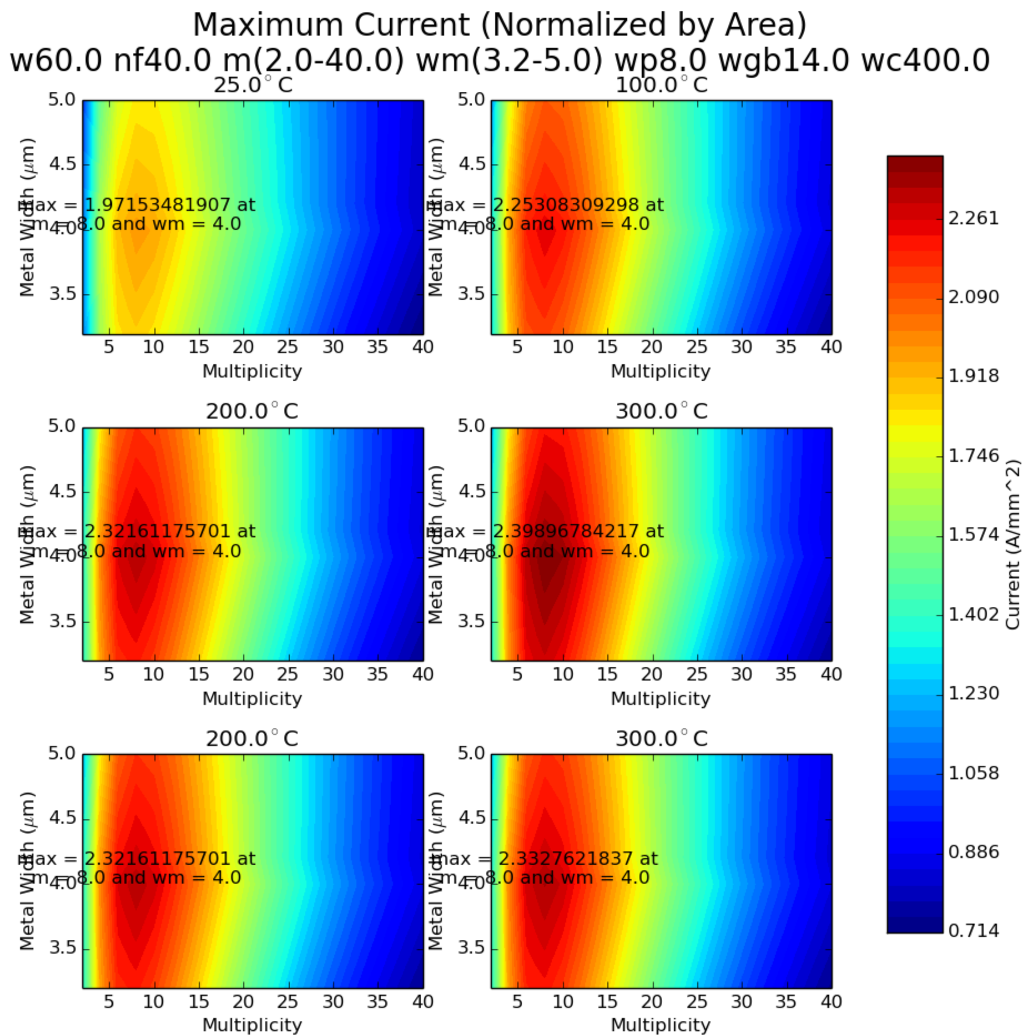


Fig. 5.3. Comparison of effective transistor width versus the width of the main metal conductors. Transistor width is adjusted by changing the number of 60 μm wide transistor blocks (multiplicity). Designs are simulated over temperature, normalized to total layout area, and plotted.

The width of metal used shows a clear preference towards 4.0 µm in both current density metrics in Fig. 5.3 as well as Fig. 5.4, and this is directly a result of the design rules. The nominal width of the drain and source metal in the Cadence p-cell is 3.8 µm, but the design rules allow a range of metal widths from 2.7 µm to 4.8 µm without decreasing the transistor pitch. Below 4.8 µm, the transistor pitch is defined by the gate, drain, and source diffusion geometry. Once the width exceeds 4.8 µm, the design rules result in a direct addition to the transistor pitch. The maximum width is de-rated down to 4.0 µm to prevent metal overlapping the polysilicon for the gate. The interaction between these two layers is unclear, but without measured yield for metal overlap, metal overlapping polysilicon was prohibited to reduce the potential risk for devices. Increasing the metal width from the minimum 3.2 µm up to 4.0 µm results in a performance increase without increasing the total area. Increasing the metal beyond 4.0 µm improves the performance of each individual transistor, but not at a faster rate than adding more transistors because the transistor losses still dominate the metal losses in all cases.

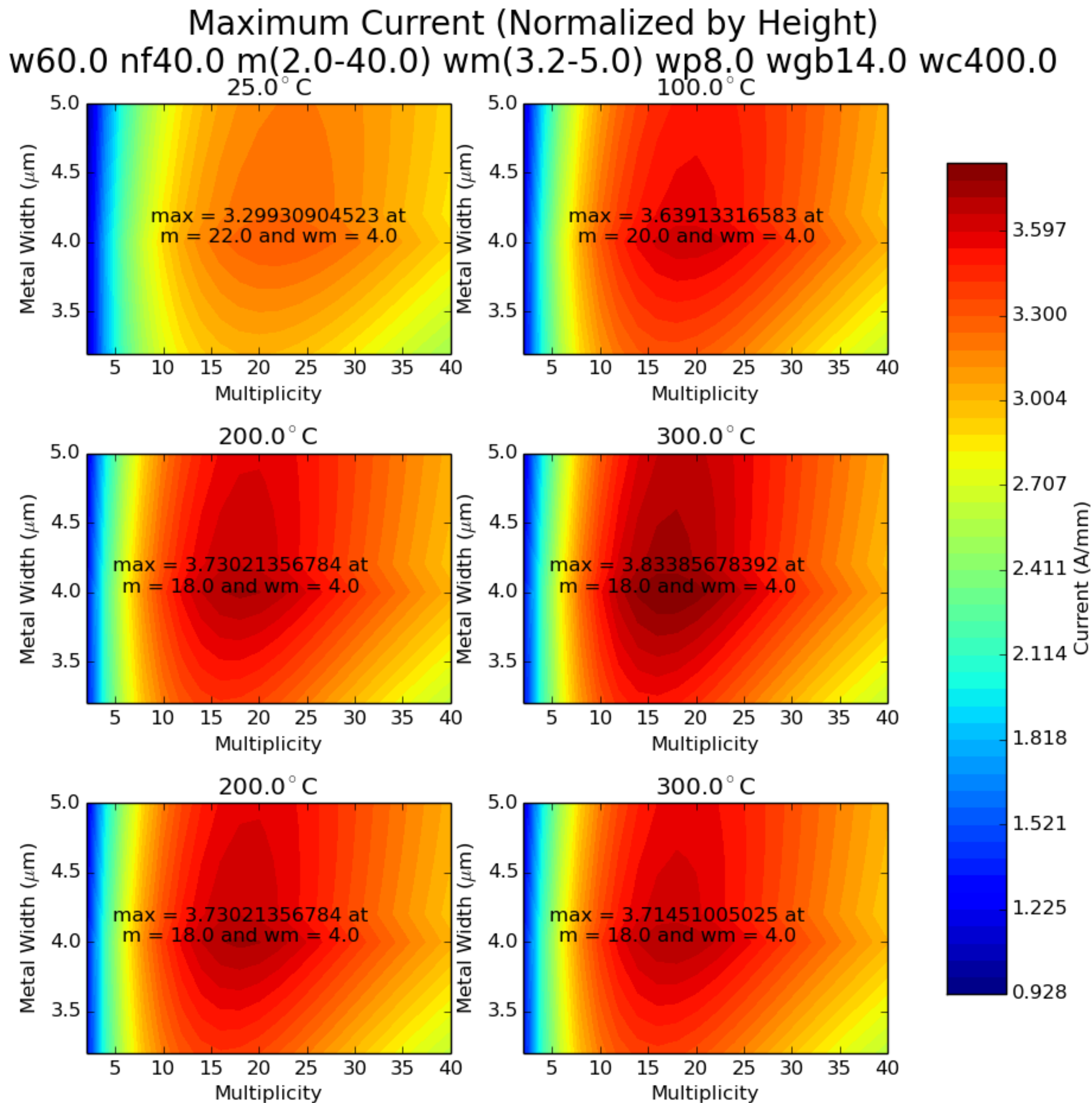


Fig. 5.4 Parametric evaluation of the static current characteristics of a NFET over size and metal width. Current is presented as A/mm. Maximum current occurs when total width is 1.32 mm at 25 °C to 1.08 mm at 200 °C. Spending additional area on metal width does not improve current density.

With the size aspect of the transistor defined, attention turned towards improving transient performance through improvement of the gate drive distribution path for the output transistor. As previously listed in Table 5.1, there are several parameters that affect the load presented by the gate of the transistor. The transient performance parameters all decrease the

fraction of active area of the transistor, reducing the current density of the device. As such, the philosophy was adopted that gate distribution network should be improved without going beyond the point of diminishing returns.

While the static parameters defined the effective width needed for the desired output performance as 1.20 mm, the number of subdivisions (or multiples) of transistor stacks was not addressed. Each added subdivision places a gate bus (wp) on the polysilicon layer that bridges all the gates together. The gate bus also is connected to a metal rail parallel to the transistor orientation that provides a low-resistance conveyance across the length of the output transistor. When working with a fixed total transistor width, increasing the multiplier (m) while decreasing the transistor width (w) also decreases the total gate width that is carrying current, which also increases performance. In order to evaluate the transient performance, a sweep was configured to iterate the width of the transistor and the number of multiples. Due to limitations in the layout generation and plotting methods, a constant total transistor width could not be implemented as a single axis. Instead, that line can be observed on a plot with the transistor width and multiplicity on separate axes. The individual width is set to range from 40 μm up to 150 μm, with a multiplicity ranging from 8 to 28 to match the total width identified in the previous static DC configuration.

The number of divisions for the gate bus increases the area usage. Each division carries a Y dimension penalty of *wp* plus the additional design rule minimums for polysilicon to non-transistor active area. The highest current density would occur with no divisions (m = 1), but would turn on very slowly as the gate polysilicon forms a distributed RC network. Fig. 5.5 demonstrates the decrease in peak current due to higher number of transistor divisions. Superimposed on Fig. 5.5 is the total transistor width response from Fig. 5.4 for a fixed metal

width (wm = 4.0). Conversely, increasing the number of divisions will improve the distribution of the gate signal. The turn-on time for the different geometries is shown in Fig. 5.6, showing the expected turn-on benefit from a higher number of divisions. A MOSFET width of 80 μm and a corresponding multiplicity of 15 provides minimal degradation in maximum current, while providing a benefit to turn-on time.

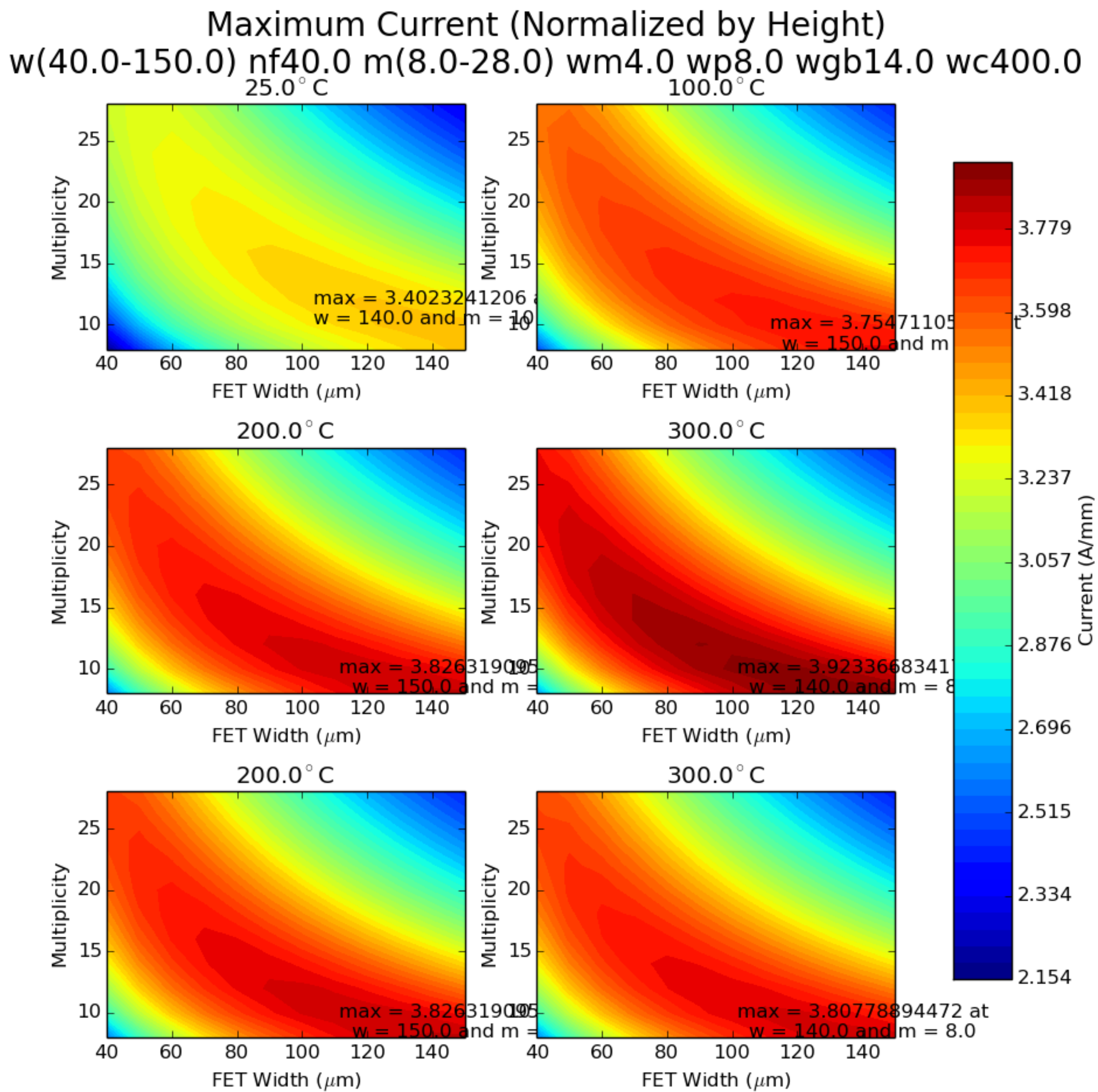


Fig. 5.5. Maximum current response while comparing the number of divisions in a design.

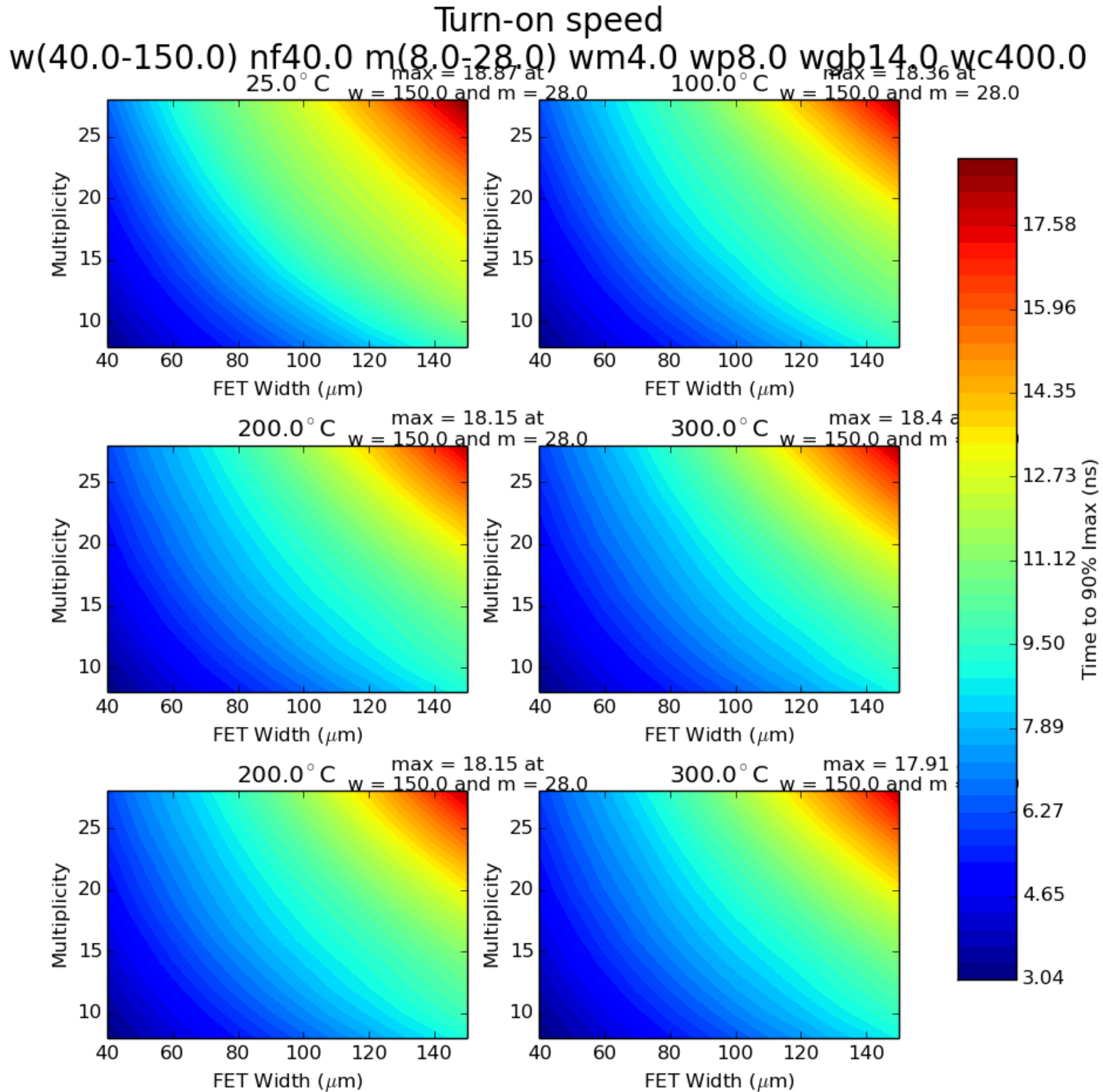


Fig. 5.6. Turn-on time comparing the number of divisions in a design.

Once the number of divisions was determined, the width of each gate signal distribution element was evaluated. The distribution consists of a metal gate bus on the top and bottom of the layout with a width of "wgb". These connect to the vertical polysilicon busses that then connect to both ends of the MOSFET gate polysilicon. By increasing the width of the metal and polysilicon gate buses, the input gate resistance decreases, which decreases the turn-on time measured in Fig. 5.7. Both gate bus elements decrease the ratio of active transistor area to total

area, and cost total DC current performance. The current density per unit area (A/mm^2) is a useful metric for monitoring the total area consumption of turn-on performance enhancing parameters. The other existing DC metric of current per unit length is not an effective metric as the current is not expected to significantly change through the variation of the chosen wp and wgb parameters. Another reason for using total area is to include the area used by the wp parameter, which is not normalized in the A/mm metric. The total current density per unit area (A/mm^2) is shown in Fig. 5.8. The current density graph shows the current density decreases faster with a linear increase of the wp parameter than the wgb parameter, and this can be easily explained by examining the reference layout in Fig. 5.1. The wgb parameter controls the height of both the top and bottom metal gate bus, and increases the layout X dimension by two times the change in wgb. Conversely, the polysilicon gate bus width occurs (m-1) times in the layout, resulting in the layout Y dimension increasing at a rate of 14 times the change in wp. Examining the 25 °C and 100 °C cases in Fig. 5.7, using minimum values for either wgb or wp results in a noticeable degradation in performance. A wp of 8.0 μm and a wgb of 14.0 μm were selected as a balance between adding additional transient performance at the cost of current density.

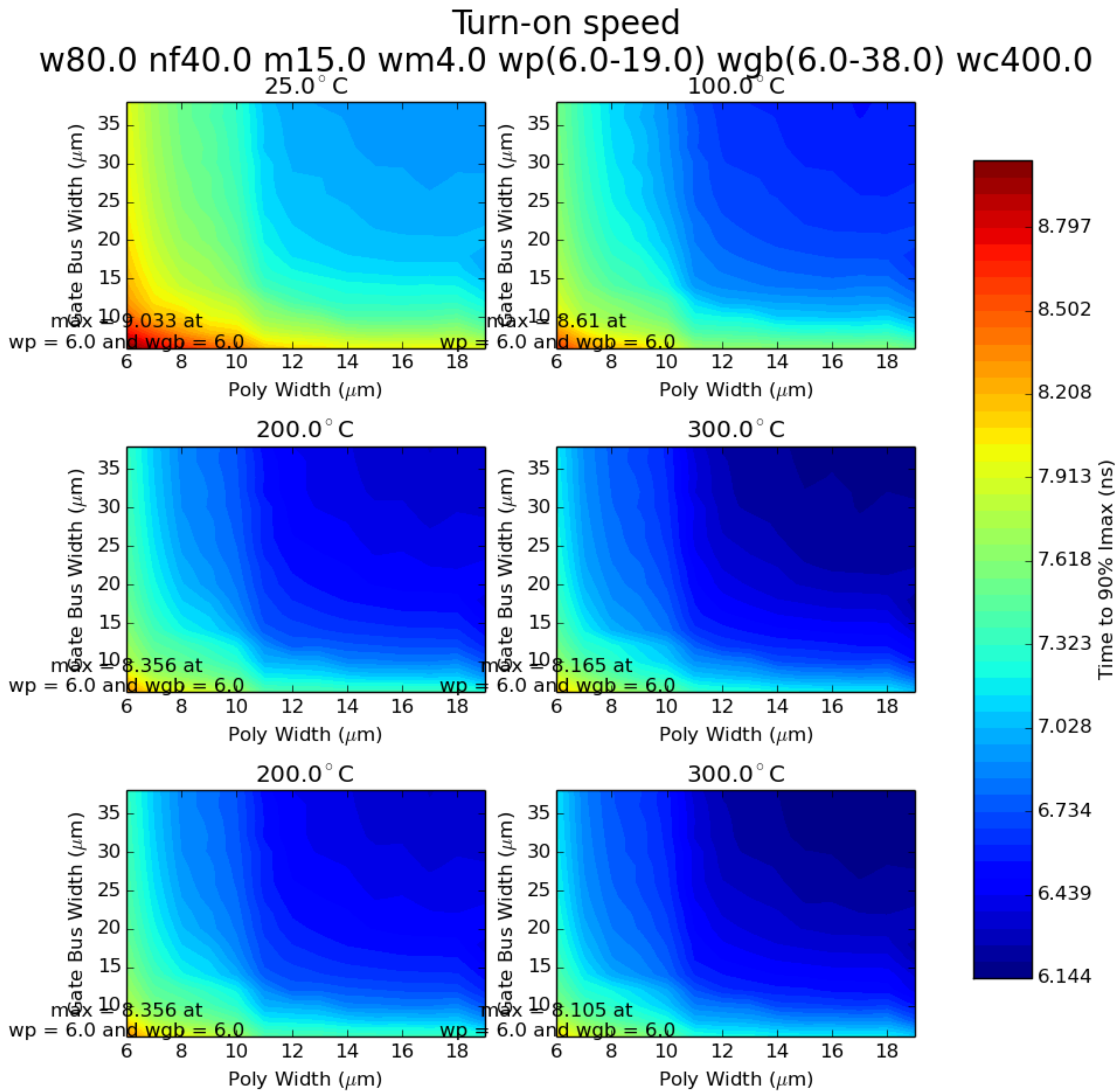


Fig. 5.7 Turn-on speed comparing gate signal distribution sizes (wgb and wp).

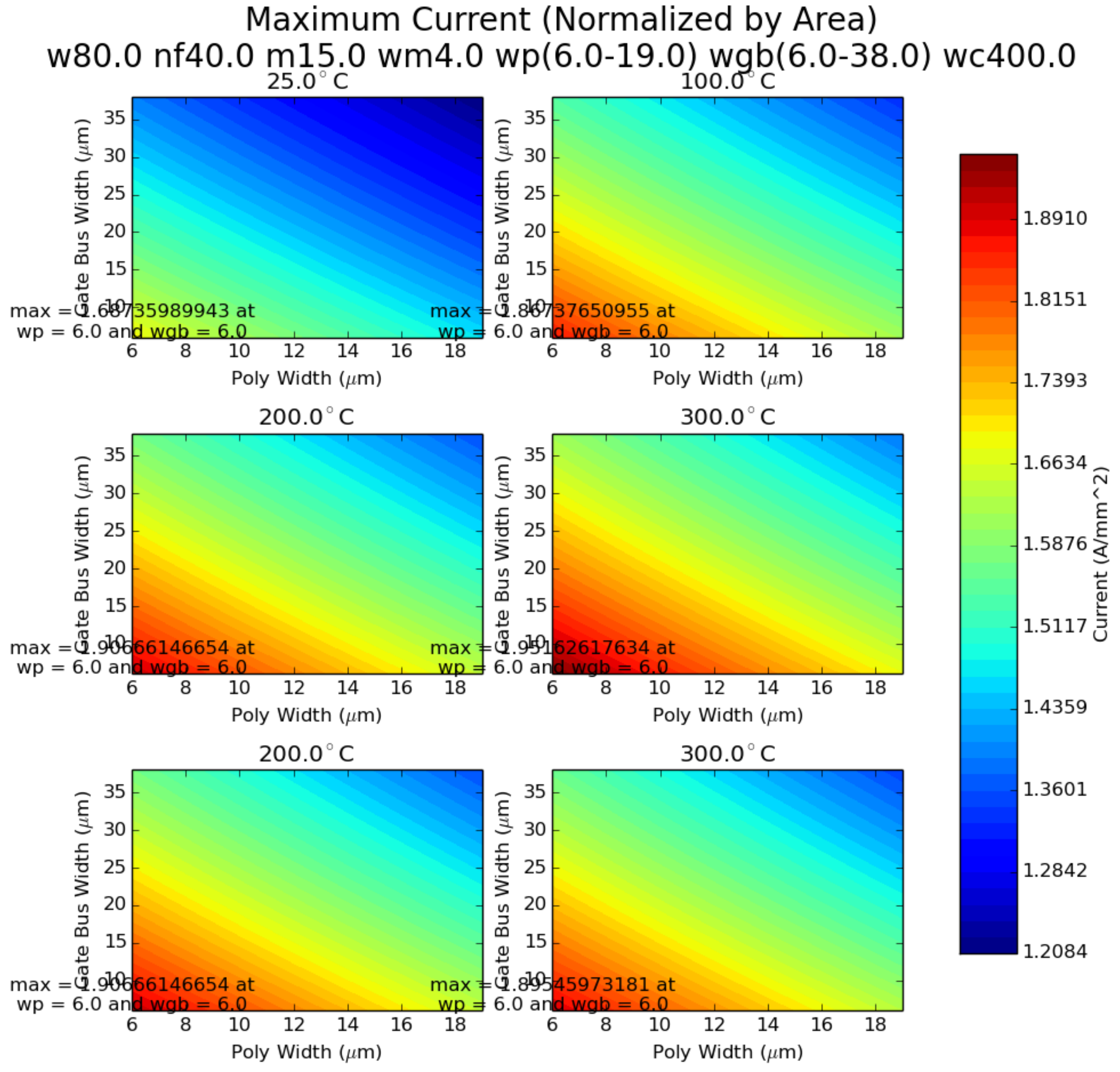


Fig. 5.8. Current density of different gate signal distribution (gp v wgb).

The remaining parameter to identify is the number of fingers in each stack. The more fingers in a transistor stack, the higher the ratio of active area to total layout area. The concern with increasing the number of transistors in a stack is the current distribution capability of the polysilicon gate bus. To that end, the main concern is the turn-on time increase with the increased number of fingers and whether increasing the width of the gate bus polysilicon improves any transient performance losses. Fig. 5.9 shows the effect on turn-on time that both

the polysilicon width as well as the number of fingers have on total performance. Increasing the polysilicon width shows negligible impact in all cases below 50 fingers. At higher number of fingers, the benefit increases slightly, though the dominant effect on performance in Fig. 5.9 is the number of fingers in the transistor. While this variable presents itself as having a great degree of freedom, in reality the number of fingers is dictated by the total number of subdivisions required by the layout as well as the active area. In order to achieve the fit required with the drive circuitry and achieve eight separate transistor slices, the value of nf was set to 30.

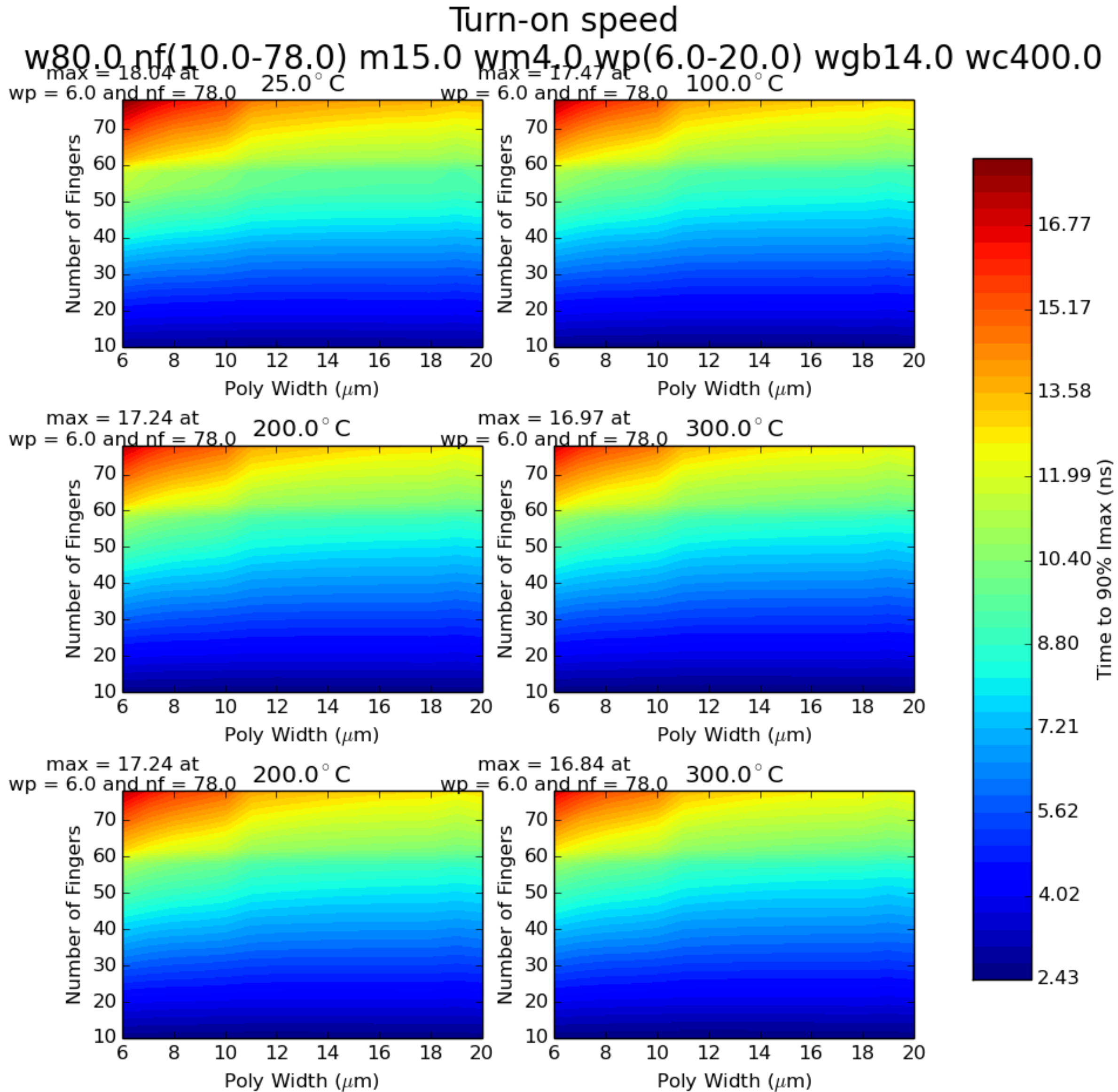


Fig. 5.9 Turn-on time comparing poly distribution width and the number of fingers.

Following the previous optimizations, the transistor cell properties listed in Table 5.2 were selected for the NFET transistor cell. These parameters resulted in a final transistor W/L ratio of 36mm/1.2µm for each NFET slice, and a total of 288 mm / 1.2 µm for the entire pull-down network. With the NFET slice identified, the PFET design remained. The gate capacitance of the PFET matches the gate capacitance of the NFET, and the same metal and polysilicon resistances apply for both cases. The transient-influencing parameters of the NFET design were

duplicated for the PFET, with the only remaining question being the total width required for a functional gate driver.

Table 5.2 NFET Transistor Array Parameters

Parameter	w	l	nf	m	wm	wp	wgb
Value	$80.0\ \mu m$	$1.2\ \mu m$	30	15	$4.0\ \mu m$	$8.0\ \mu m$	$14.0\ \mu m$

In order to identify the optimal PFET characteristics, the simulation performed in Fig. 5.3 is replicated for the PFET model. Case statements adjust the polarity of voltage sources in the SPICE testbench for a PFET case, allowing for minimal code change to examine PFET devices. The PFET current density plot (A/mm^2) is shown in Fig. 5.10, and the linear current density (A/mm) is shown in Fig. 5.11. An immediate anomaly can be seen between the two 200 ℃ and 300 ℃ cases. The middle row contains the models for devices that saw multiple temperature cycles over extended time at elevated temperatures. These run 1 structures showed significant degradation in performance after multiple thermal cycles, resulting in an apparent decrease in performance as temperature increased. Additional devices were characterized with the intent of minimizing thermal cycles and time at high temperatures, giving the "fresh" models in the bottom rows. For the purpose of the gate driver design, the original "aged" models show poor performance that cannot be compensated, while the "fresh" models provide acceptable drive strength.

The PFET design provided a wide range of acceptable values, from the minimum effective width of 960 µm shown in Fig. 5.10 (m = 12), up to the linear current limit of 2,240 µm (m = 28) shown in Fig. 5.11. Again, the metal width of the transistors stayed at the optimal 4.0 µm value for the NFETs, which is expected as the metal losses would be even smaller in the

weaker PFET. A multiplicity of 20 was selected to provide additional current drive over the existing NFET size of 15, while not approaching the point of diminishing returns on the “fresh” 200 °C and 300 °C models. Additional multiples were limited by the total area available. The result is the PFET layout using the parameters listed in Table 5.3. The multiplicity was the only parameter to change.

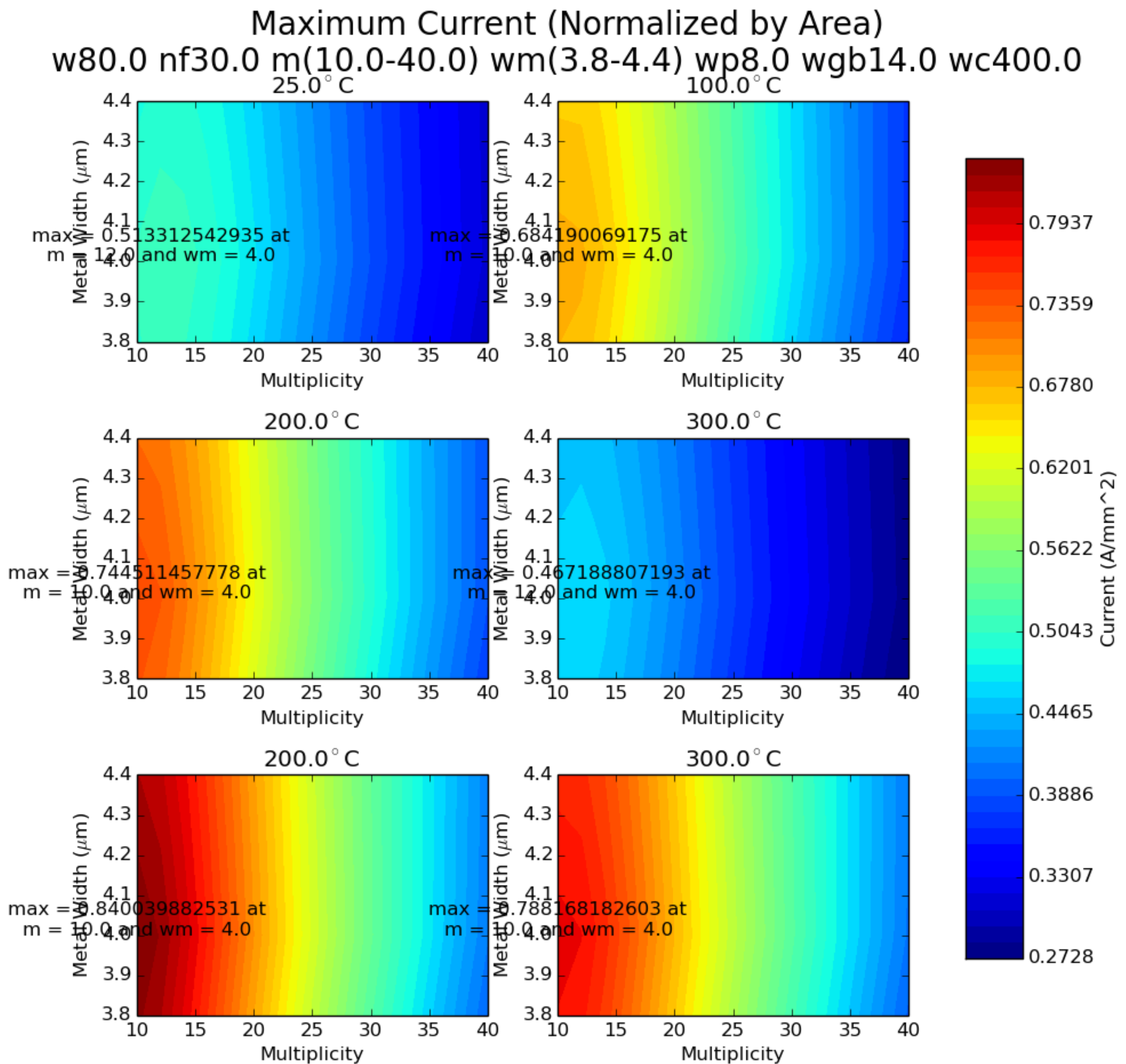


Fig. 5.10. PFET current density in A/mm^2.

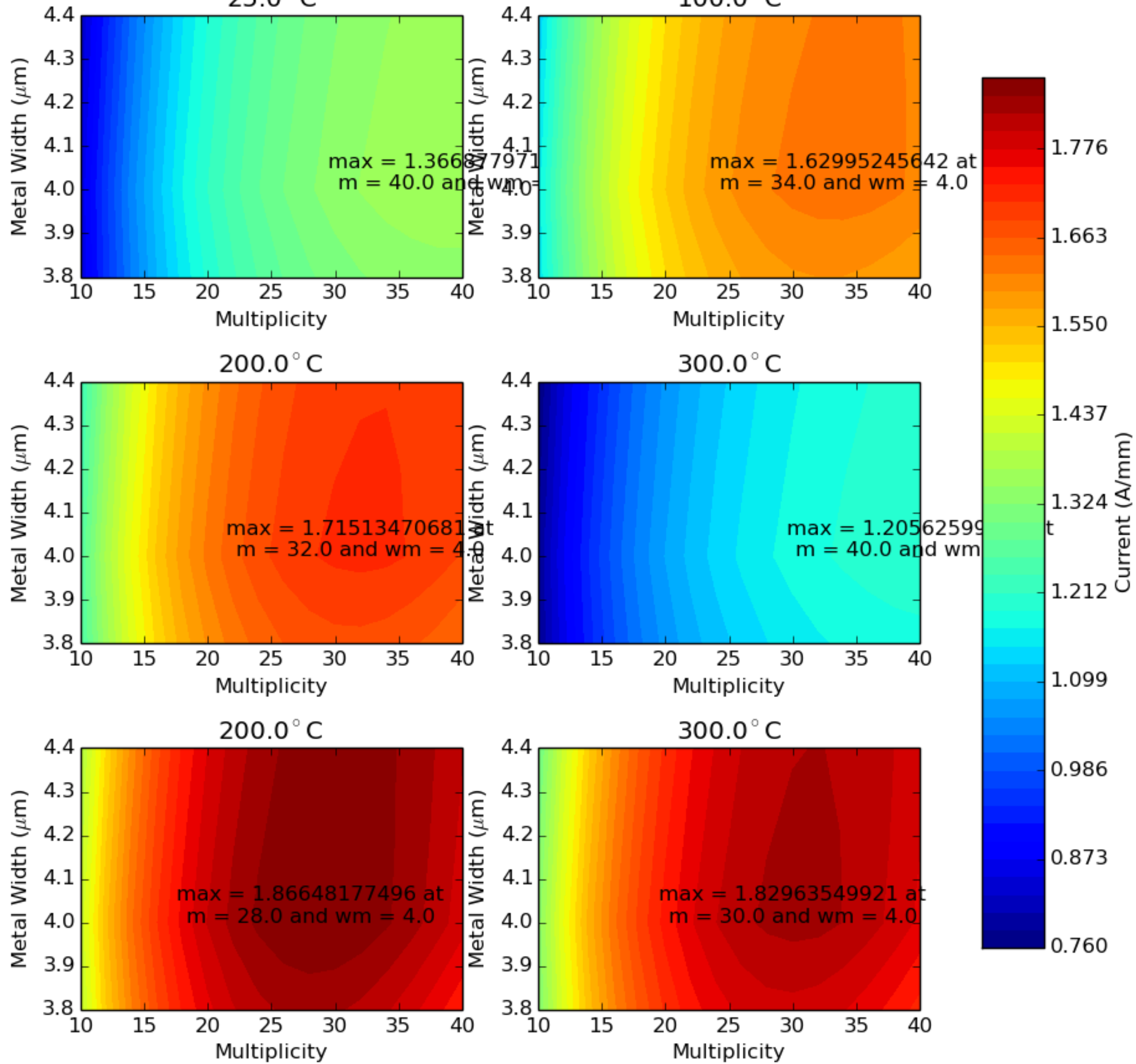


Fig. 5.11. Maximum current in A/mm for a PFET.

Table 5.3 PFET Transistor Array Parameters

Parameter	w	l	nf	m	wm	wp	wgb
Value	80.0 μm	1.2 μm	30	20	4.0 μm	8.0 μm	14.0 μm

Summary

A high quality layout with optimal parameter selection was required for the gate driver IC. A manual approach to layout generation and evaluation was deemed inefficient and likely to lead to a suboptimal design, so a series of numerical methods were developed to evaluate the parameter space. Parametric layout generation of output transistors was accomplished, as well as automated layout verification. Simulations including extracted parasitic resistances and capacitances allowed for informed comparisons on layout performance across all simulation scenarios. Layout results were plotted to allow for easy comparison between varying layout parameters.

The tool was used to manage the complexity of geometrical-performance tradeoffs and inform the designer as to which selections would lead to the best solutions for the NFET and PFET transistor layouts. Static DC characteristics were used to identify the required transistor size, while subsequently parameters affecting switching performance were adjusted. The end result was two GDSII layouts that were identified as "best" layouts that were imported directly into the Cadence Virtuoso environment.

6 Design for Test

Motivation

With the initial design of the gate driver for run 1, the functionality was designed with the assumption of good circuit yields. Unfortunately, circuit yields were very poor from the initial batch of wafers. A number of design decisions made for the initial design resulted in a circuit that was difficult to diagnose and evaluate, and contributed to the failure of the first run to produce a working gate driver.

The initial design required the connection of many pads in order to demonstrate functionality. The original design traded on-chip complexity for packaging complexity to eliminate the need for extensive V_{DD} and V_{SS} routing and the corresponding compromise in control signal routing. The output pads joined two transistor slices, but the V_{DD} or V_{SS} connection for each output transistor slice pair was isolated, requiring four power V_{DD} and V_{SS} connections. The buffers driving the gates of each transistor slice was powered by the closest power V_{DD} or V_{SS} connection and an additional power pin to eliminate the need for routing the opposite rail across the design. The ESD and I/O V_{DD} and V_{SS} were isolated from the core V_{DD} and V_{SS}, resulting in the addition of two more pairs of pins required for operation. Fully powering the design required a total of 14 power and ground connections. In a package or a module, this number of connections is not a significant drawback. It was necessary to drive or actively control at least 10 input pads in order to demonstrate functionality, shown in Table 6.1. Although a packaged part with this number of connections would not be a concern, available probing equipment was limited to eight connections. This limitation prevented effective probing of the device to verify functionality.

Table 6.1 List of Connections Required for Minimal Pass 1 Functional Testing

Pad	Voltage
VDD_IO[1|2]	15
VSS_IO[1|2]	0
DVDD_[PD|PU]	15
DVSS_[PD|PU]	0
PVDD_PD (pull-down only)	15
PVSS[1-4]	0
GD_OUT	Output
SR_OR_[PD|PU]	15
OE_[PD|PU]	15
DVDD_COMMON	15

To compound the difficulty of probing the die, the pad metallization proved to be unreliable. The Raytheon HiTSiC process contains a high temperature refractory metal not commonly used in standard IC processes. This metal, by itself, cannot be directly wire bonded. The pad passivation openings are gold plated as the last step in manufacturing to provide a wire-bondable surface. However, this plating adhered poorly to the pad metal, resulting in poor pad durability. The poor adhesion caused the gold plating to detach during gold ball bonding immediately after attaching the gold ball to the pad's gold plating, leaving bare un-plated pads, shown in Fig. 6.1. Further attempts to bond a pad with missing gold plating were generally unsuccessful, giving poor wire bonding yields. In addition to the poor adhesion, pads that were probed on the probe station before bonding demonstrated much higher failure rates. This reduced the viability of probing to select a functional circuit for packaging, as the validation of the circuit would lead to a device that could not be packaged. With the high pad count of the gate driver, this made the circuit difficult to package.

Fig. 6.1. Bonding pad from the first pass after attempting 1 mil gold ball bonding three times. The gold coating on the pad (yellow) has been torn off where the ball bonds were made, leaving the underlying metal below.

The final nail in the coffin for the first design was poor yield of large power transistors across the wafer. Test structures placed across the wafer of single transistor slices showed a high incidence of failure, primarily from gate to source shorting and gate to drain shorting. One probed wafer demonstrated a yield of less than 60 %, with no indication of a pattern based on device channel length or positioning on the wafer. In addition to the large output transistor slice failures, smaller arrayed transistors, such as the ones used in the output for the shared pad library, demonstrated problems with other designs fabricated on the wafer. Consulting with the foundry, these findings were confirmed as a manufacturing fault, due to a specified design rule providing insufficient clearance between the source and drain contacts to the gate region. While

this was addressed in future runs by a design rule change and improved drain-source contacts, the existing run could not be modified. With such a high failure rate, the chance of identifying a device with all 16 output transistor slices functioning was highly improbable. In order to have a 10 % chance of a complete 16 transistor gate driver, individual transistor yield should be greater than 86.5 %. This low yield, combined with the low bonding yield and the inability to use a probe station to evaluate the health of a given die, resulted in a very low yield circuit that was untestable. While the pad plating adhesion and the power transistor shorting were identified as manufacturing defects and not a mistake from the design, the impact of the defects was magnified by the lack of consideration for testability.

Characteristics of failures

The primary failures identified from the first version were related to the output transistors. The core logic did not exhibit issues, nor did core logic gates used in other designs show significant issues. The only other circuits consistently demonstrating the faults seen in the gate driver were the digital output pads. Based on this observation, and the requirement that the output devices operate for a design to be functional, the primary focus on design improvements were the output transistors. As each output transistor is an array of smaller transistors, a short-circuit condition could affect a small fraction of a large output transistor array. This leads to partial failures, which may not completely compromise the output transistor functionality. Some failures begin as abnormally low gate voltage tolerance ($< V_{DD}$) that develop into a short circuit once sufficient voltage is applied. An example is shown in Fig. 6.2, with a V_{GS} sweep that fails mid-sweep.

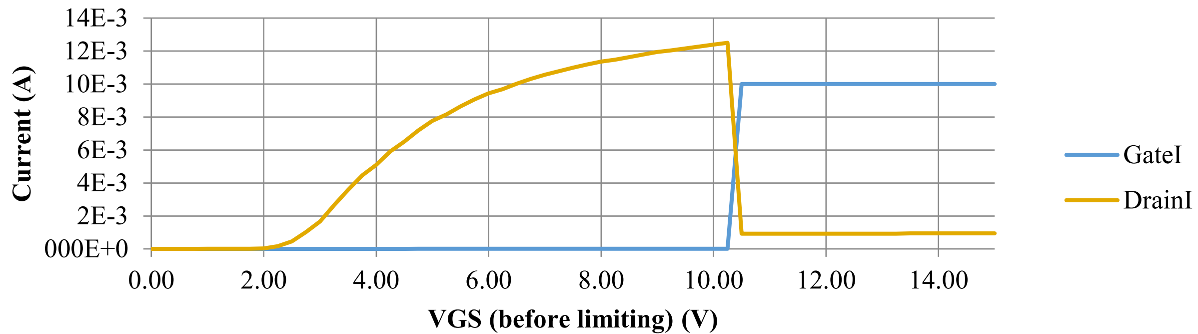


Fig. 6.2. Gate oxide breakdown on a 1.2 μm channel length transistor during a V_{GS} sweep, with a V_{DS} of 0.1 V. Breakdown occurs at a V_{GS} of 10.5 V, and I_G reaches the programmed current limit of 10 mA. Subsequent runs indicate a gate-source short of 375 Ω.

From wafer testing, the primary failure mode was the shorting of the gate to the drain or source. A much rarer fault of a drain-to-source short was observed as well. The magnitude of the fault was variable as well, as some designs exhibited amounts of leakage to the drain and source that were not necessarily fatal to the design. To improve the potential yield of a die with multiple output transistor slices, the failure modes and their impact on the rest of the circuit is important.

The rare drain-source short, illustrated in Fig. 6.3, directly impacts the entire output stage. In moderate leakage current conditions, the drain-source short will show up as an increase in quiescent current when the transistor is in the off state. An example of an NFET drain-source fault is shown in Fig. 6.4 with varying values for R_{short}. As the magnitude of the failure increases, the output transistor is capable of drawing current greater than can be tolerated by a design specification, leading to parametric failure of maximum quiescent current, insufficient output voltage swing, or even reduction of the power supply voltage due to excessive current consumption. This failure mechanism cannot be directly mitigated as long as the transistor remains connected in-circuit.

$$I_{DDQ} = D \cdot \frac{V_{DD}}{R_{short} + R_{DS(on)}} \tag{6.1}$$

$$D = \frac{t_{on}}{t_{on} + t_{off}} \tag{6.2}$$

$$V_{OUT} = V_{DD}\left(\frac{R_{short}}{R_{DS(on)} + R_{short}}\right) \tag{6.3}$$

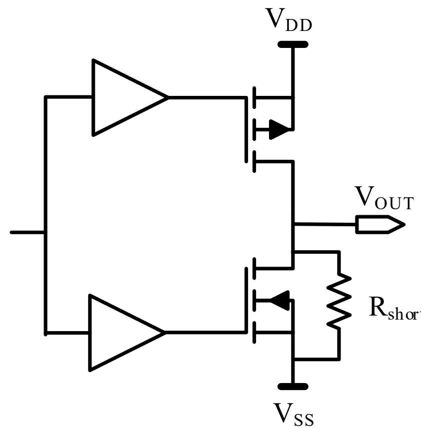


Fig. 6.3. Topology of a drain-source short.

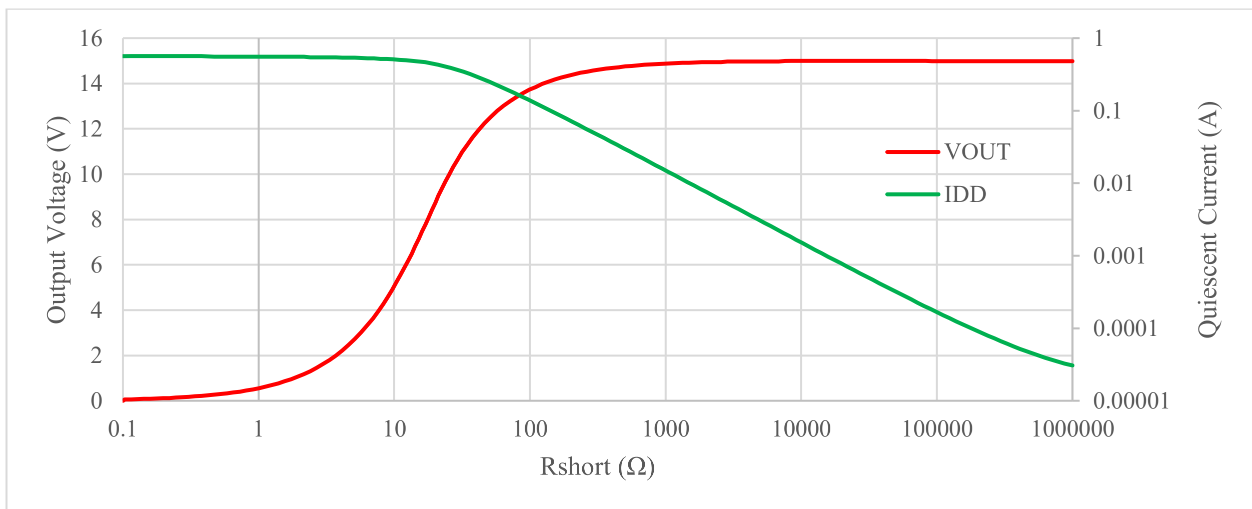


Fig. 6.4. Circuit response of an example power inverter over a range of drain-source leakage. A resistor connected to the drain and source of a NFET is varied in resistance from 0.1 Ω to 1 MΩ. The added resistance contributes to the quiescent current (green) draw up to the point where it overpowers the PFET pull-up. After that point, the output voltage (red) is affected, and current is limited by the saturation current of the PFET.

The most common fault observed was a gate-source leakage path as seen in Fig. 6.6. Fortunately, this is also the easiest failure mode to bypass. As long as the leakage resistance is significantly less than the drive strength of the driving inverter, the leakage will only contribute to an increase in quiescent current, shown in Fig. 6.5 and Eq. (6.7). As the magnitude of the gate-source leakage increases beyond the ability of the driving inverter to overcome, the drive strength of the output transistor will decrease, as shown in Eq. (6.5). Large values of leakage can result in the output transistor not turning on at all. However, this fault does not result in a malfunction while the output transistor is expected to be off. If the expected V_{GS} on the output transistor is 0 V, then the fault will not impact the system. Clearly, in a system with a single transistor with a gate-source short, the system will be compromised. However, if the output transistor is subdivided and each part is driven independently, the entire circuit can remain functional by keeping the faulty transistor off.

$$I_{DDQ} = D\left(\frac{V_{DD}}{R_{short} + R_{drive}}\right) \tag{6.4}$$

$$R_{DS(on)} = \frac{1}{k_n'\left(\frac{W}{L}\right)\left(V_{DD}\left(\frac{R_{short}}{R_{short} + R_{drive}}\right) - V_{thn}\right)} \tag{6.5}$$

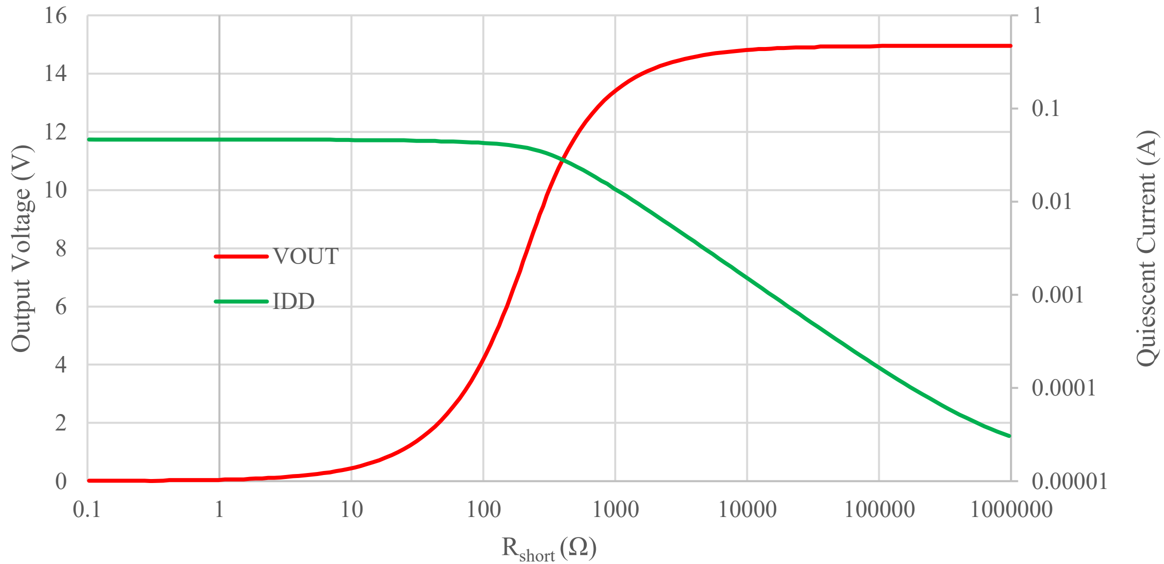


Fig. 6.5. Circuit response of an example power inverter over a range of gate-source leakage. A resistor is placed across the gate and source of the power NFET, and the value is varied from 0.1 Ω to 1 MΩ. The resistance contributes to increased quiescent current (green) of the system to the point where the resistance decreases the gate-source voltage (red). Smaller resistances result in a current limited by the saturation current of the output driver.

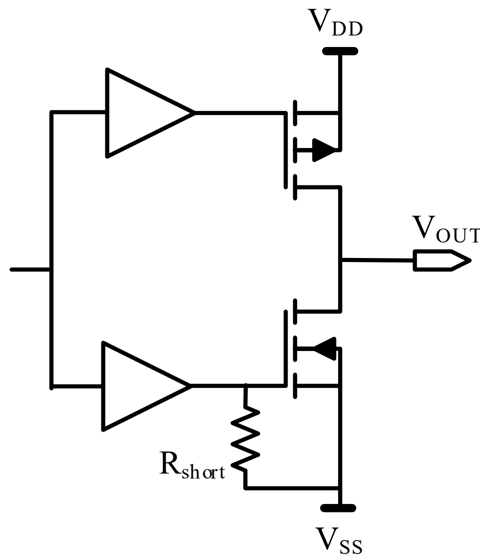


Fig. 6.6. Topology of a gate-source short.

Gate-drain shorts, shown in Fig. 6.7, also show up frequently. Inserting a resistor between the gate and drain breaks DC isolation between stages. Eliminating the isolation is an issue, since the DC voltage on the gate of a digitally switched MOSFET is typically at the opposite voltage rail as the drain. This causes the drain to constantly oppose the gate voltage on the MOSFET;

pitting the MOSFET driver against the large power MOSFET. Also, different from the other cases, this failure mode interferes with both output states of a driver. With the gate driven to the positive voltage rail, a high gate-drain resistance will show up as directly contributing to quiescent current I_{DDQ} in Eq. (6.6). Fig. 6.8 demonstrates the behavior of an output circuit suffering a gate-drain fault. As the leakage path resistance decreases to a similar magnitude as the driving inverter, the gate voltage applied to the output transistor will begin to decrease, which will increase the effective drive strength of the transistor. As the resistance of the leakage decreases further, the gate and drain will effectively have the same voltage, as the gate voltage will be drawn down to be equal to the drain voltage, reducing the drive strength. In the opposite case, where the output MOSFET is driven off (V_{GS} = 0 V), the leakage path will again contribute to quiescent current through the drive buffer. The behavior will model a simple leakage path until the gate voltage is pulled above the threshold voltage of the MOSFET. At that point, the power MOSFET will begin drawing current through the drain-source path. Higher leakage currents will dominate the drive buffer, resulting in a simplified gate-drain connected load. Both modes disrupt normal operation of the output stage, resulting in device failure when a gate-drain short is observed. This failure mode can also occur simultaneously with a gate-source short. This results in a device that is functionally very similar to a drain-source short, but also draws significant gate current. For that reason, a distinction is made between a simple gate-source short, and a gate-drain short compounded with a gate-source short.

$$I_{DDQ} = \begin{cases} \dfrac{V_{DD}}{(R_{short} + R_{drive})} & \dfrac{V_{DD}R_{drive}}{R_{short} + R_{drive}} < V_{thn} \\ \dfrac{V_{OUT}}{R_{short} + R_{drive}} + \dfrac{k_n'}{2}\left(\dfrac{W}{L}\right)\left(\dfrac{V_{OUT}R_{drive}}{R_{short} + R_{drive}} - V_{thn}\right)^2 & \dfrac{V_{DD}R_{drive}}{R_{short} + R_{drive}} \geq V_{thn} \end{cases} \quad (6.6)$$

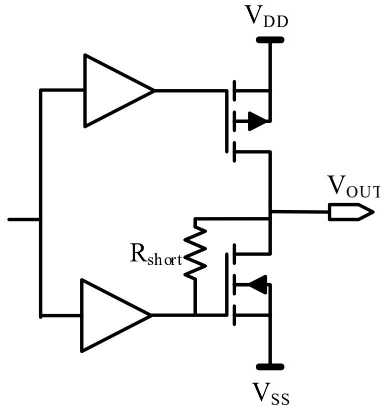


Fig. 6.7. Topology of a gate-drain short.

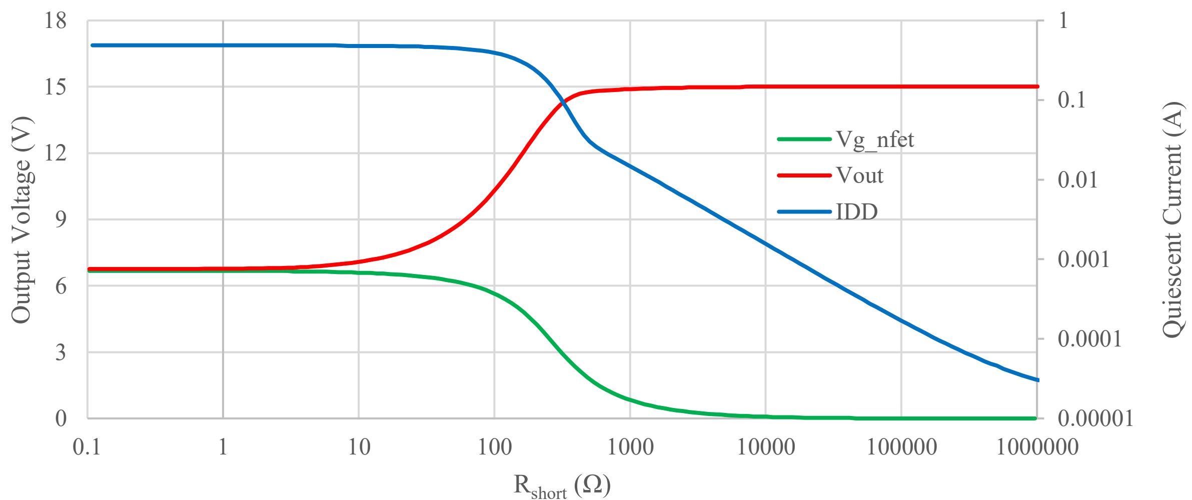


Fig. 6.8. Circuit response of an example power inverter over a range of gate-drain leakage. A resistor is placed between the gate and drain of an output NFET and the value is swept from 0.1 Ω to 1 MΩ. The NFET gate is driven to 0 V with an expected drain voltage of 15 V. Quiescent current (blue) increases with decreasing resistance to the point where the NFET gate voltage rises above V_T. On the left side, the current becomes dominated by the NFET driven into the saturation region, drawing additional current as V_{GS} (green) rises and V_{DS} (red) falls.

All faults listed will result in a leakage path that is proportional to the resistance to a point. It is important to note that below a certain threshold, each failure mode only shows up as increased quiescent current. As the leakage decreases towards a short-circuit, the current becomes limited by the saturation current of another portion of the circuit and the failure manifests itself with a change in output voltage. The thresholds and characteristics for each fault

are shown in Table 6.2. With gate-drain and gate-source leakages, the limitation is the power output devices, which can source significant current under short-circuit conditions. Additionally, these defects affect the entire device, as they connect to the drain. The gate-source shorts limit the maximum current to the strength of the driver of the output transistors, which is significantly smaller than the maximum current of the output transistors. Additionally, the gate-source shorts only affect the transistor gate it is connected to, which can be leveraged to give degraded performance instead of total die failure.

Table 6.2. Characteristics of Short-Circuit Failure Modes.

Failure Type	**10% V_{OUT} Threshold**	**I_{MAX}**	**Fights**
Drain-source	85 Ω	570 mA	Output PFET
Gate-source	1 kΩ	45 mA	NFET Driver
Gate-drain	600 Ω / 250 Ω	500 mA	NFET Driver Output PFET

In-circuit Fault Detection

Once the major faults have been identified, the focus naturally shifts to in-circuit identification and classification of fault conditions. The end goal is to allow for at least a pass-fail evaluation of a die, and optimally would allow for binning of different magnitudes of faults. If each output transistor could be individually probed without internal circuit interference, then each failure mode could easily be quantified and catalogued. For single devices, this is feasible, but can be pad intensive. Once devices are connected in parallel, or reduced to the point where additional probe pads contribute significant area, measuring individual transistors becomes difficult. Clearly, a more sophisticated system is required for an integrated system, especially for a multiple transistor system as described in chapter 4. Given an array of transistors in parallel, each with independently controlled gates, but common sources and drains, goals should be set to

evaluate how successful a testing regime is in evaluating a system. In order of importance, a testing scheme should be able to do the following:

- Generate a clear pass/fail decision on a die
- Distinctly identify each failure mode
- Identify the location of each failure
- Qualitatively evaluate the severity of the system damage
- Quantitatively measure the severity of the system damage

Meeting only the first criteria can be done simply by actively driving each transistor and observing the quiescent current of the system. As a CMOS system, very low DC quiescent currents are expected. Measuring the power supply current when driving a high and low signal will provide an opportunity for each fault to exhibit its quiescent current fault mode [58]. No additional on-chip circuitry is required, and the inputs can be driven directly. The magnitude of the quiescent current in each output state can be used as a metric for determining the health of the circuit, and a pass/fail decision can be made based on a quiescent current threshold. This method has limited ability to discern between failure modes present, and in some cases will not allow for locating the fault. The opposite approach is to leave each transistor unconnected, and probe each device individually. While this easily exposes and isolates each device for individual testing, configuring the circuit from this test configuration to an active configuration requires a large number of interconnects. Measuring each device would also require a large number of connections, and a large number of analog test circuits, and both of these items are expensive for automated testing. Electrically isolating the devices with a series transistor is not practical as the drive strength would decrease by at least a factor of four for a given area. A compromise

between the two extremes is to connect all sources and drains together, individually drive the gates, and leave the pull-up and pull-down transistors unconnected.

Targeted leakage measurements have already seen development for bidirectional digital I/O pads as well as through-silicon via (TSV) health evaluation. Concerns about leakage paths through ESD diodes or I/O transistors in pad cells led to a time-domain solution for evaluating the pad leakage [59]. This method drives the output to V_{DD} and then disables the drive to the tristate output buffer. The input buffer on the pad is monitored until the leakage causes the charge stored on the pad node to discharge below the input threshold of the buffer. For TSV testing, the concern is that a pin-hole opening in the insulation around the TSV will result in a leakage path or short circuit. The technique of charging a capacitive node to a specified voltage and sampling the output after a time period is classified as "charge and float, wait and sample", or CAF-WAS [60]. This method allows for a digital circuit implementation with adjustable thresholds.

The proposed method allows for the determination of all criteria listed above. With the pull-down NFETs and the pull-up PFETs isolated, the test and verification of each can be isolated, and the test process is essentially symmetrical. As a result, the test process will be described for an NFET array. The drivers for the NFET gates must have the ability to disable the drive (e.g., a high-impedance output) to allow the NFET gate to float. This is important to allow gate related shorts the opportunity to clearly manifest themselves. The other additional test circuit needed to run the test is a driver for the drain voltage of the NFET. Similarly, this drain driver must be able to switch between an active mode and a high-impedance mode. The gate and drain voltages will be monitored during the tests to evaluate the health of the devices.

Voltage sensing for transistor health monitoring has several important characteristics for a successful system implementation. Existing analog designs already fabricated or designed in the Raytheon HiTSiC process exhibited many limitations on a proposed voltage monitoring system. At the time of design, accurate current or voltage references were unavailable as a design element. All existing amplifiers required both voltage and current references from external sources, which would increase the pad count significantly in a test circuit. Existing amplifiers and comparators also consumed significant area of 0.2 mm^2 per OTA, which made an instance per transistor slice impractically large. With the amount of variation exhibited by the process, pure analog circuits have a high amount of risk for properties such as gain, offset, and common-mode range. Further examination of the problem shows that the number of comparison voltages is limited based on the failure modes. A heavily weighted inverter provides adequate precision for measurements while also generating a digital output signal, and requiring no additional support circuits such as reference voltages or currents. The weighted inverter topology provides the additional benefit of providing a reference threshold that is near the V_{TH} of the dominant device that should track with global process variation. The weighted inverters used in the gate driver IC are shown in Table 6.3 below.

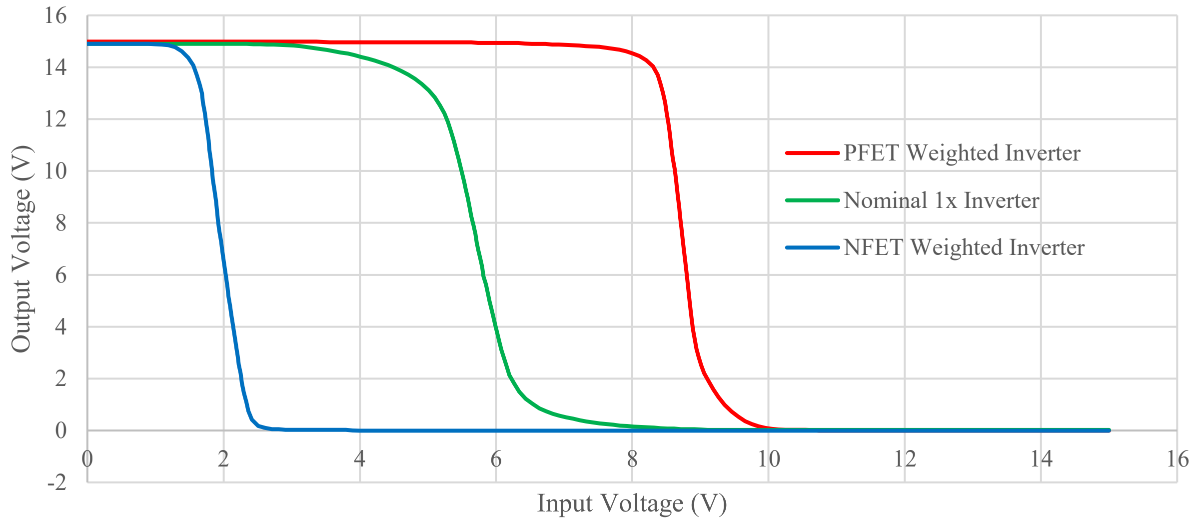


Fig. 6.9. Weighted inverter thresholds, compared to a standard 1X inverter. The PFET-weighted inverter is shown in red, and the NFET-weighted inverter is shown in blue.

Table 6.3 List of Weighted Inverter Characteristics

Type	NFET W/L	PFET W/L	V_{INL}	V_{IN50}	V_{INH}
N-Weighted	160 µm / 1.2 µm	20 µm / 5.0 µm	1.67 V	1.98 V	2.31 V
P-Weighted	Dual Cascoded 5.0 µm / 5.0 µm	160 µm / 1.2 µm	8.44 V	8.76 V	9.22 V

With the ability to selectively activate and deactivate the drive to both the gates and the drains of the pull-down NFET, transient characterization is possible for each connected transistor slice. In general, the principal is to apply test voltages to the gate and source of the output device that should be stable, assuming a functional device. The test then switches the gate and drain drive to a high-impedance state, allowing the gate and drain to float at the driven voltages. In an ideal transistor, these voltages should be maintained indefinitely, and in a compromised device the voltages will rapidly decay. The time it takes for the gate and drain voltages to decay can be measured to give an estimation of the internal leakage resistances of the device for a qualitative

evaluation of each transistor slice. Independent measurement of the threshold voltage can also be used to improve the accuracy of the measurement, allowing for a quantitative measurement of transistor health.

The drain-source short is the simplest test to measure with this configuration, though the additional leakage cannot be localized to a single device. The test configuration involves driving $|V_{GS}|$ to 0 V, and $|V_{DS}|$ to 15 V. A simple drain-source leak will cause $|V_{DS}|$ to fall from 15 V towards 0 V. An example characteristic is shown below in Fig. 6.10, where each curve represents the amount of time required for a given drain-source resistance to cause the drain voltage to fall from a $|V_{DS}|$ to below the high (red) and low (blue) thresholds of the weighted inverters. The left end of each line indicates the point where the drive circuit is unable to overcome the drain-source leakage and pull the drain voltage above the weighted-inverter threshold. On the right end of each graph, the time to switch increases roughly proportionally to the leakage resistance. The test configuration and internal circuit delays cause lower granularity in the middle and left part of each graph.

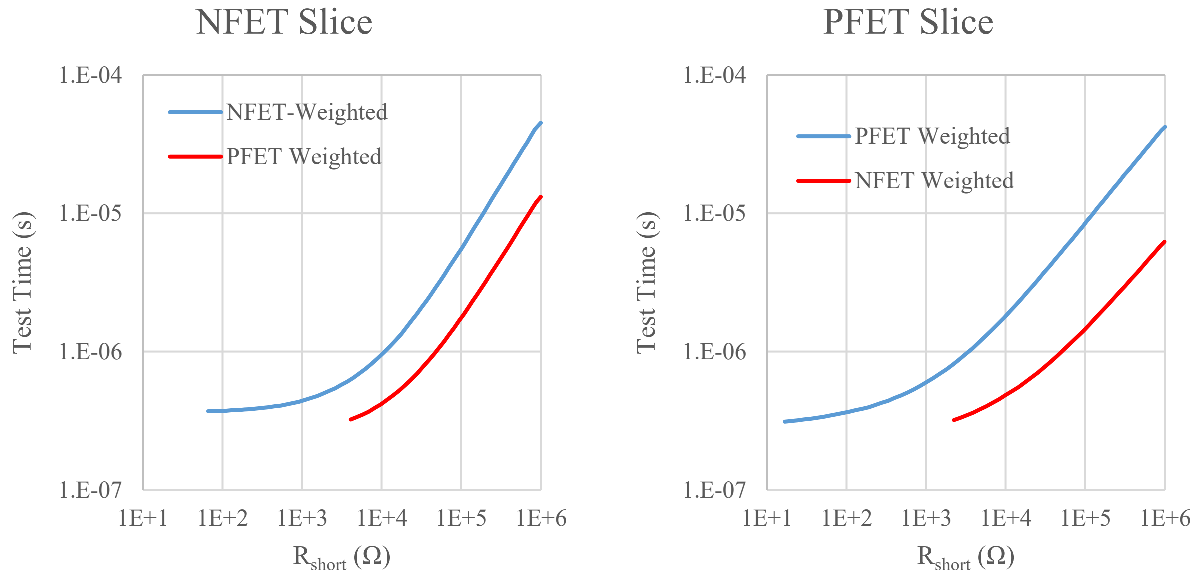


Fig. 6.10. Drain-source short characteristics for NFET (left) and PFET (right) devices. The red line indicates the larger $|V_{DS}|$ threshold from the weighted inverter pair, and the blue line indicates the lower threshold.

The detection of gate-source shorts is also fairly simple. As previously mentioned, the gate-source short can be observed by setting $|V_{GS}|$ to 15 V, and measuring the gate voltage after a period of time. In Fig. 6.11, the results of the test are given. On the left portion of each graph, lower resistances exceed the strength of the gate buffer. A failure is positively identified, but the degree of failure cannot be accurately assessed. In the bottom and curved portion of the graph, internal delays contribute to the non-linearity of the system. With gate-source shorts, all gates can be tested simultaneously, allowing for faster test times.

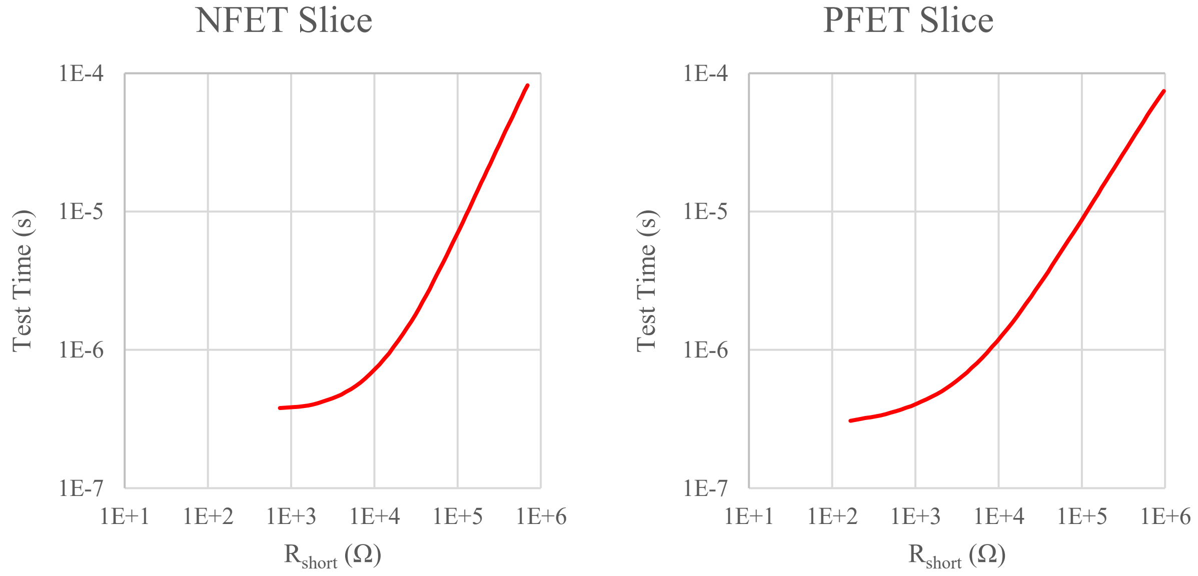


Fig. 6.11. Gate-source short characteristics for NFET (left) and PFET (right). The red curve represents the amount of time required to indicate a fault for an equivalent gate-source resistance.

Gate-drain leakage can be characterized in two methods. The first option is to configure the transistor as "on" and set $|V_{DS}|$ to 0 V, and the second option is to configure the transistor as "off" and set $|V_{DS}|$ to V_{DD}. The first method relies on measuring the gate voltage leakage through R_{GD} to ground, as other transistors in parallel will also be on. The characteristics of this fault detection method are shown in Fig. 6.12. The second method allows dual verification of the fault, from both the gate side and the drain side. Here, the weighed inverter shows off its advantage in detecting faults. For an NFET driver, the feedback inverter will be weighted to have a transition as near to the $V_{GS(th)}$ of the NFET as possible. As $|V_{GS}|$ rises from V_{DS} through R_{GD}, the gate voltage will rise past the NFET $V_{GS(th)}$. This turns on the output transistor, which depletes the stored charge C_{DS}. The timing for sensing the drain voltage is shown in Fig. 6.13. Additionally, with the NFET-weighted inverter on the gate of the MOSFET, a rise in voltage above $V_{GS(th)}$ can be detected, identifying which transistor slice contains the gate-drain short.

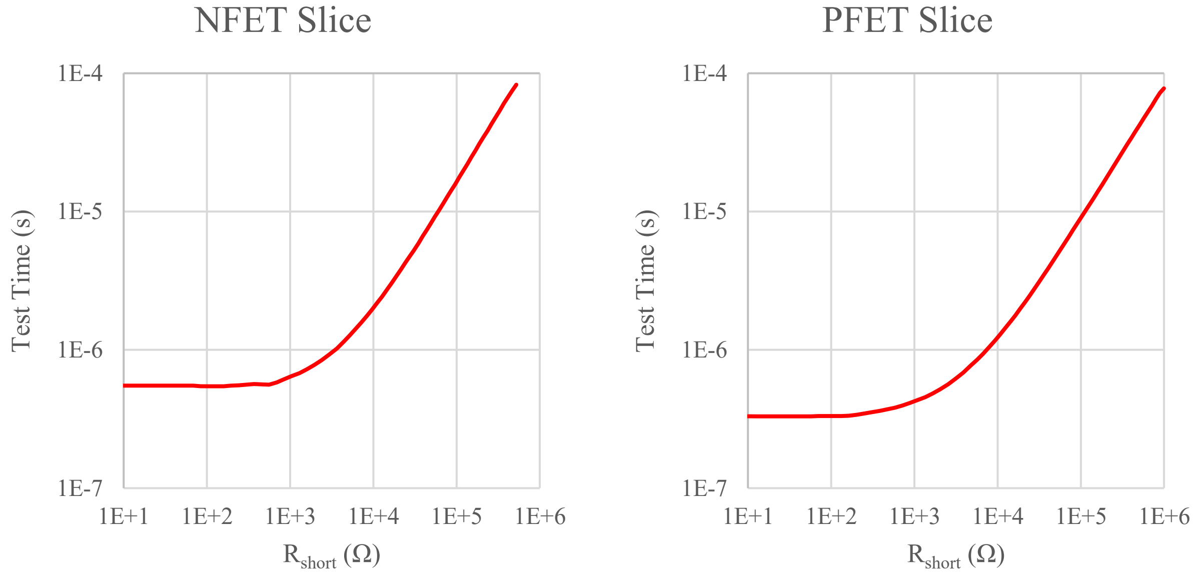


Fig. 6.12. Gate-Drain short characteristics, with an initial condition of V_{GS} = 15V, and V_{DS} = 0V. The curve indicates the time required for the gate voltage to discharge to the drain voltage through the added equivalent gate-drain resistance, as measured from the gate.

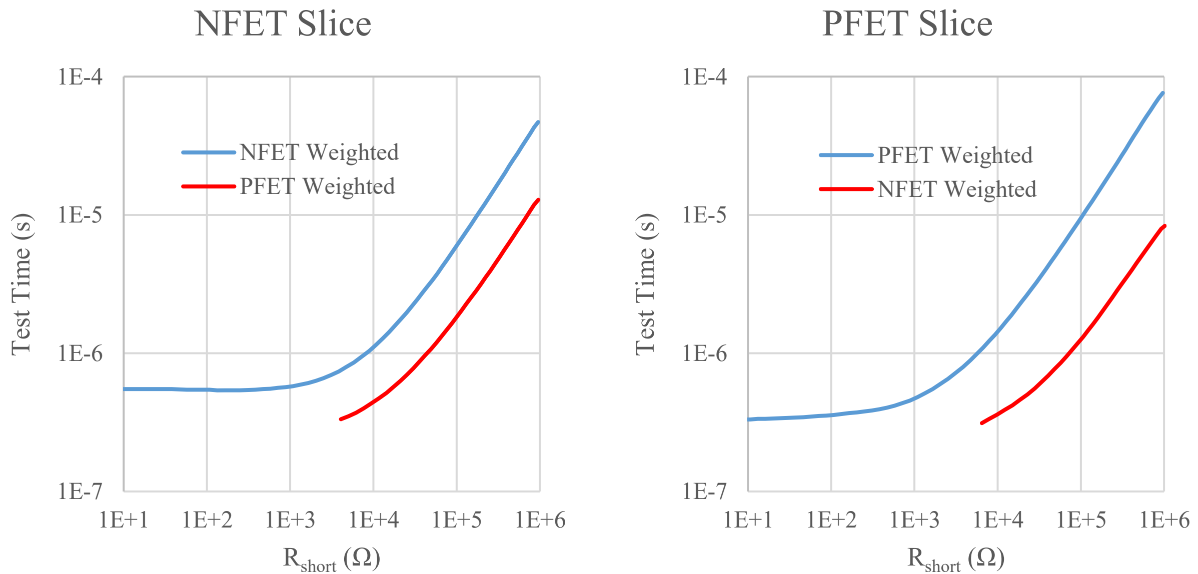


Fig. 6.13. Gate-drain short characteristics, with an initial condition of V_{GS} = 0V, and V_{DS} = 15V. The weighted inverters measure the drain voltage as it leaks through to the gate, turning on the transistor. The red line indicates the higher $|V_{DS}|$ threshold, and the blue line indicates the lower $|V_{DS}|$ threshold.

Sources of error

With the test configuration outlined, there are multiple potential sources of error in measurement towards deriving a leakage resistance measurement. With the configuration as mentioned and assuming a blind wafer with no prior measurements, there are inherent circuit parameter variations. Considering the base equation Eq. (6.9), coefficients will be added to compensate for measurement error. The effective gate capacitance of the MOSFET devices can vary due to process and lithography differences between runs. For purposes of evaluation, a scaling factor of C_{err} will be added as a proportional adjustment. Another source of variation is the MOSFET threshold voltage, which directly influences the switching thresholds of the weighted inverters. Again, a proportional factor V_{err} will be used to adjust for threshold voltage variation. The final source of potential error is a timing error t_{err}, resulting in a different sampling time than intended. Potential sources for timing error include varying internal propagation delays, flip-flop setup times, and slow/ambiguous switching of the weighted inverter. These represent an amount of time added (or subtracted) to the measurement time. The time error is considered a linear addition as the delay is relatively constant compared to the time measurement. These errors are inserted into Eq. (6.10), which is then solved for the resistance error induced by the timing error in Eq. (6.11) and the capacitance error in Eq. (6.12).

$$V = V_o e^{-\frac{t}{RC}} \tag{6.7}$$

$$t = -\ln\left(\frac{V}{V_o}\right) RC \tag{6.8}$$

$$R = \frac{t}{-\ln\left(\frac{V}{V_o}\right) C} \tag{6.9}$$

$$R = \frac{t + t_{err}}{-\ln\left(\frac{V \cdot V_{err}}{V_o}\right) C \cdot C_{err}} \quad (6.10)$$

$$R_{t_{err}} = R + \frac{t_{err}}{-\ln\left(\frac{V}{V_o}\right) C} \quad (6.11)$$

$$R_{C_{err}} = \frac{R}{C_{err}} \quad (6.12)$$

Usage

With the simulations provided, the leakage measurements are easy to perform by watching the weighted inverter outputs and measuring the time between the beginning of the test and the change of the weighted inverter for each gate. Unfortunately, this approach requires significant number output resources to support a large design. To combat this, the weighted inverters are parallel-fed into the configuration shift register. Instead of waiting for the weighted inverter to switch values, the state of every weighted inverter is simultaneously loaded into the configuration register, where it will be shifted out for analysis. The shift register readback method changes the approach to evaluating leakage, as the parallel output sensing of each transistor's leakage current is impractical. Instead, a test pattern is generated that sets a timing threshold and evaluates all transistor slices based on the fixed timing in the pattern. A passing result for each slice means that the leakage is greater than the threshold set by the pattern. A threshold based test is useful for evaluating a pass-fail metric, as a maximum total leakage current can be set as a failure threshold. This current threshold can be translated into a test time using Eq. (6.13) below based on the physical properties of the circuit.

$$t = -\ln\left(\frac{V_{DD}}{V_o}\right)\left(\frac{V_{DD}}{I_{leak}}\right) C \quad (6.13)$$

Further refinement of the current binning can be achieved with subsequent tests. Logarithmically spaced times will result in a useful grading of each individual die, and further refinement can be achieved with a successive approximation algorithm.

Improved circuit testability

Adding a test mode to the circuit improves the ability to diagnose issues, but there are inherent challenges that can make an integrated circuit easier or harder to test. As mentioned, the first prototyping run had issues with the gold finish adhesion to the base metal. The gold finish is necessary to allow for gold ball wire bonding, however the gold is very thin and is easily damaged. Using tungsten needle probes on pad surfaces resulted in scratches that made the pad unsuitable for bonding. Working under the assumption that each die required verification at probe before use, allocations needed to be made to allow for probing damage to pads. The solution used is to include pads that are dedicated to testing. With the design core-limited, extra pads were available on the chip perimeter. All functions required for testing were given dedicated sacrificial pads for probing, and the internal circuit arbitrates which pin is in use through the test-mode enable (TM_EN) signal. These pads were used as separate inputs, but alternative options include an input pad cell with two physical pads driving a single inverter.

Another part of testing complexity observed in the previous run is the large number of pad connections that were necessary in order to successfully power the circuit. The number of freely movable probes is limited in the system, and adding many probes increases the difficulty to probe a given circuit. One shortcoming of the first pass design was a lack of internal connections for power and ground nets. With only one metal routing layer in the process, a tradeoff was required between routing signals and power. With a packaged part, better power and ground connections could be made off-chip with separate pads without inserting excessive

resistance in high-current drive signals. If the circuit yield was high, this compromise would not be noticeable as connecting multiple power and ground pins on a module, package, or PCB is simple. However, the necessity to probe each pin resulted in a large number of probes required to validate functionality. To avoid this issue, split nets are prohibited in the design. Every power and ground pad is connected through metal routing to every other power and ground pad, though multiple connections will improve performance. The compromise to solid routing of power and ground on a one metal process is that many signals, including the gates for all output transistors, require a polysilicon "jumper" to make a connection.

Another technique to reduce the pad count required for at-probe testing is to combine functionality of pins. For this effort, a test-mode enable signal is derived from the VDD pin used to power the circuit. This circuit, shown in Fig. 6.14, consists of a diode and resistor to allow for the circuit to provide a logic '1' to the test mode enable logic when power is applied to the test VDD pad, and a logic '0' under normal operating conditions. This test mode enable signal allows for internal selection between the test mode pins and the normal mode pins, and allows for the system to change the fundamental behavior.

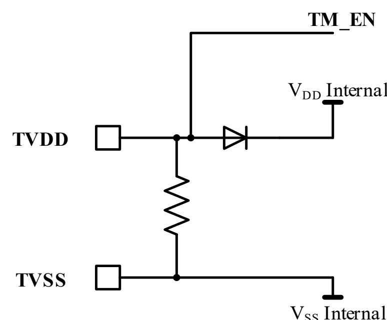


Fig. 6.14. TVDD and TM_EN pad cell schematic.

With the TVDD signal generating the TM_EN logic, the test mode pad count was minimized to six pads. In terms of probe equipment available, the next significantly useful pad count reduction is down to four. Reducing the pad count to four would allow testing on semi-

automated and automated probe stations in the laboratory, further reducing test time. However, this would require a complicated test mode state machine. The TDI and SDO pins would be combined to a single tri-state pad, and the TEN functionality would be derived from an internal state machine and the TCK pin. This added size and point of failure was undesirable in the system, and a single pad reduction to five pads did not provide a significant benefit. The final pad functionality for the system is shown below in Table 6.4.

Table 6.4 List of Test Mode Pins and Functions

Pad Name	Direction	Functionality
TVDD	Power	Supply power to circuit, drive TM_EN
TVSS	Ground	Ground
TDI	In	Serial data input
TCK	In	Serial data clock
TEN	In	Test enable – begin high-Z test
SDO	Out	Serial data output

Wafer test results

The initial die testing began after the wafers were received. An initial test program was devised to allow the execution and observation of a reconfigurable test pattern. This test program leverages a test instrument control library in Python that uses the PyVISA [61] library to directly control the function generators and oscilloscopes, and includes a simple Graphical User Interface (GUI) for improved user interaction using the PySide [62] Qt-binding libraries for Python. The interface window generated is shown below in Fig. 6.15, with most of the intended functionality implemented for the first wafer tests.

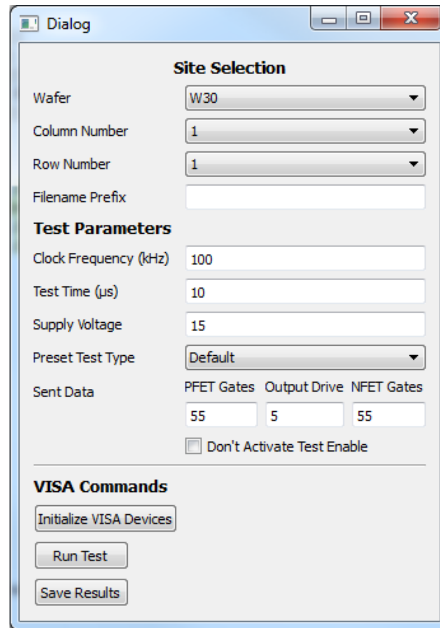


Fig. 6.15. Example test configuration GUI for wafer level gate driver verification.

Wafer 31 was selected as the first choice wafer to dice based on favorable measurements of the test structures. In anticipation of this, wafer level testing of the gate driver was performed on this wafer. Two Tektronix AWG3022B arbitrary function generators were used to generate the three input waveforms. The function generators were programmed over a VISA interface, with the TCLK and TEN pins on the first generator and TDI on the second. The TCLK and TEN signals were set to the same generator so that the high impedance time could be accurately controlled without any synchronization issues between multiple function generators. The trigger output of the first generator was connected to the input of the second generator, and the oscilloscope was set to single-trigger on the clock waveform to allow the entire test to be started by triggering the first function generator.

The shift register values were programmed in hexadecimal values representing the digital enable state of the gate. At the time the program was being written, the exact tests that would be run were unknown, so this flexibility was maintained instead of implementing the “Preset Test Type” menu. Also included in the window were wafer and site tracking information, allowing

recorded data to be easily traced to a specific site. Other parameters, such as the supply voltage, clock frequency and test time, were included in order to provide control over test conditions. The test parameters were then used to generate test vectors for both function generators, upload them to the function generators, and initiate the test by triggering the first function generator. The oscilloscope, connected to TDI, TEN, TCLK, and SDO, records the entire test. Ideally, the digital output data would be automatically decoded and presented to the user, but that was a feature skipped in the interest of time. Decoding 20 bits of serial data can be tedious on an oscilloscope, but with clever use of cursors and the relatively simple test results expected, the values can be evaluated quickly. The program button labeled "Save Results" will save the test configuration and the oscilloscope screen for later record keeping or review.

Two test patterns were used for general gate driver testing. A basic premise implemented was that the device under test (DUT) was completely functional, and further detailed tests would be run if any failure was detected in the initial tests. Towards that effort, two test patterns were used to look for gate-drain and gate-source shorts. The first test was to look for gate-drain shorts, and was implemented by setting $|V_{GS}|$ to 0 V (0x00 in the command column) and $|V_{DS}|$ to 15 V (0x1 in the output drive column). The gate voltage is expected to remain at the commanded gate voltage. The readback values are inverted, nominal logic values, so the expected result for the NFET gates is 0xFF, and the PFET gates is 0x00. Ideally, the drain-source voltage should not change, but it is only considered an issue if it drops below the second threshold. The fail mask mentioned in the tables contributes to the failure evaluation logic in Eq. (6.14).

$$Measured \oplus Passing \times Fail\ Mask \tag{6.14}$$

Table 6.5 Gate-Drain Test Results

	Command	Passing Result	Fail Mask
NFET Gates	0x00	0xFF	0xFF
Output Drive	0x1	0x3	0x6
PFET Gates	0x00	0x00	0xFF

The second test performed the gate-source short detection. By setting $|V_{GS}|$ to 15 V, and $|V_{DS}|$ to 0 V, all gate-source oxides are tested at the same time. It is important to test the gate-drain oxide first in order to avoid any false readings for the gate-source oxide. The detailed control setup is shown in Table 6.6. If both tests provide a passing result, then further investigation is not necessary and the site is marked as working. This simple pass/fail threshold was used for both tests in order to evaluate wafer 31 prior to wafer dicing. Two example test results are demonstrated in Fig. 6.16 - Fig. 6.19 showing representative test mode waveforms from the site at row 5 column 4, and row 1 column 2. The first two figures demonstrate an easily decipherable gate-drain leakage (Fig. 6.16) and gate-source leakage (Fig. 6.17) oscilloscope traces. The vertical cursors mark the boundaries of the NFET gate result, the output drive result, and the PFET gate result.

Table 6.6 Gate-Source Test Structure

	Command	Passing Result	Fail Mask
NFET Gates	0xFF	0x00	0xFF
Output Drive	0x4	0xC	0xF
PFET Gates	0xFF	0xFF	0xFF

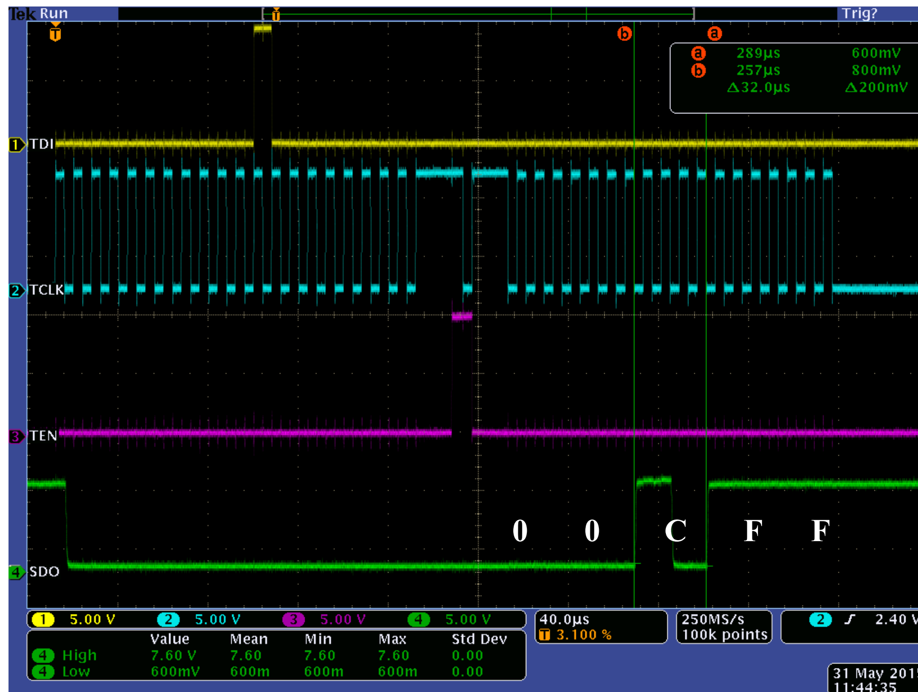


Fig. 6.16 R5C4 gate-drain Test Pass. The yellow trace is the test data input (TDI), the cyan trace is the test clock (TCLK), and the magenta trace is the test enable input (TEN). The green trace observes the serial data output (SDO) from the gate driver IC. The serial data output reads 0x00CFF, which indicates a passing result.

Fig. 6.17. R5C4 gate-source Test Pass. The yellow trace is the test data input (TDI), the cyan trace is the test clock (TCLK), and the magenta trace is the test enable input (TEN). The green trace observes the serial data output (SDO) from the gate driver IC. The serial data output reads 0xFF300, which indicates a passing result.

An example of the test failing is given by the row 1, column 2 test site. This die consumed over 300 mA at a supply voltage of 15 V without additional connections, clearly indicating a fault. Reducing the power supply to 10 V in order to sustain operations resulted in an operational test-mode shift register, allowing some diagnosis of the device. Fig. 6.18 shows a gate-drain leakage path in slice 7 of the NFET test structure. Continuing with the gate-source test in Fig. 6.19, slice 6 of the PFET side shows a fault. While these two observed faults may not be the sole contribution to the large power supply quiescent current, the information is useful for wafer-wide statistical comparison of power transistor gate oxide health.

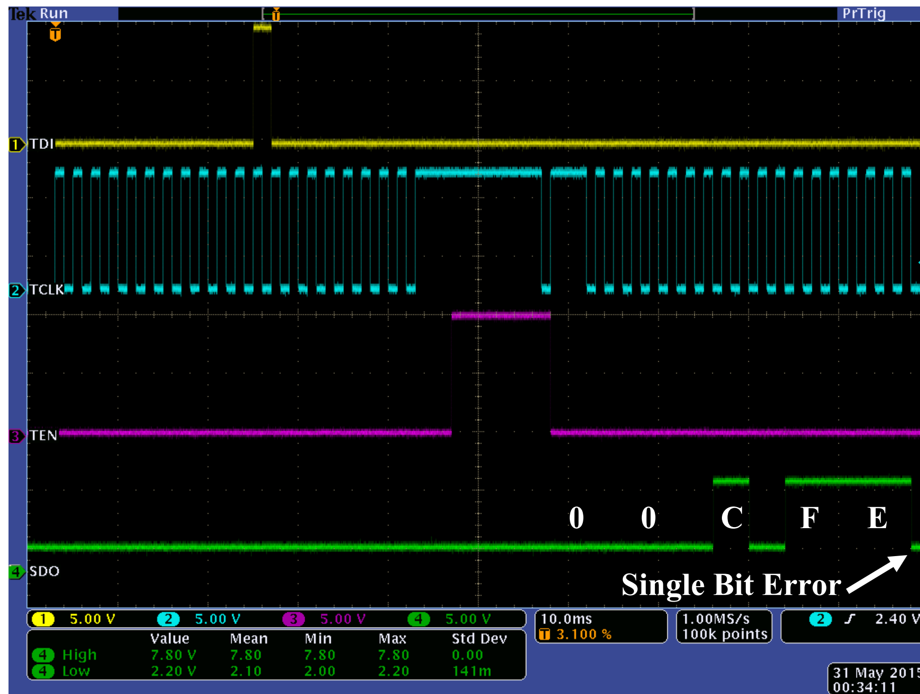


Fig. 6.18 R1C2 gate-drain Test with fault in 19th bit. The yellow trace is the test data input (TDI), the cyan trace is the test clock (TCLK), and the magenta trace is the test enable input (TEN). The green trace observes the serial data output (SDO) from the gate driver IC. The serial data output reads 0x00CFE, which indicates a fault (expected 0x00CFF).

Fig. 6.19 R1C2 gate-source test, with a fault in 6th bit. The yellow trace is the test data input (TDI), the cyan trace is the test clock (TCLK), and the magenta trace is the test enable input (TEN). The green trace observes the serial data output (SDO) from the gate driver IC. The serial data output reads 0xFD300, which indicates a fault (expected 0xFF300).

Considering the entire wafer, there are 26 instances of the 20 mm x 12.5 mm reticle, with several partial impressions. Due to the lower right corner location of the gate driver sub-site, there are only 22 complete drivers. An additional partial sub-site, row 6 column 4, was complete enough for testing and verification, but lacked the final passivation and pad metallization necessary for a usable die. This partial site is included in the statistical analysis as it tested functional, for a total of 23 test sites. A grid numbering shorthand was used to identify sub-sites based on their location on the wafer, abbreviating “row” as R, and “column” as C. For example, the previously mentioned row 6 column 4 site is abbreviated as R6C4. Future die locations will be given using this shorthand. Out of all 23 sub-sites, only one site failed with no functionality, at R2C1. This die had high quiescent current and no response from the test mode interface, in addition to visible surface defects on the die. While the die was faulty, no specific information could be identified regarding the output transistor health, so it was omitted from per-transistor

statistics. In all other cases, the test mode circuitry was able to provide health information on the output transistors. Examining a wafer map of the faults in Table 6.7, three failures were on the bottom edge of the wafer, while the other two were not isolated to a specific area. Failures by type are given in Table 6.8. The test results for the wafer are broken down in Table 6.9, showing the rates of failures for wafer 31. The total number of die evaluated as passing electrical tests was 77.3 %. After further examination of the faults by transistor slice, 1.1 % of devices showed faults resulting from gate oxide defects in the power transistors.

Table 6.7 Wafer 31 Test Results

		Column			
		1	2	3	4
Row	7		OK	OK	
	6	OK	OK	1 GD NFET	OK, Partial
	5	OK	OK	OK	OK
	4	OK	OK	OK	OK
	3	OK	OK	Moderate Drain Leakage	OK
	2	Visible Surface Irregularities, High Current, No Response	OK	OK	
	1		High current, 1 PFET GS, 1 NFET GD	1 GS NFET	

Table 6.8 Breakdown of Faults by Site Coordinates

Test Mode Bad	**High IDDQ**	**Gate-Source**	**Gate-Drain**	**Drain-Source**
R2C1	R1C2 R2C1	R1C2 R1C3	R1C2 R6C3	R3C3

Table 6.9 Wafer 31 Probe Test Statistics

	Type	Fraction affected	Percentage
Gate Driver Sites	**Die successfully screened**	23/23	100%
	Die evaluated by test mode	22/23	95.7%
	Known good die	17/22	77.3%
	High IDDQ	2/23	8.7%
	Drain-source	1/23	4.4 %
Individual Transistors	**Gate-drain**	2/352	0.6 %
	Gate-source	2/352	0.6 %
	Healthy gate oxides	348/352	98.9 %
	Unevaluated Transistors	16/368	4.3%

Conclusions

A design for test method was implemented in the gate driver IC for the purpose of evaluating the health of the gate driver at probe prior to packaging. The previous run indicated a yield issue with the power transistors, which is also the largest active area. Additional logic and low-risk circuitry added a small overhead to the existing circuit area due to existing features in the logic. A leakage quantification method was implemented to evaluate the health of the output transistor gate oxide. The area overhead was relatively small, though a penalty in propagation delay was observed. The DFT method was used to successfully evaluate the health of an entire wafer at probe. The observed yields improved greatly over the previous run, validating cautious design decisions.

Lessons Learned

The necessity for designing for testability is always present. Large structures present possibilities for failure that must be anticipated. Designing with experimental processes also exposes additional risk to immature design rules. The more complicated the behavior of the device, the more sophisticated the testing capabilities must be in order to allow for successful identification of known-good parts and individual failure mechanisms.

7 Testing

Once the gate driver die are validated as "known good", then the evaluation of the gate driver performance and features can begin in earnest. The high current and high speed nature of the gate driver design preclude any useful performance evaluation in a probe station. Two main test cases are investigated: high temperature operation, and operation inside a power module. As part of the larger NSF-BIC project, group testing fixtures were used when possible, although limitations in the existing setup required significant enhancement for testing full gate driver functionality. For the high-temperature testing, the gate driver was integrated into existing high temperature test fixtures. These fixtures were validated as a method for accurately supplying and controlling die temperature, and then tested over various power MOSFET loads and temperatures. Power module integration was performed in a custom-designed layout with silicon carbide power devices operating under load. The results for each section will be presented and evaluated.

High temperature test design

Once the design was validated at probe, a wide temperature range evaluation of the design was required. A constant issue faced during the testing is the lack of techniques readily available for high temperature testing. Common passive components lack high temperature ratings, with capacitors approaching their limits at 225 °C. Die attach materials are limited past 300 °C, and no solders are rated beyond 320 °C. High temperature circuit boards are limited to ceramic-based designs of LTCC, DBC, with other traditional materials such as FR4 and Rogers 4350 rated for limited exposure to 260 °C. As such, many of the testing design decisions revolve around abusing materials beyond the rated temperatures to achieve a quick measurement at temperature.

Prior experiences with high temperature electronics testing [15], [22] started with some general strategies that worked well for short periods of time. This used a low temperature FR4 PCB with circuit loads, a high temperature daughterboard made of Rogers 4350 with the bare die under test mounted directly to the board. A copper "hot finger" was mounted to the bottom of the board, with a thermocouple mounted between the daughterboard and the hot finger for temperature measurements. The hot finger assembly was placed on a hot plate, allowing for the high temperatures of the hot plate to be selectively directed towards the DUT with minimal heat directed to the low temperature supporting circuitry. Due to the nature of the circuit tested in this program, the DUT generated significant heat due to the high power consumption, reducing the power input needed from the hot plate. This setup allowed measurement of DUT performance above 400 °C.

Initial high temperature measurements began with a similar system custom designed to be generic for high temperature to allow for many devices fabricated to be tested at high temperature without significant test cost. A similar hot finger design was used, but a QFP 64-pin package was used to contact the hot finger instead of directly making contact with the daughterboard. The SiC die was attached to the QFP package using Epotek P1011 conductive epoxy, and the die was wire-bonded using 1 mil gold ball bonding. The QFP 64 package had a large, unformed leadframe that was left intact in order to allow the package to be suspended inside a cutout of the daugherboard. The daugherboard had a hole cut out of the center to allow the QFP package to be suspended, and was made out of Rogers 4350. The QFP package was soldered to the daugherboard using a 95%/5% lead/tin solder with a solidus temperature of 305 °C. An aluminum hot finger with a position for a thermocouple was machined to fit, and pin headers were used to connect wires and boards to the daughterboard.

A modified test setup was developed to allow for higher temperature testing with higher reliability. The previous QFP64 package was exchanged for a LDCC68 package primarily for logistical reasons. One tangential benefit of the LDCC package was that the package leads were recessed from the top of the package, providing protection to wirebonds. Additionally, the leads were brazed onto the ceramic package instead of being suspended in glass frit, eliminating one of the weak spots in the test setup. Another issue observed was that at high temperatures, the hot finger made out of aluminum began to soften. Unlike the ARPA-E test setup used in a previous project, the hot plate set point needed to be higher than the target temperature, meaning that the hot plate surface exceeded 650 ℃ at the highest temperatures. While all other properties of the aluminum hot finger were satisfactory, the low melting point of the aluminum provided a source of concern for the safety of the test setup. A copper based hot finger was machined in order to eliminate the concern of melting at high temperature. A summary of the testing efforts is shown below in Table 7.1, indicating the packaging used, and the testing outcome for each die.

Table 7.1 List of packaged gate drive die with test results

Site	Package	Status	Temperature Tested	Voltages Tested
R6C1	LDCC68	Success	25 ℃ - 528 ℃	12 V, 15 V
R4C2	LDCC68	Failure	25 ℃	
R7C3	QFP64	Failure	25 ℃	
R2C3	QFP64	Success	21 ℃ - 427 ℃	15 V
R3C4	LDCC68	Success	25 ℃ - 530 ℃	12 V
R3C1	Power Module	Success	25 ℃	15 V

High temperature setup evaluation

One concern with the hot finger test setup was the accuracy of the temperature measurement provided by the thermocouple placed on the top of the hot finger. At the temperatures involved, estimating temperature gradients is difficult at best. A strategy was devised to measure how closely the thermocouple temperature measured at the bottom of the DUT package tracks the actual die temperature. A US Sensors model PPG102A6 1 kΩ platinum resistance temperature detector (RTD) was identified as a potential temperature sensor, with an accuracy of ± 0.15 °C and a high temperature rating of 600 °C. The temperature sensor is small relative to the gate driver die, at a size of 1.6mm × 1.2 mm, and had wire leads for connection. A blank 5mm × 5mm SiC die was attached to the cavity of a LDCC 68 using the same conductive epoxy used for die attachment. The temperature sensor was attached to the blank die in the center with conductive epoxy for good thermal transfer. The two RTD leads were bent to cover multiple pins. The multiple pin contacts allow for 4-wire resistance measurement allowing for minimal error from inconsistent contact and lead resistance. The LDCC 68 package was then soldered to a daughterboard as with the other test configurations. The entire configuration was designed to show optimal physical matching between the test module and the die used in the test fixture. The final setup can be seen in Fig. 7.1 after testing and after being removed from the Rogers daughterboard.

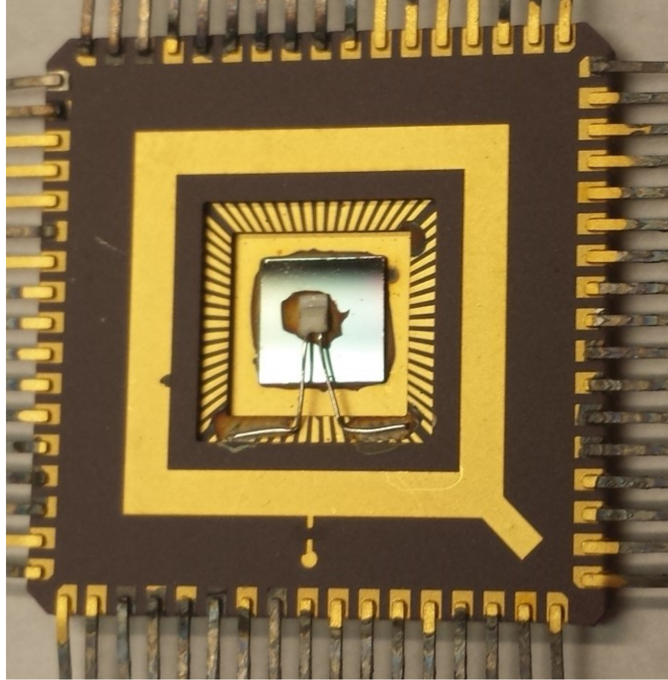


Fig. 7.1. RTD sensor configuration for hot finger thermocouple temperature correlation testing. The LDCC 68 package is identical to other packages used for testing. A blank SiC die was used to simulate the circuit under test, and the platinum RTD was epoxied to the top of the die. The RTD leads span multiple pins to allow for 4-wire Kelvin connections. Discolorations are due to the solder and high temperatures used during testing.

Both the old aluminum and the new copper hot finger were evaluated using the RTD fixture. The aluminum setup was measured first, with the RTD connected to a Keithley 2602 Source Measure Unit (SMU) as a constant current source of 1 mA for the 4-wire resistance measurement. The thermocouple was connected to the hot plate sensor input, and was used as a control feedback for the hot plate. Due to the thermal behavior of the system, this configuration leads to temperature overshoot as the hot plate internal PID algorithm is not tuned for significant decoupling between the heat source and the thermocouple. Temperatures and resistances were logged over time, and the temperature set point was increased in 100 °C increments as is shown in Fig. 7.2. The experiment ran until the RTD resistance measurement became intermittent and failed, after about 35 minutes. For the second experiment, a logging system was configured to increase logging frequency. The thermocouple was connected to a voltmeter in order to automate

readings. As a result, the hot plate ran open-loop, without feedback to correct the hot plate temperature to the commanded temperature, as is shown in Fig. 7.3. The test with the copper hot finger lasted longer, almost one hour and ten minutes, with the test ending after the wire insulation leading to the daughterboard melted and shorted the RTD terminals. The copper test fixture resulted in multiple discontinuities in the measurement data, side effects of intermittent connections.

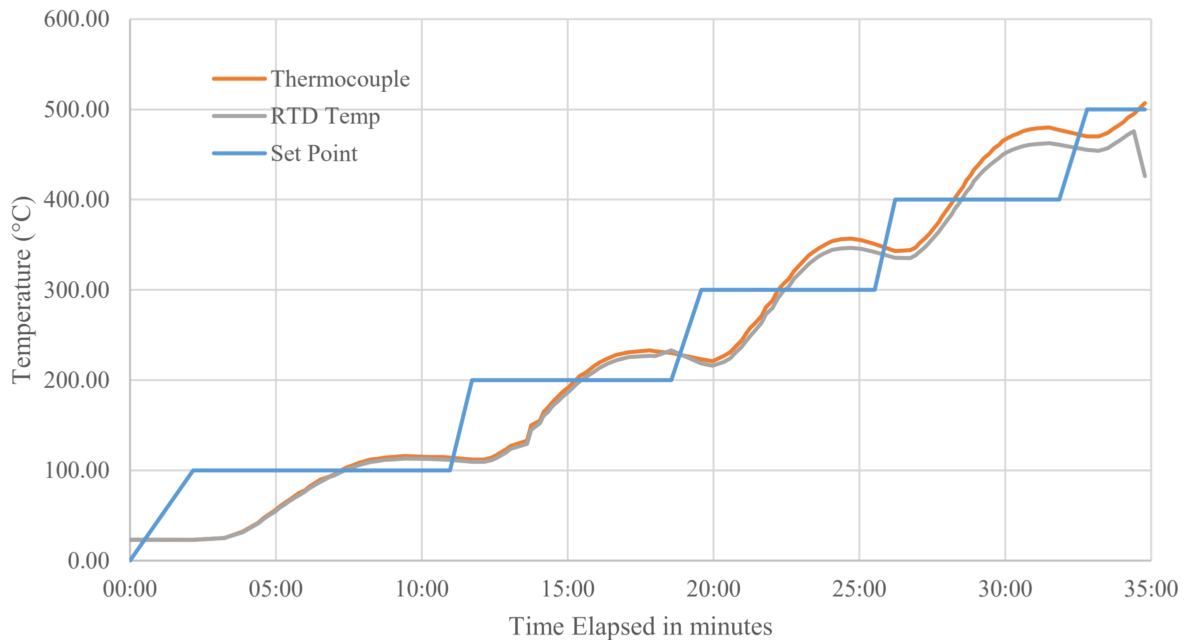


Fig. 7.2 Aluminum hot finger temperature calibration results. Hot plate used the thermocouple as feedback, resulting in temperature overshoot.

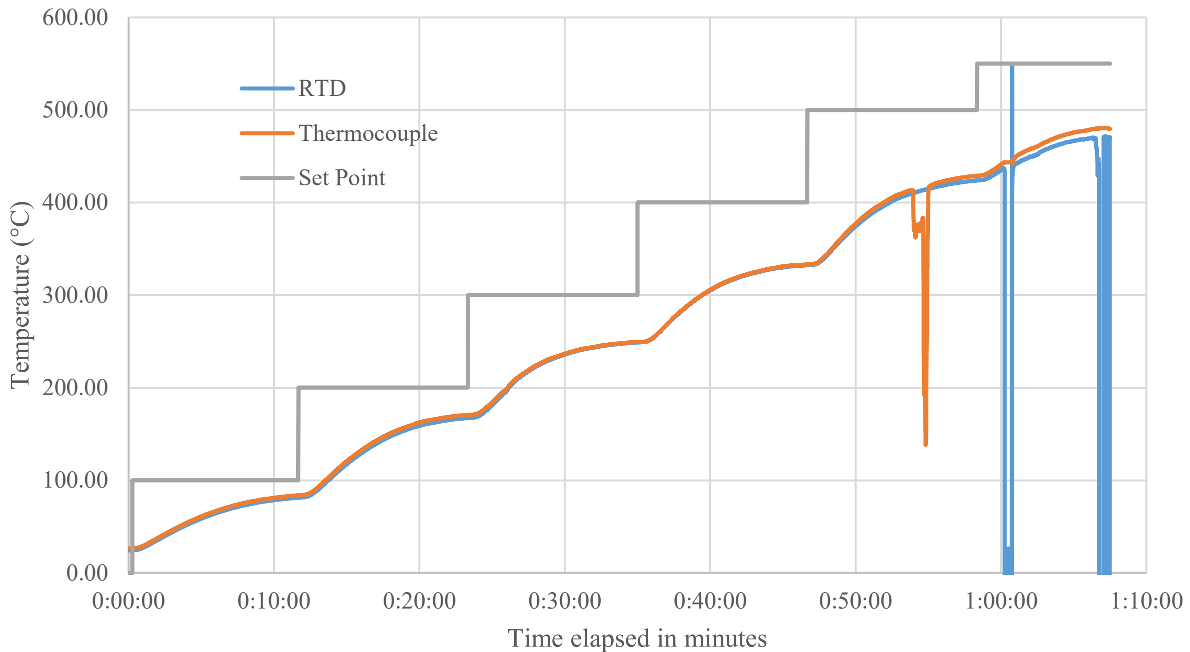


Fig. 7.3. Copper hot finger calibration results. Hot plate was controlled without feedback, resulting in the underdamped temperature response. Negative discontinuities were a result of intermittent thermocouple contact at 55 minute, and RTD lead shorting at 1:00:00 and 1:07:00.

The wide temperature characterization of both test fixtures are done in order to identify potential issues and discrepancies between the easy to measure thermocouple and the actual die temperature. While the discrepancy should be small, without characterizing it the reported temperatures will not have a good sense of accuracy. In Fig. 7.4, both the copper and aluminum hot finger performance over temperature are plotted. The major discontinuities in temperature caused by intermittent continuity issues were removed, and the figure shows how much hotter the thermocouple reading is compared to the RTD measurement. Two interesting features emerge immediately, in that the copper configuration shows a lower error over temperature, and the copper shows lower variation at each temperature. The better absolute value can be attributed to the higher thermal conductivity of copper (385 W/m·K) compared to aluminum (237 W/m·K) [63] in combination with the larger surface area of the copper finger design (361 mm^2) over the aluminum design (127 mm^2). The other issue identified with the aluminum hot finger data is that

the variation at a given temperature is larger than the corresponding copper data. Examining the transient data from Fig. 7.2, after the temperature overshoots the set point, the hot plate is turned "off" to reduce the temperature. The cooling that results shows poorer correlation between the thermocouple and RTD than when the hot plate is actively heating the system. No overshoot is present in the copper test setup as the hot plate regulated the surface temperature to the set point, and as such the heat transfer is always consistent.

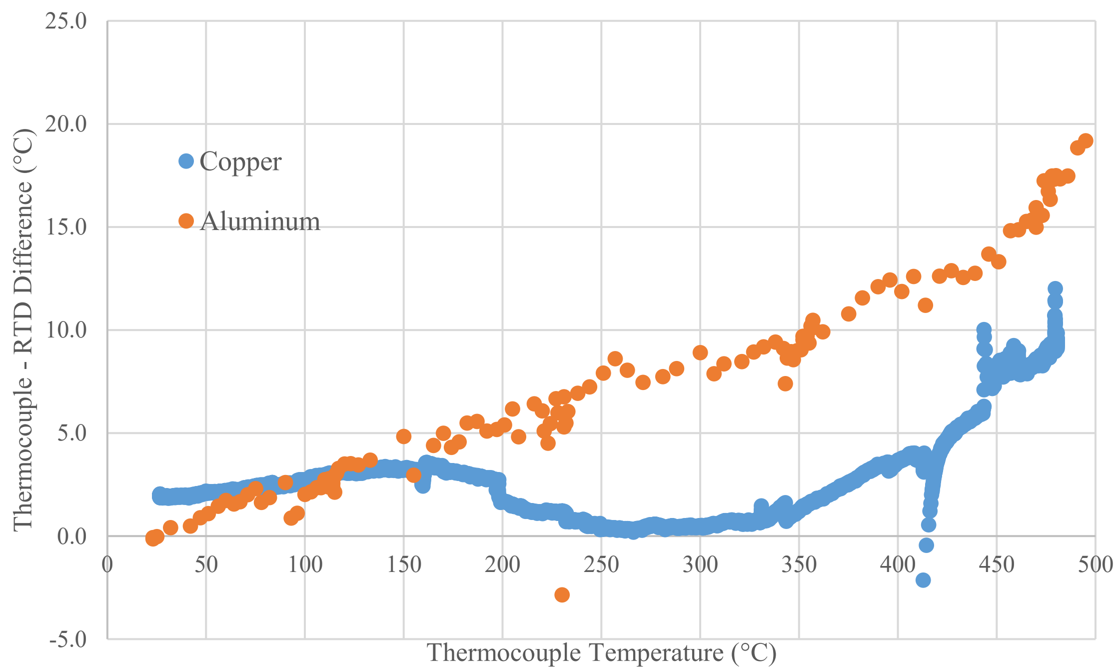


Fig. 7.4. Temperature deviation between the thermocouple at the bottom of the LDCC 68 package and the top surface of the SiC die measured with the RTD. Measurement issues caused large discontinuities at points, major outliers have been removed.

Test Setup Configuration

An FPGA with a level shifter on the output was used as an intermediary between a function generator and the gate driver under test. While the simple drive mode does not require any advanced processing, programming the shift register and two level drive strength modes require additional sophistication. Again, an additional function generator is a possible option, but

programming the shift register would require real-time modification of the arbitrary waveform of the function generator, which would require significant programming effort. Instead, a Terasic DE2 board with an Altera FPGA was selected to run the control logic, and a function generator to create the gate driver input signal. The gate driver input is controlled by the function generator as the interface for adjusting and setting pulses is significantly better than what could be developed quickly on the FPGA board. For the configuration register, there are 16 bits that control the drive strength enable signals, and an additional 2 bits that control the test mode settings. The DE2 board includes an array of 18 slide switches connected to the FPGA, allowing for a 1 to 1 mapping of control registers to switches. Momentary push buttons on the board are used to shift out data and reset the internal FPGA state machine. A CMOS level shifter, CD4504, is used to convert the 3.3 V logic outputs of the FPGA to the 15 V logic inputs of the gate driver. The output of the level shifter is used as the timing reference for all propagation delay measurements.

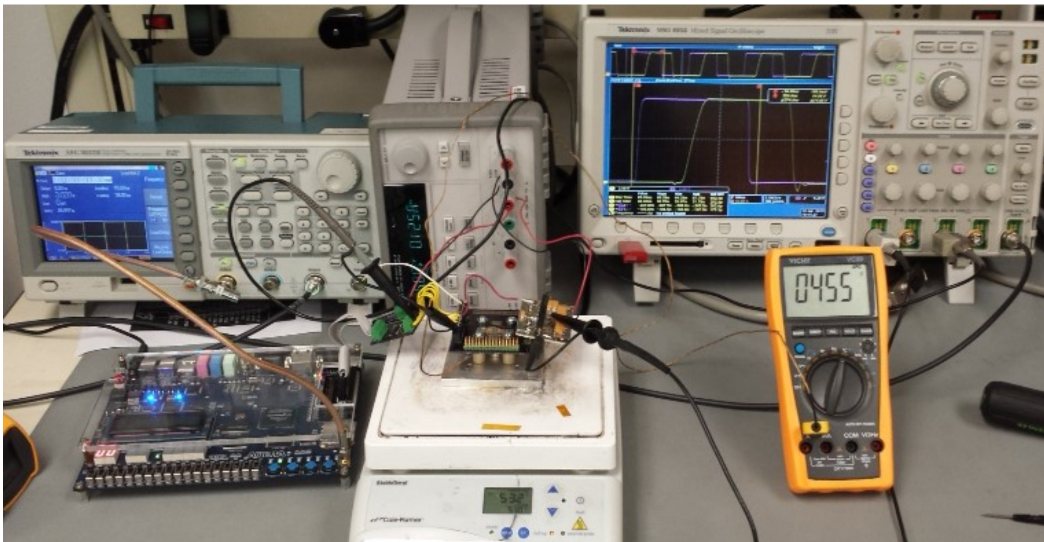


Fig. 7.5. Hot plate testing using the QFP64 test configuration with the gate driver loaded with a C2M0280120D power MOSFET. Measured temperature at the bottom of the package is 455 °C, with an input frequency of 1.0 MHz.

Table 7.2 Summary of High Temperature Testing

Wafer	31	31	31
Site	R2C3	R3C4	R6C1
Data	HP Testing 7-13-15	Gd_temp_meas_9-4	Gd_temp_meas_9-4
Package	QFP64	LDCC68	LDCC68
Hot Finger	Aluminum	Copper	Copper
Load MOSFET	C2M0280120D	C3M0065090D	C3M0065090D
Load C_{ISS}	259 pF	660 pF	660 pF
Max Temperature	450 ℃	530 ℃	528 ℃
V_{DD} Tested	15 V	12 V	12 V, 15 V

Two different high temperature test configurations were used to test the gate drivers over temperature, and are distinguished by which package was used for each die, summarized in Table 7.2. A switch from the QFP64 package to the LDCC68 package resulted in a redesign of the high temperature test board used to augment the device testing. This board contained power supply decoupling capacitors, a 0.2 Ω gate resistor for current measurement, as well as a power MOSFET load. Testing goals also differed between the two test setups as the tests were performed at a different time.

The first test setup focused on observing the functionality over temperature and demonstrating the maximum and minimum drive strength. The load selected was the Cree C2M0280120D, a 1200 V, 280 mΩ power MOSFET. This device was relatively small, resulting in minimal rise and fall time increases over an unloaded driver at room temperature. The driver was tested over temperature using the aluminum hot finger up to a temperature of 454 ℃. Two different conditions were tested with a V_{DD} of 15 V: full drive strength, and minimum drive strength. This corresponds to all transistor slices active, and only one transistor slice active. The 10 % to 90 % transition times for the full drive strength case are given in Fig. 7.6, and the minimum drive strength is given in Fig. 7.7. The full drive strength rise time contains a minima

at 150 °C, with performance degrading at either higher or lower temperatures. The full strength fall time conversely has a local maxima at 150 °C, which then decreases to a nominal value at 200 °C. Mild degradation in fall time is seen above 200 °C. Comparing the full drive strength to the minimum drive strength in Fig. 7.7 shows a dramatic decrease in rise time between 20 °C and 150 °C, and modest increases above 225 °C. Fall time in the minimum drive strength condition remains stable, with local maxima at 150 °C and above 375 °C. Comparing the two conditions when normalizing the rise and fall times, the broader temperature results can be seen in Fig. 7.8. With all rise and fall times set to a value of one at room temperature, the fall times show degradation over temperature, but the magnitude of the degradation is limited. The rise time shows improvement at increased temperatures. This difference in behavior can be explained by the dimensions of the PFET and NFET devices. As the NFET transistor slice losses are metal dominated, the temperature behavior is relatively flat and shows little improvement over temperature. However, the PFET losses are transistor dominated, and improvement in transistor drive strength over temperature result in improved transistor drive strength and a corresponding improvement in performance. Propagation delay was also measured and plotted in Fig. 7.9. The propagation delay decreased over temperature to a minimum at 250 °C, and then slowly increased as temperature increased. The total decrease in propagation delay is likely a function of the control and driving logic increasing in strength with temperature and thus decreasing the propagation delay.

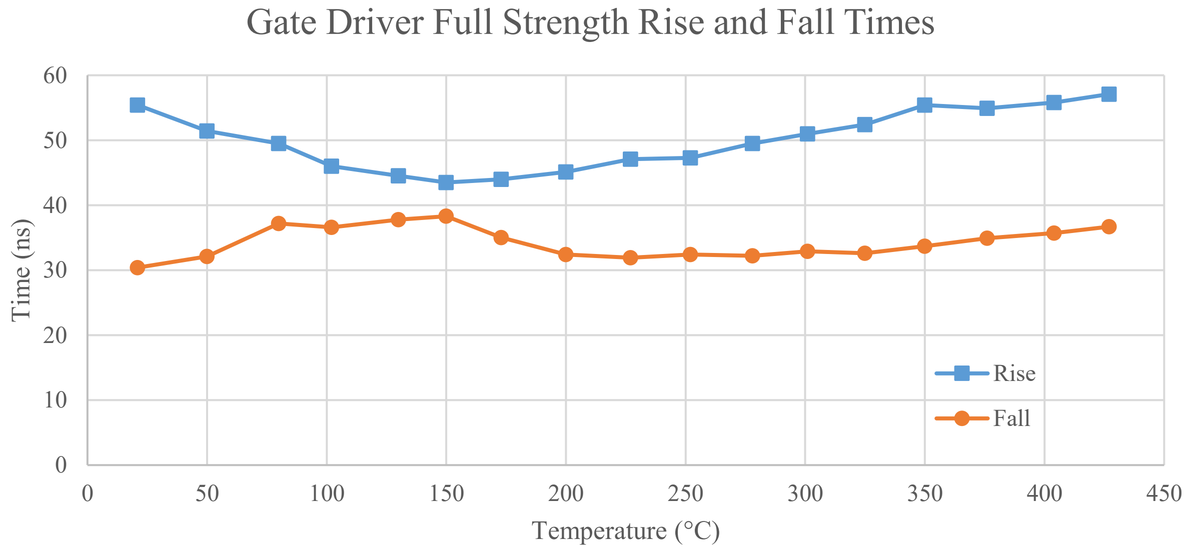


Fig. 7.6 Rise and fall times driving a C2M0280120D MOSFET at 15 V by W31 R3C2.

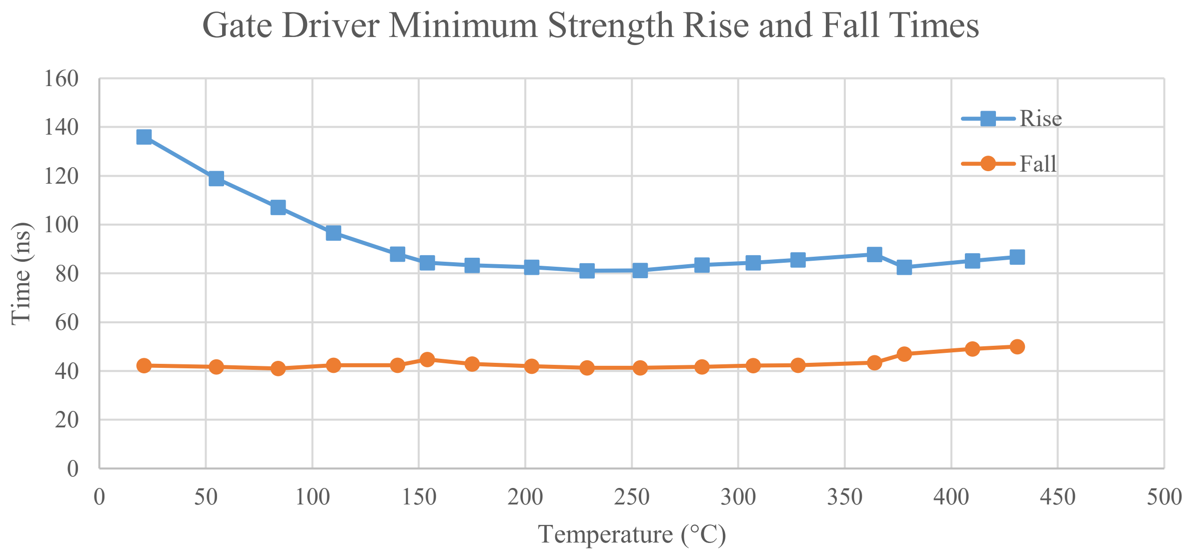


Fig. 7.7 Rise and fall times driving C2M0280120D at minimum drive strength by W31 R3C2.

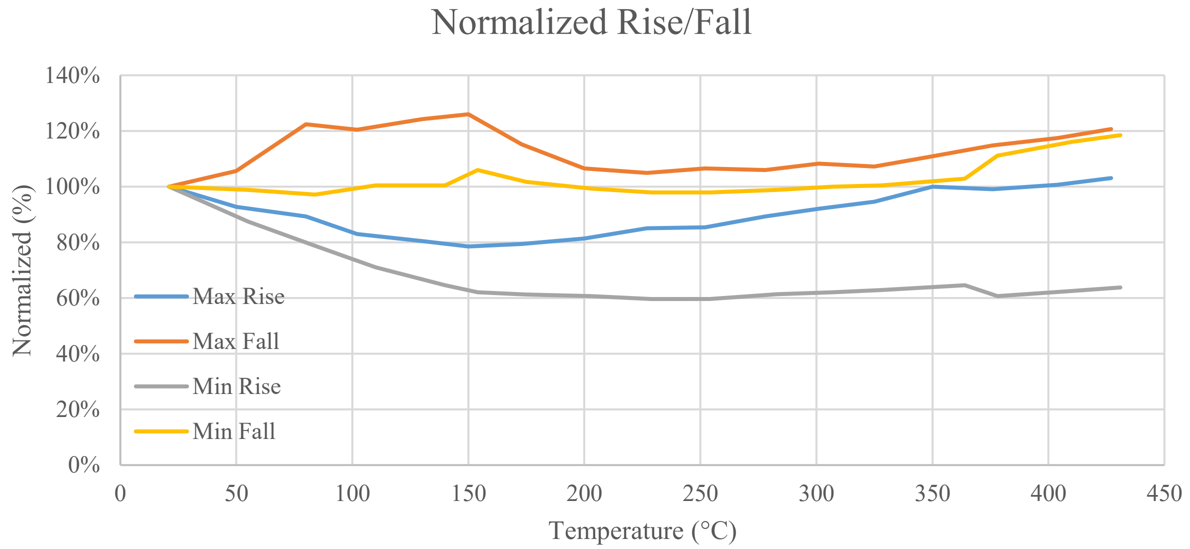


Fig. 7.8 Comparison of rise and fall time variation over temperature while driving a C2M0280120D MOSFET by W31 R3C2. Times are from the minimum and maximum drive strengths, and are normalized to the value at 25 °C.

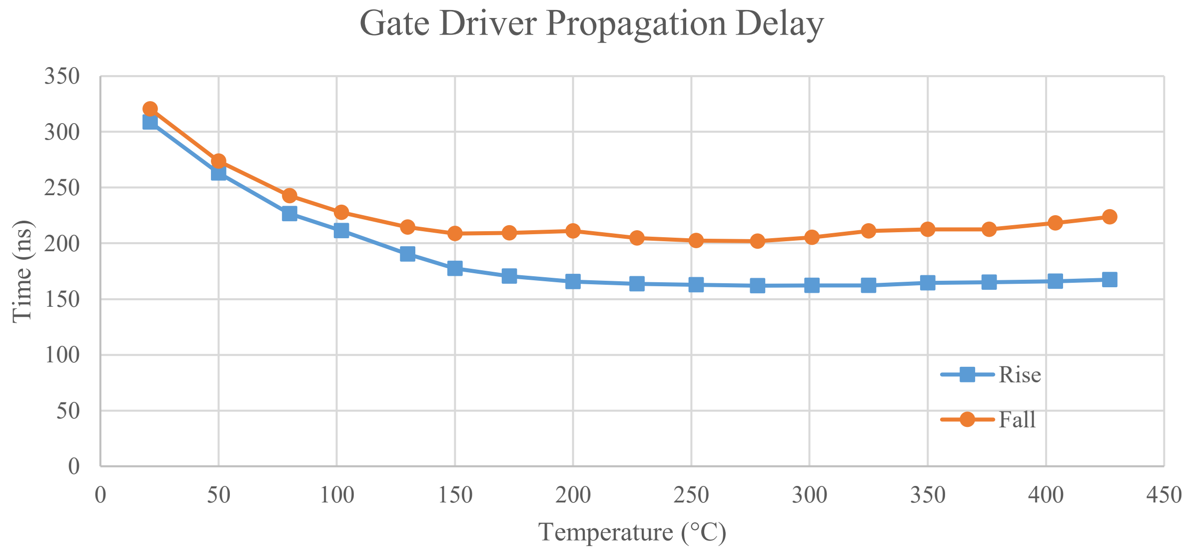


Fig. 7.9 Propagation delay over temperature driving a C2M0280120D at 15 V by W31 R3C2.

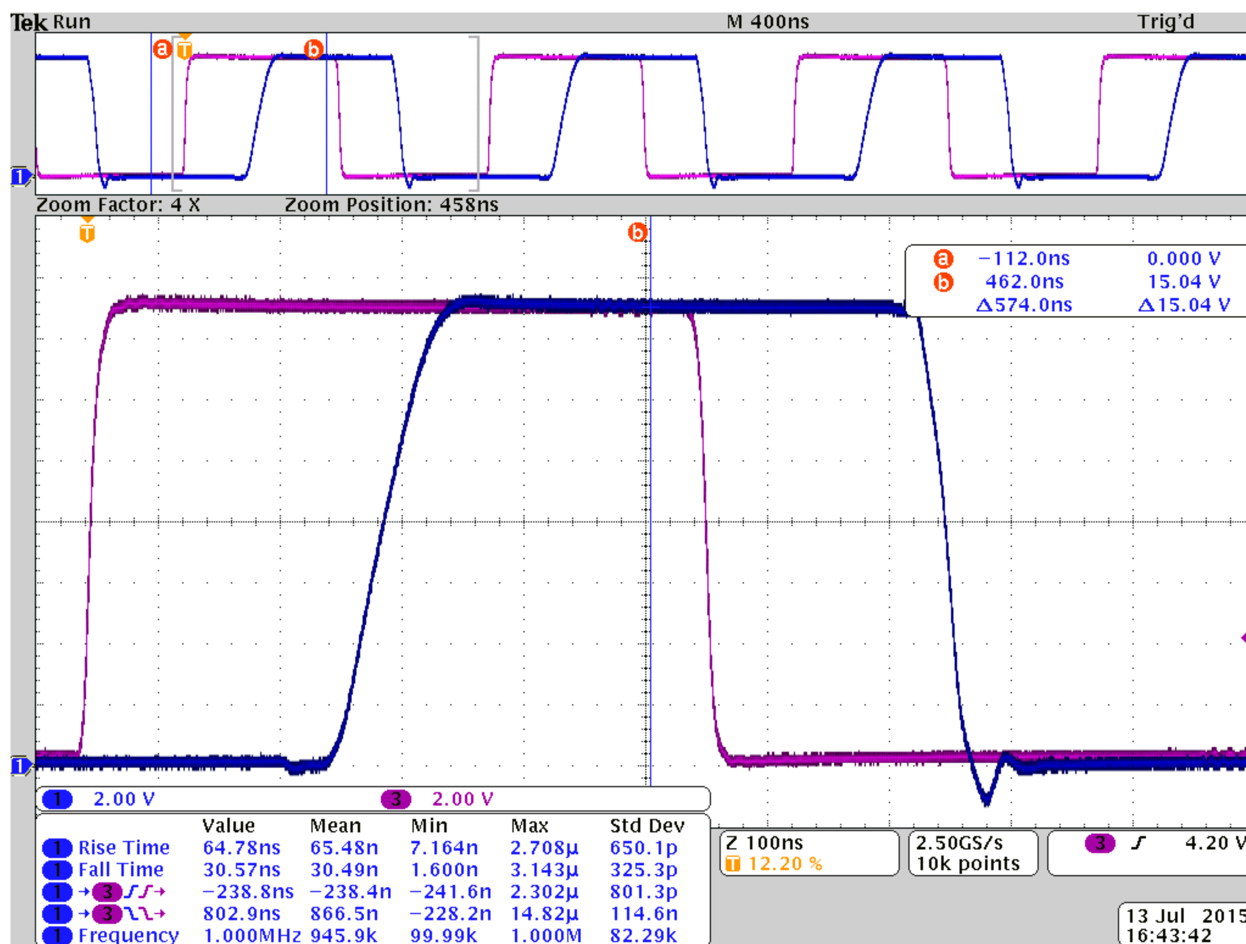


Fig. 7.10. Representative driver output at 454 °C with a 1 MHz square wave driving C2M0280120D MOSFET by W31 R3C2.

After the R3C2 sample was tested, no further QFP64 packages were available, and the test setup was reconfigured for the new supply of LDCC68 packages. Three devices were packaged, using subsites R4C2, R3C4, and R6C1. After packaging, R3C4 demonstrated significant power supply quiescent current that increased above a power supply voltage of 13 V. As such, testing for this die was limited to 12 V over temperature. While functional, the results were interesting from the perspective of examining a partially degraded die with an unidentified failure mode. The second die tested over temperature, R6C1, did not have quiescent current issues, and was tested at 12 V as well as 15 V.

The DC drive strength was also measured to evaluate the strength of the driver under static conditions. These static conditions exist during normal operation when the driver must resist external influences to the gate voltage. The prime example of such an event is cross-talk, when the drain voltage changes during switching, resulting in a gate current coupled through C_{GD}. The output of the gate driver was set to either V_{DD} or V_{SS}, and a Keithley 2602 Source Measure Unit (SMU) applied a current bias of 100 mA to the OUTPUT pin. When the output was low, the current was sourced into the OUTPUT pin, and when the output was high, the current was sunk from the OUPUT pin. A relatively large current of 100 mA was selected in order to reduce potential measurement errors due to leakage, and to approach the magnitude of currents expected during cross-talk events. Two sites, R6C1 and R3C4, were tested over temperature with a V_{DD} of 12 V, and R6C1 was also tested at a V_{DD} of 15 V. In Fig. 7.11, the two die demonstrate different results over temperature, with the R3C4 die developing worse NFET performance, and the R6C1 die developing worse PFET performance. Comparing die R6C1's 12 V performance in Fig. 7.11 to the 15 V performance in Fig. 7.12 shows little difference between the measured output resistance at 12 – 15 V. Site R6C1 exhibits a small degradation in NFET behavior over temperature, starting at 1.17 Ω at room temperature increasing to 1.59 Ω at 528 ℃.

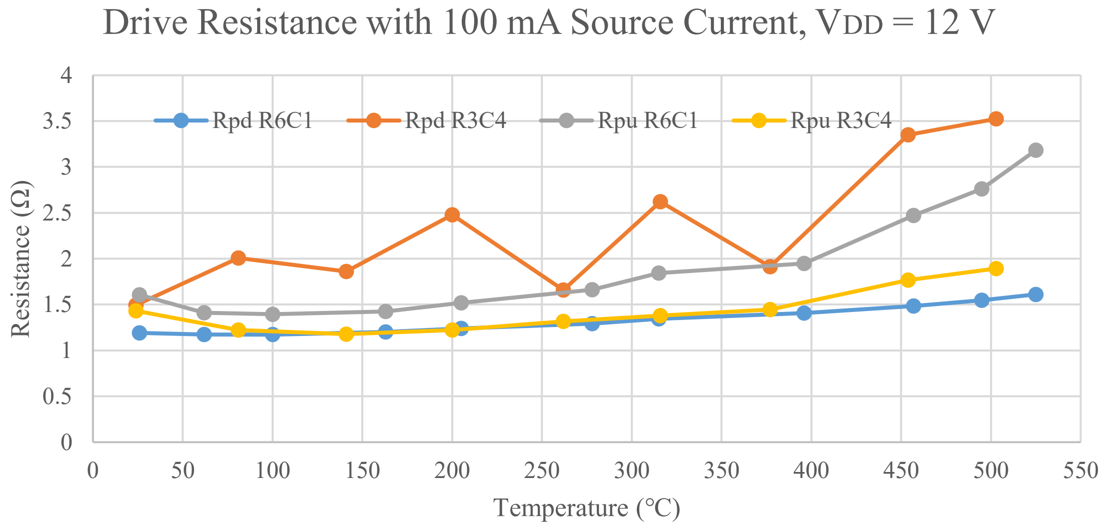


Fig. 7.11. Gate driver output resistance over temperature, with a V_{DD} of 12 V.

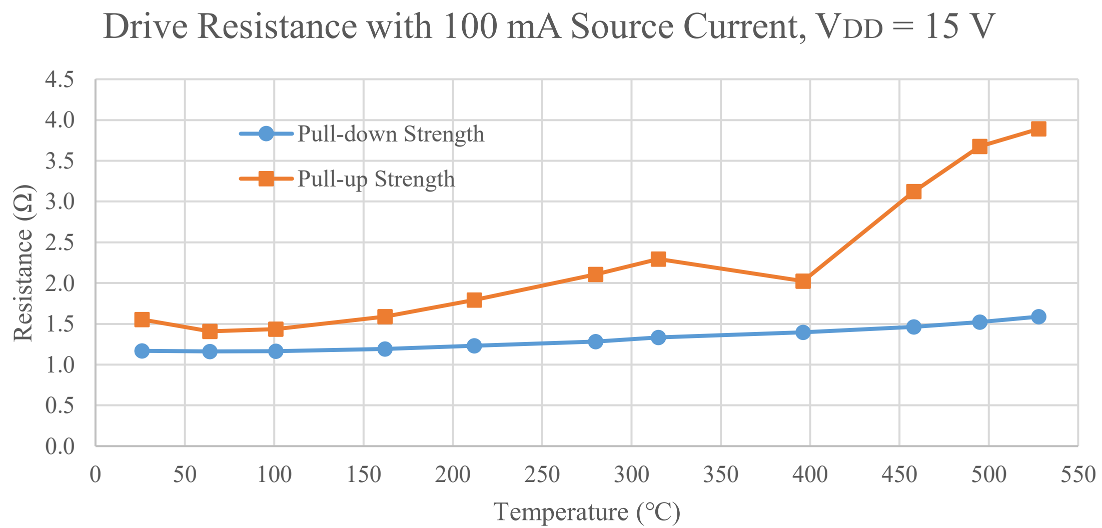


Fig. 7.12. Gate driver output resistance of W31 R6C1 over temperature with a V_{DD} of 15 V.

The transient behavior was recorded over temperature as well, consisting of rise and fall times of the gate driver output, as well as propagation delays from the input pin to the output pin. Again, the R3C4 die was tested at only 12 V, while the R6C1 die was tested at both 12 V and 15 V. The 12 V results, shown in Fig. 7.13, demonstrate a steady increase in rise and fall times over temperature for both die. The fall time for the R6C1 die is dramatically better than the R3C4 device, and the 15 V results of the R6C1 die in Fig. 7.14 are slightly worse than the 12 V times

in Fig. 7.13. These results, coupled with the drive strength comparison, suggest a minimal difference in the transient behavior between the power supply voltages. In fact, the only parameter to demonstrate significant variation over supply voltage was propagation delay from the input to the output. The propagation delay at 12 V, shown in Fig. 7.15, as well as the delay at 15 V in Fig. 7.16, both show an initial decrease in time as the temperature increases to 150 °C, and a steady increase in delay after 200 °C. However, the difference between the 12 V R6C1 data in Fig. 7.15 to the 15 V R6C1 data in Fig. 7.16 is a consistent 100 ns decrease with the additional supply voltage. The logic propagation delay is sensitive to supply voltage variation, causing the large difference between the two figures. A representative switching waveform captured is shown in Fig. 7.17 while the die was heated to 530 °C.

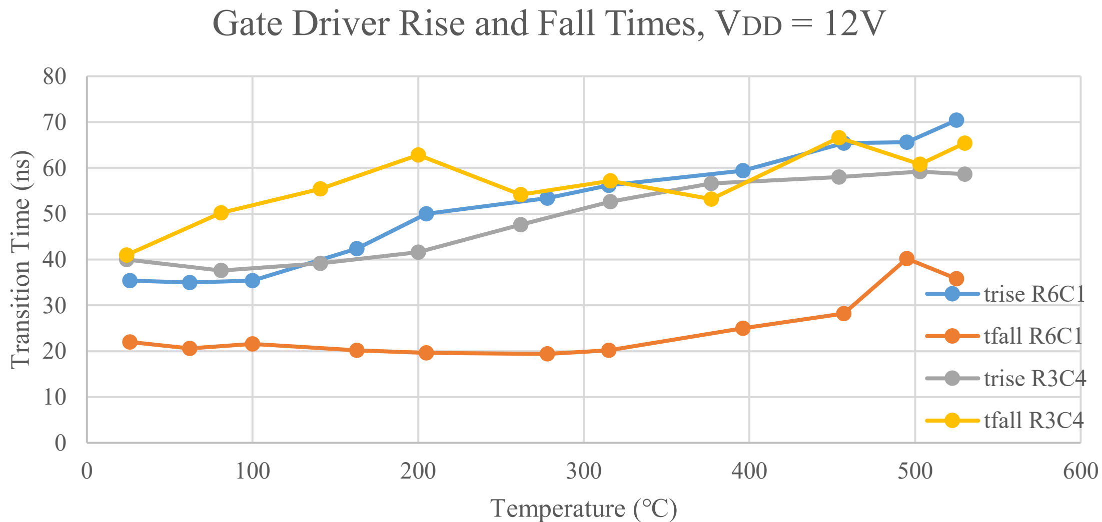


Fig. 7.13. Gate driver rise and fall times over temperature at a V_{DD} of 12 V driving the gate of a C3M0065090D power MOSFET.

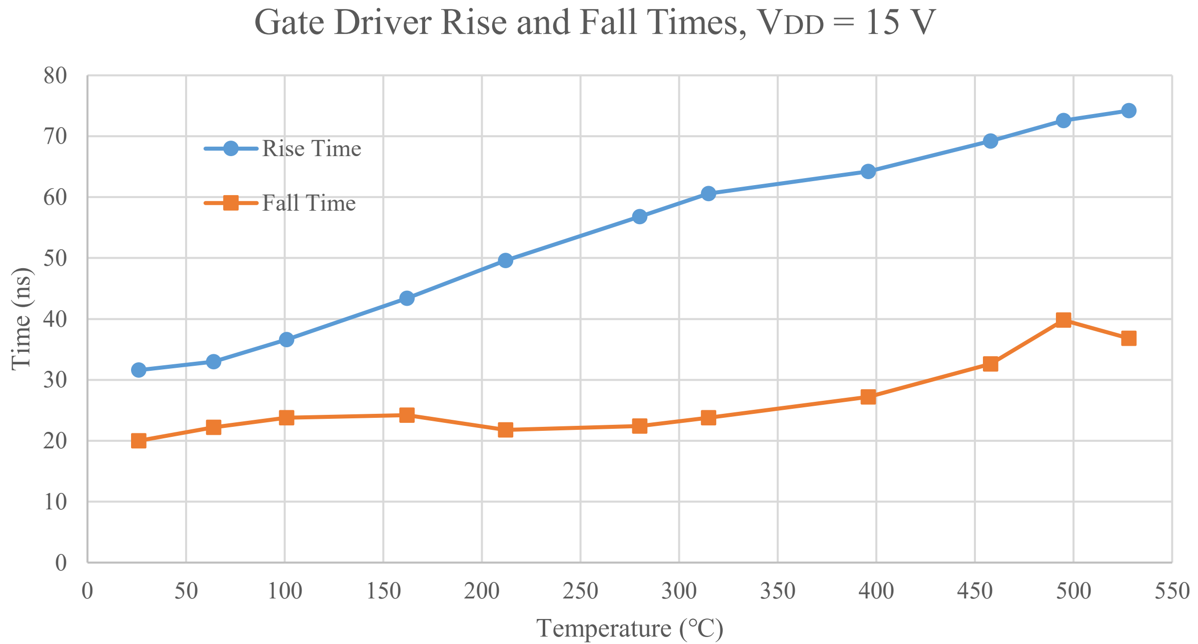


Fig. 7.14 Gate driver rise and fall times over temperature for die W31 R6C1 with V_{DD} = 15 V driving the gate of a C3M0065090D power MOSFET.

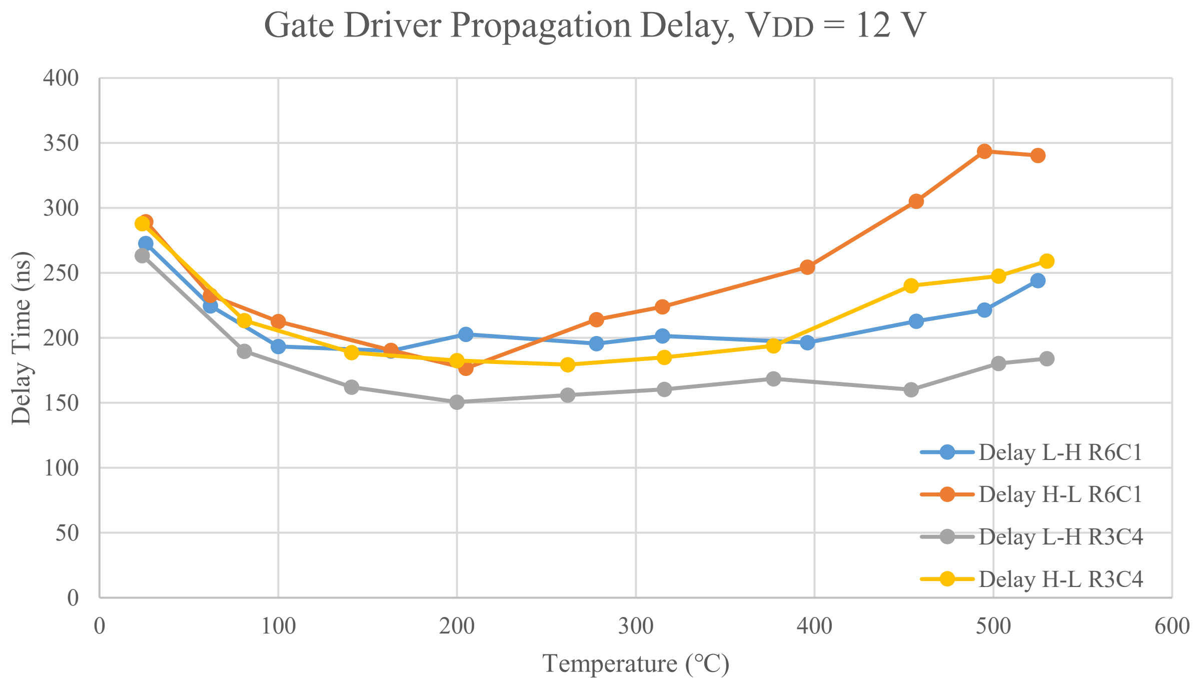


Fig. 7.15. Gate driver propagation delay with V_{DD} = 12 V driving the gate of a C3M0065090D power MOSFET.

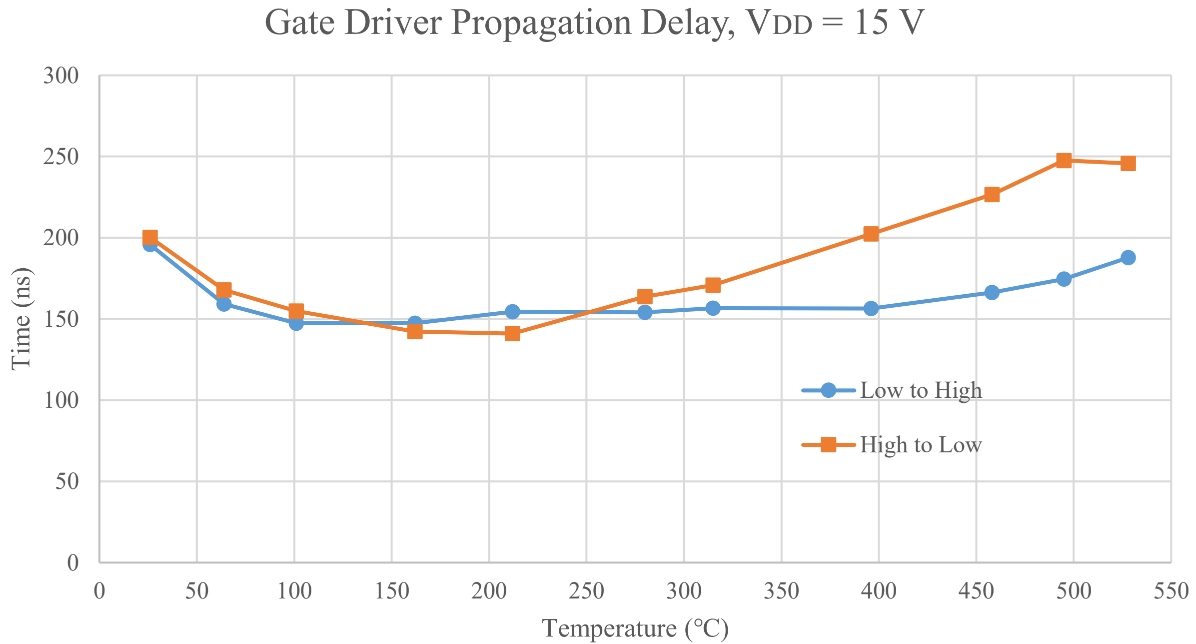


Fig. 7.16. Gate driver propagation delay of W31 R6C1, with V_{DD} = 15 V driving the gate of a C3M0065090D power MOSFET.

Both test setups presented similar data on the performance of the gate driver over temperature, though there were some differences. The most obvious difference between the two setups is the difference in the power MOSFET used as a load for the gate driver. In the first test, a C2M0280120D device was used with an effective gate capacitance of 259 pF. At room temperature, the difference between the loaded and unloaded driver was minimal due to the small capacitance. With the latter two tests using the larger C3M0065090D device, the 660 pF gate capacitance presented a loading effect on the gate driver, slowing transition times. The degradation in drive strength of the pull-up devices recorded in Fig. 7.12 manifest itself in the degradation in rise times in Fig. 7.14. Compared to the rise and fall times of Fig. 7.6, there is a degree of site to site variation between the two drivers that is also present and is difficult to quantify entirely. However, the basic trends of each waveform indicate useful examples of behavior over temperature.

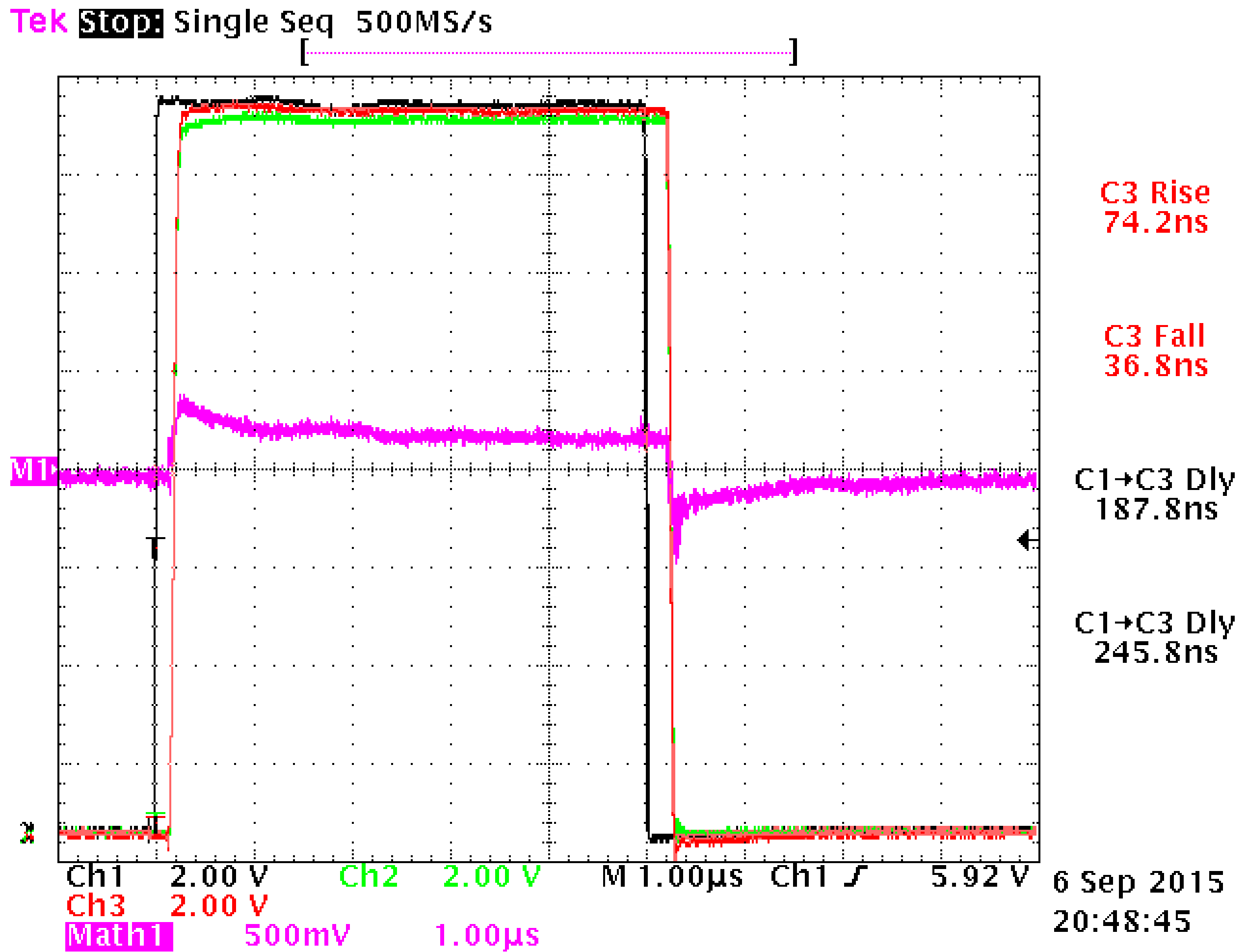


Fig. 7.17. Representative waveform R6C1, 528 °C, driving a C3M0065090D.

Power module integration

One of the goals of the gate driver is to simplify the system design to the point where the driver can be integrated into the power module. To that end, a double-pulse test module was designed to demonstrate a gate driver directly driving a CPM3-0900-0065B MOSFET. A 15 A, 1200 V Cree SiC diode, CPW4-1200-S015B, was used as the high-side switch to simplify the design setup. The gate driver was directly connected to the CPM3-0900-0065B MOSFET, and a CD4504 level shifter and a 78L05 linear regulator were added to allow direct interfacing to the FPGA board previously used for the high temperature tests. Capacitors were added for the low

voltage supplies, and a high voltage ceramic capacitor was added for the high voltage supply in order to reduce the loop inductance. The complete schematic is shown below in Fig. 7.18. In the interest of rapid prototyping, the board was developed for a standard PCB design with FR-4 material. An Electroless-Nickel, Immersion Gold (ENIG) finish was chosen for the PCB, as previous experiments verified the capability for this finish to accept 1 mil gold ball bonds used with the gate driver, as well as 5 mil aluminum wedge bonds used for power devices. The board layout was optimized around minimizing the total switching loop inductance, or the inductance in a loop through Q2, D1, and C3. An additional pad for the MOSFET source was added as a Kelvin connection to minimize the common source inductance between the power loop and the gate drive loop. The final board layout is shown in Fig. 7.19 as wirebonded.

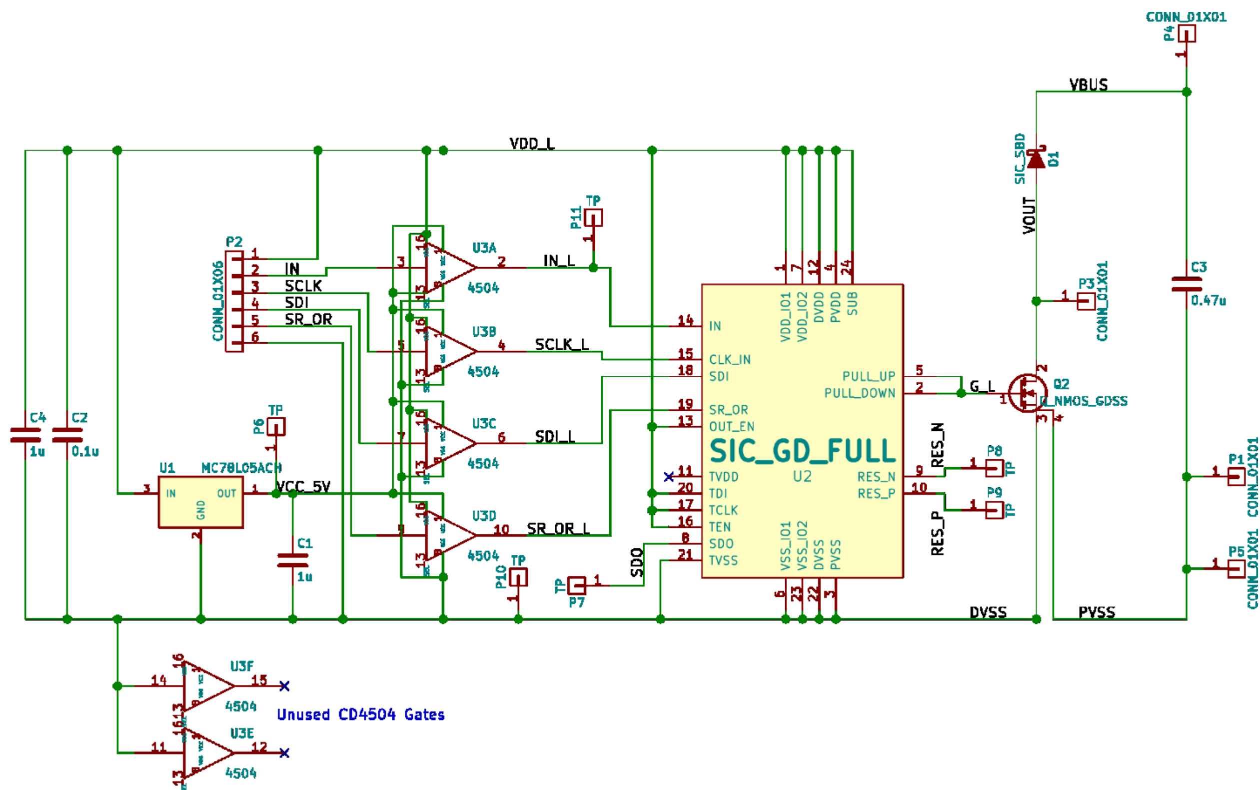


Fig. 7.18. Power module demonstration schematic.

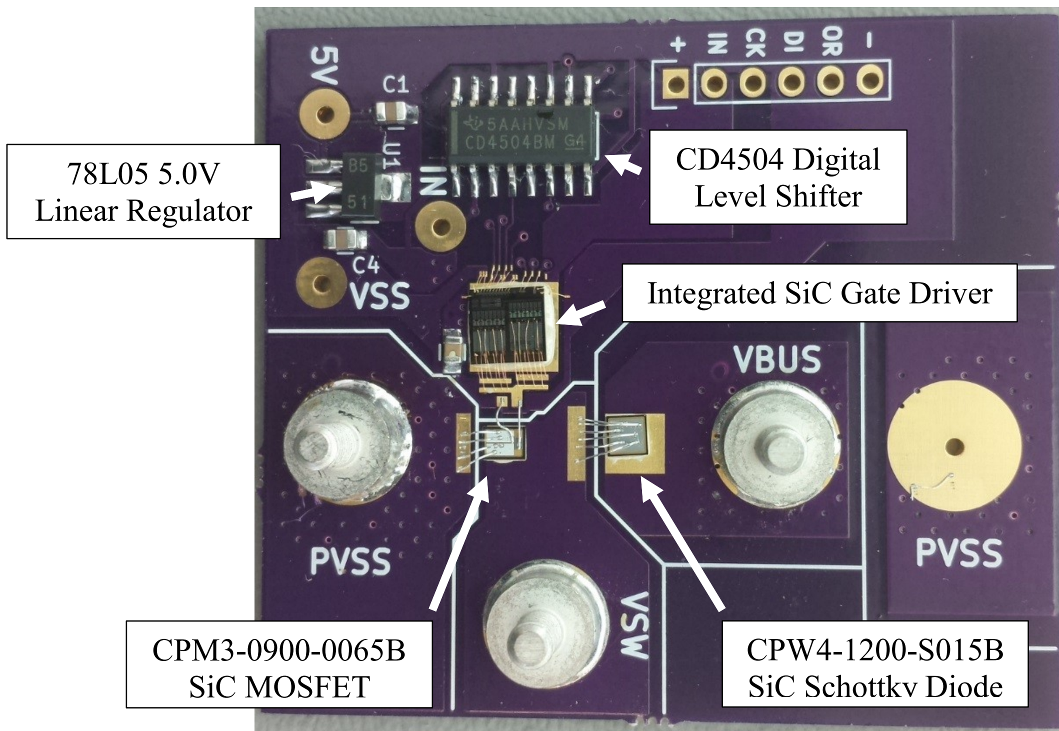


Fig. 7.19. Assembled board with all components soldered and wirebonded.

High voltage testing was performed by connecting a Sorensen SGA 800/19 and a 150 µF, 450 V aluminum electrolytic capacitor across VDD. A 200 µH air-core load inductor was used for the clamped inductive load. Due to the prototype nature of the boards, the diode and MOSFET have limited continuous power capability. As a result, a fixed number of short (5 µs to 50 µs) pulses were used to achieve the desired test current. A function generator was used to generate the pulses, and the period of the pulses was adjusted to reach the target current. The size of the inductor was sufficiently large that the current did not degrade significantly over the time between the pulses. An image of the test setup is presented in Fig. 7.20.

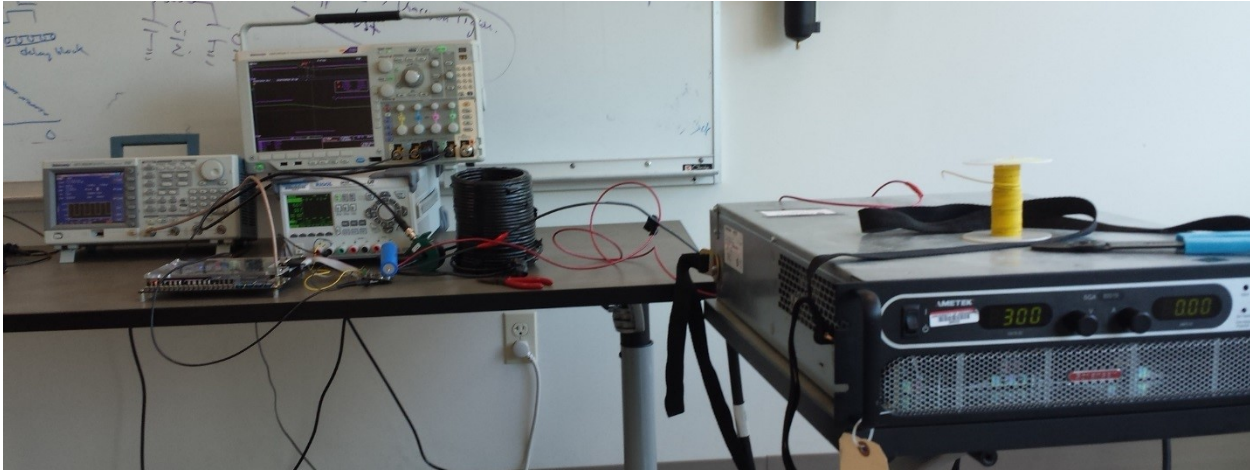


Fig. 7.20. High voltage test setup with power supplies, function generator, FPGA board, oscilloscope, and module.

Given no restrictions on signals, a few signals are interesting for a gate driver test result. Arguably, the most important signal of a gate driver is the output, or the gate of the power MOSFET. Both the output voltage and current are useful for evaluating the condition of the MOSFET as well as validating performance requirements. Also important is the drain-source voltage and current. When properly combined, the drain-source voltage and current can be used to calculate switching loss. The logic input to the gate driver is necessary to determine propagation delay, and the bus voltage is useful for evaluating whether power supply voltages change significantly after switching edges. The other signal useful is the inductor current, as it will show the load condition at the moment of switching.

Monitoring the waveforms of this test setup is a troublesome affair. Direct measurement of the drain-source current is impossible without the addition of significant source inductance. Existing techniques for current measurements either lack the necessary bandwidth, such as Rogowski coils or Hall effect sensors. Others have significant insertion inductance that distorts the system response to switching at high speeds, such as broadband current transformers, surface mount current shunts, and coaxial current shunts. Combined with the difficulty of synchronizing

the current waveforms with the voltage waveforms, the drain-source current was deliberately omitted in order to improve total system performance. Power supply current provides limited information about the current flowing through the transistors due to the capacitors placed across the power supply. In fact, the only current that can safely be monitored without impacting the system performance is the inductor current. A current transformer, a Pearson current monitor model 411, was placed on the VBUS lead of the load inductor in order to know the current at MOSFET turn-on and turn-off. The 20 MHz bandwidth of this sensor is not critical, as the actual inductor current will change relatively slowly, and the lower cut-off frequency of 1 Hz eliminates the observed droop on the time scale used for switching.

Another complication in measurement is the problem of signal coupling through the ground lead of the oscilloscope used for voltage measurement. Under the most benign conditions, multiple ground connections provide a signal coupling path between power signals. At the worst case, a current path will form between the ground leads of separate probes, leading to measurement error in both channels and potentially resulting in corruption of the gate drive signal. If the gate drive signal fails to turn the power MOSFET on and off quickly and without negative feedback, the power module may oscillate and lead to module destruction. Multiple ground connections for the oscilloscope are avoided for these reasons. However, without a direct ground connection, an oscilloscope probe will not provide consistent results when presented with noise and fast varying signals.

The probing strategy used for the high voltage testing is to measure the inductor current and the drain-source voltage of the power MOSFET. Only one ground lead is used at a time, causing only one voltage signal to be usable. The current transformer monitoring the inductor current is isolated, and can be added without extra consideration. Some waveforms include a

gate-source signal, but the gate-source signal is corrupted as soon as the drain voltage switches at higher voltages. Due to the high bandwidth requirement of the power module, standard 10x attenuation, 500 MHz bandwidth passive probes were used for drain-source voltage measurements. These probes have a limit of 300V on the input, which restricts the maximum bus voltage used. The standard ground and tip clips were removed, and a wire holder was fashioned to hold the bare probe in contact with a wire target for minimum ground loop distance, shown in Fig. 7.21. Optimization of the ground lead inductance is critical in maintaining a high resonant frequency for the oscilloscope probe. A long ground clip would decrease the resonant frequency of the oscilloscope probe into the range of frequencies that are present in the system, skewing the measured results. Gate voltages are also difficult to accurately measure, as the only target for probing is a portion of the gate pad on the power module PCB. Extreme care must be taken to precisely place the probe tip without touching the wire-bonds to the power MOSFET or gate driver, and a direct ground connection is impractical.

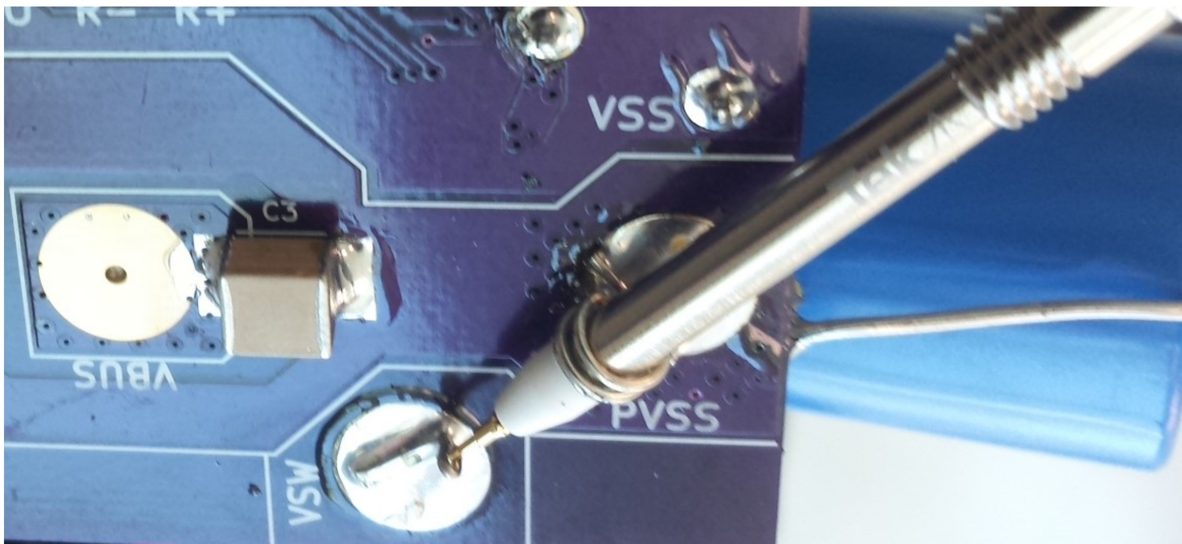


Fig. 7.21. Oscilloscope probe connection for the drain-source voltage measurement. A wire loop holds the ground portion of the oscilloscope probe and makes the connection to PVSS, and a wire target is soldered to VSW for the tip contact. The probe is taped using electrical tape (not shown) for additional mechanical stability.

With the unknown performance of the gate driver and system, the voltage was increased in small increments in order to allow for data collection at each point in case of catastrophic failure. The DC bus voltage was increased from 60 V up to 300 V while a series of pulses were applied to increase the current through the inductor up to 10 A. A representative waveform of the test configuration for 10 A is shown in Fig. 7.22. Higher currents demonstrated in other 300 V waveforms are achieved by using four pulses instead of two. The bus voltage of 300 V was selected as an upper limit for two primary reasons. The first was the 300 V oscilloscope voltage probe limit, which restricted the maximum usage voltage without causing damage. The other limit was the module itself, as the power devices are not encapsulated. As bonded, the module was close to requiring encapsulation in order to ensure that creepage requirements as well as air breakdown voltages are not exceeded. A silicone encapsulant could be used to increase the breakdown voltage significantly to the point where the devices are the primary breakdown limitation.

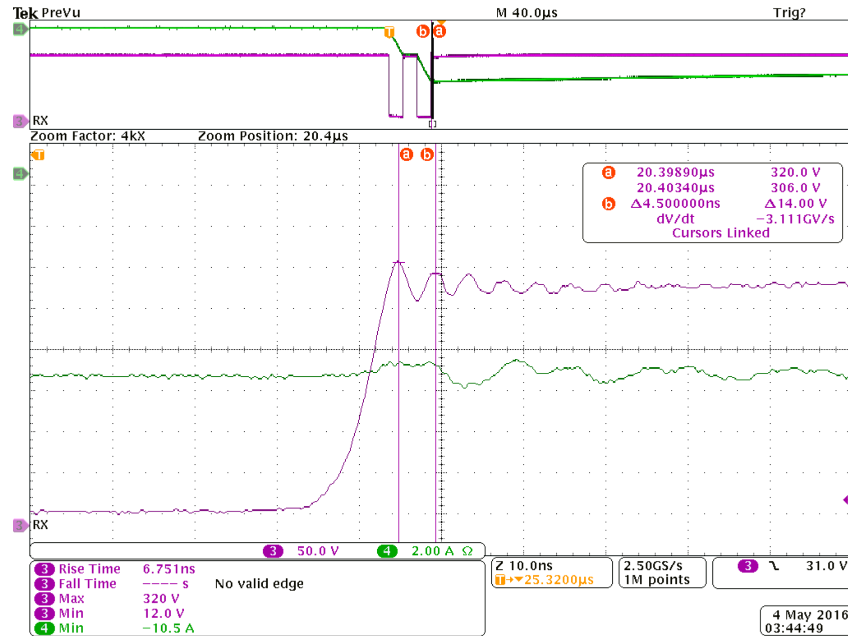


Fig. 7.22. Representative oscilloscope waveform demonstrating the 2 pulse test configuration and turn-off waveform at a V_{BUS} of 300 V and an inductor current of 10 A.

Stepping the bus supply voltage from 60 V to 300 V demonstrates several interesting trends. The data across voltage, is shown in Fig. 7.23. A few interesting points immediately are demonstrated. The first is that the overshoot waveform remains relatively consistent in magnitude across all load voltages, changing in magnitude only when the load current increases. As a result, the worst overshoot by percentage occurs at the lowest voltages, and the lowest overshoot by percentage of V_{BUS} occurs at 300 V. As expected, the magnitude of the current determines the magnitude of the step response after the MOSFET fully turns off. Another characteristic of the system is that the ringing after turn-off rapidly decays within 30 ns at 5 A, and 45 ns at 10 A.

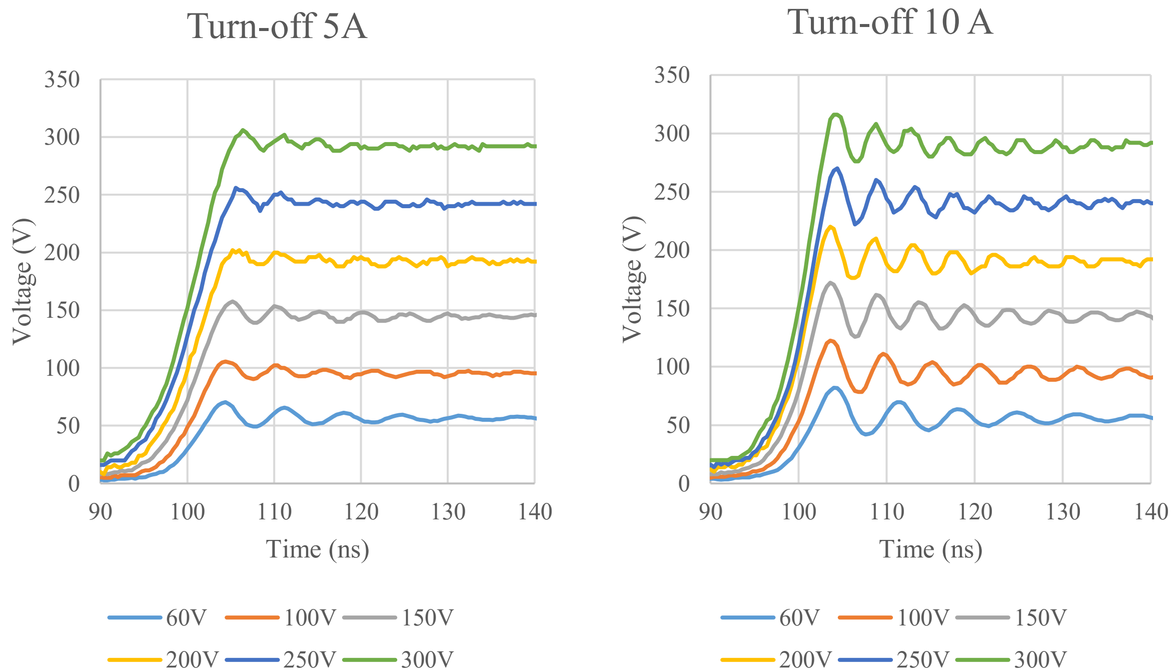


Fig. 7.23. Turn-off waveforms at a constant inductor current across multiple bus voltages. Waveforms have been synchronized to the peak voltage after turn-off. The different load conditions are based on the inductor load current at the time of turn-off.

Another interesting effect demonstrated by the turn-off waveform corresponds to the frequency of the overshoot ringing. As can be seen in Fig. 7.23, the period of the 60 V signal differs significantly from the period of the 300 V signal. This difference in period and respective

frequency can be explained by the nonlinear capacitance behavior of the power MOSFET. The C_{OSS} curve of the C3M0065090D MOSFET [33], which includes C_{DS} and C_{GD}, varies significantly over voltage as shown in Fig. 7.24. The ringing signal is caused by excitation of the series RLC circuit formed by the C_{OSS} of the MOSFET and the wiring inductance going through the high-side switch, low-side switch, and the power supply capacitor. Assuming that the capacitance is dominated by the MOSFET C_{OSS}, and that the wiring inductance remains constant, the inductance of the module can be estimated by using equation (7.2) and the C_{OSS} versus V_{DS} curves provided in the datasheet. The peak-to-peak period of the waveform is measured using the oscilloscope cursor functions, which provide limited accuracy at the frequencies of interest. Comparing the measured frequencies to the expected frequencies given fixed inductances as shown in Fig. 7.24, the calculated inductance of the module based on the drain-source capacitance varies between 4 and 6 nH.

$$f = \frac{1}{2\pi\sqrt{LC}} \tag{7.1}$$

$$L = \frac{1}{(2\pi f)^2 C_{OSS}(V)} \tag{7.2}$$

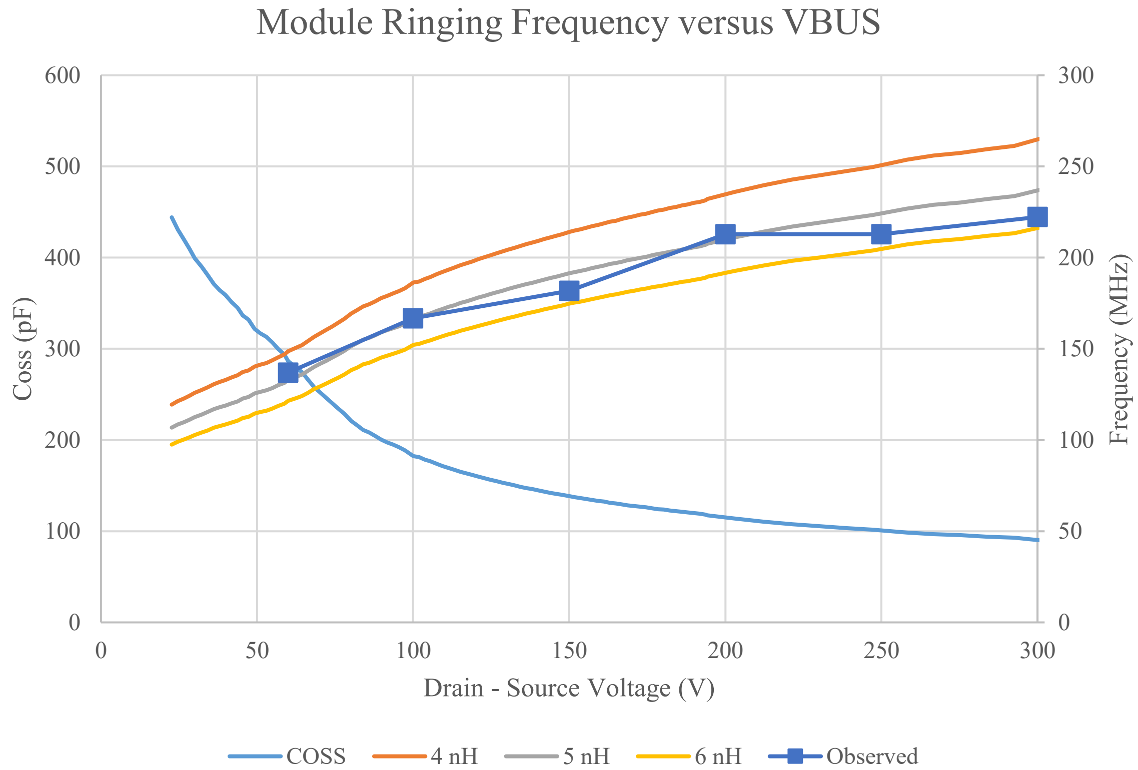


Fig. 7.24. A comparison of C_{OSS} of the C3M0090065 device across applied drain voltage. By assuming a fixed inductance in a resonant circuit, a projected series resonant frequency can be projected as a function of V_{DS}. The observed resonant frequency tracks between 4-6 nH loop inductance.

The turn-on waveforms remain tame by comparison, as shown in Fig. 7.25. A small amount of ringing is present at the 60 V test, presumably a feedthrough of the high-side ringing. Turn-on time increases slightly with increased V_{BUS}, but not by a large margin.

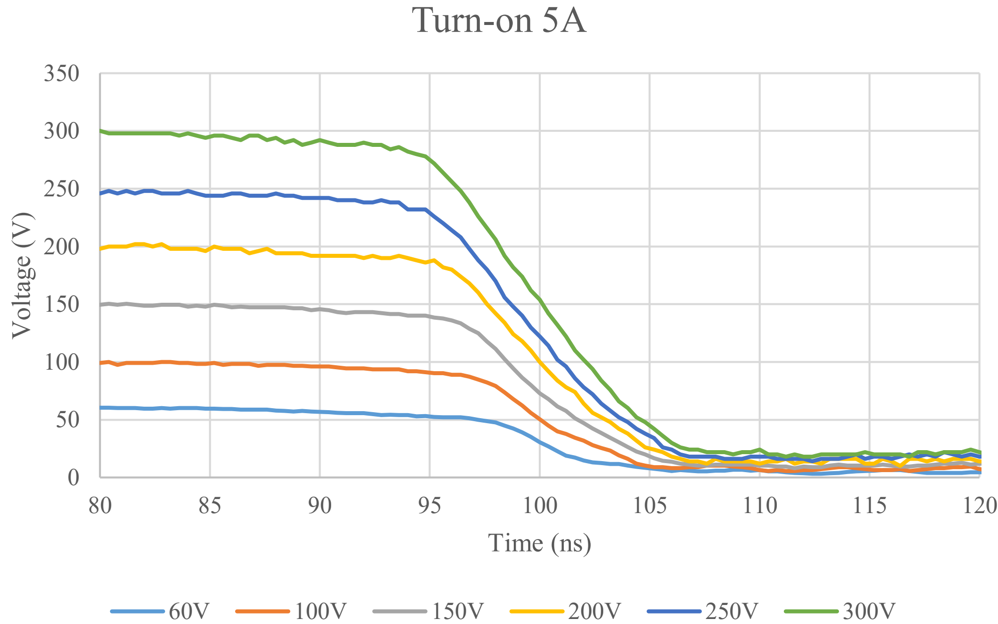


Fig. 7.25. Turn-on waveforms at a constant inductor current across multiple bus voltages. Waveforms have been synchronized to the 50 % transition point.

While a variation in bus voltage exposes voltage dependencies in effects such as turn-on time and ringing frequencies, changing the load current demonstrates other effects. The turn-on waveform at 300 V bus voltage shows little variation over a load current variation from 5 A to 15 A, as seen in Fig. 7.26. The major variation comes in at turn-off, where the load current is responsible for driving the output voltage high. The turn-off current tested in Fig. 7.27 ranges from 5 A up to 20 A, and demonstrates a significant difference between the 5 A case and the higher current cases. With the load current responsible for charging the output capacitance C_{OSS}, the lower 5 A current takes longer to increase the voltage above VBUS. The rise time of the voltage waveform decreases with increasing load current, and the corresponding *dv/dt* increases. The peak overshoot voltage, which is determined by the step response from the current waveform, increases as the load current increases. Even at the higher currents, the peak overshoot reached only 342 V, or about 17 % overshoot from the bus voltage immediately after which the switching transients dissipate.

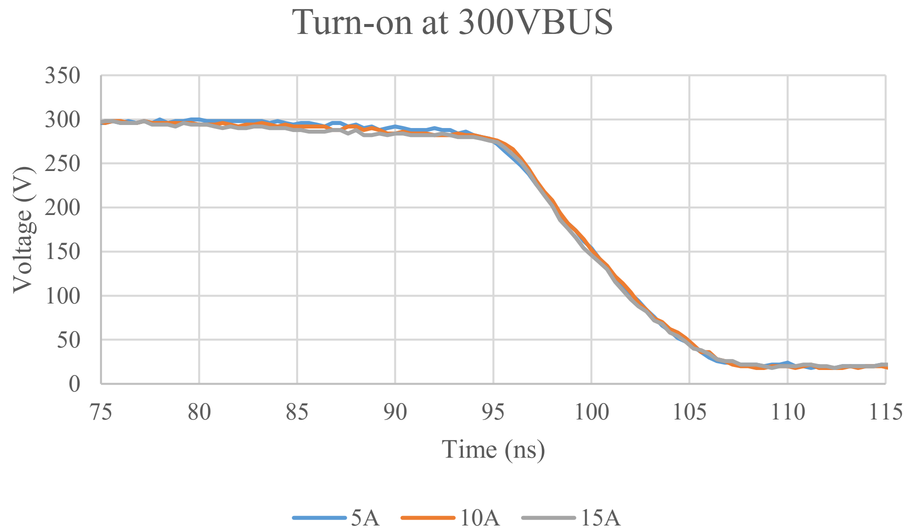


Fig. 7.26. Turn-on waveforms with varying inductor current and a 300 V DC bus voltage.

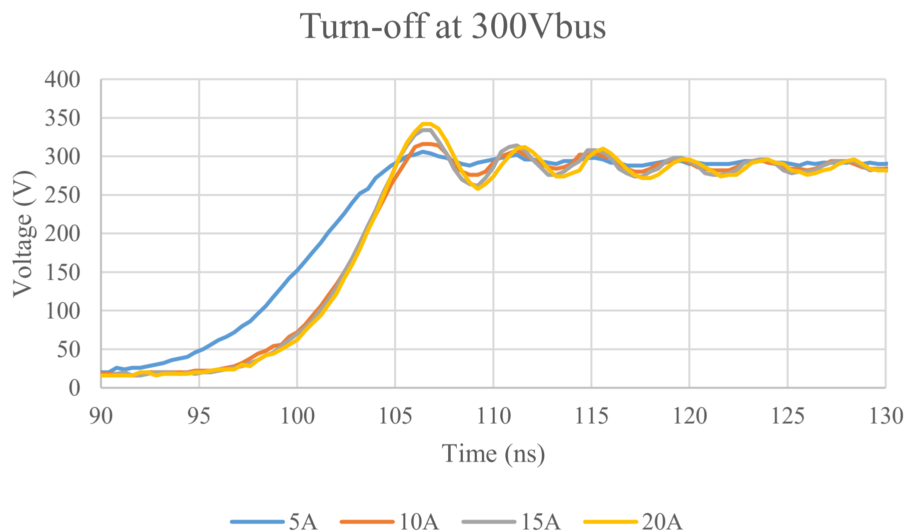


Fig. 7.27. Turn-off waveforms with varying inductor current and a 300 V DC bus voltage. Waveforms are roughly synchronized to the peak voltage after turn-off.

Adjustable drive strength demonstration (driving a loaded FET)

Another one of the claims of the setup was that the drive strength of the gate driver can be adjusted to affect switching parameters. While the drive strength directly impacts the gate-source voltage waveform, this also affects the drain switching waveforms. A bus voltage of 60 V was used to demonstrate the adjustable drive strength feature. This voltage provides some of the worst switching behavior due to the relatively high C_{OSS} and C_J of the power devices at that

voltage. By percentage, this voltage demonstrated the worst overshoot of all voltages. An additional factor was that a lower drain voltage is safer in the event of device failure. A two pulse setup was used as before with the higher voltage configurations, and the load current was targeted as 5 A for turn-on and 10 A for turn-off. The drive strength was varied for both the turn-on and turn-off cases at the same time using the FPGA board switches to specify which drive transistors were active at the same time. Table 7.3 shows the progression of switches used to test the different drive strength, with the programming byte implemented on both the pull-up and pull-down configuration register. The waveform for each drive strength is shown in Fig. 7.28, with a clear progression from maximum drive strength to minimum drive strength demonstrated qualitatively.

Table 7.3 Drive strength progression

DRIVE STRENGTH	PROGRAMMING BYTE
8	11111111
7	01111111
6	01011111
5	01010111
4	01010101
3	00010101
2	00000101
1	00000001

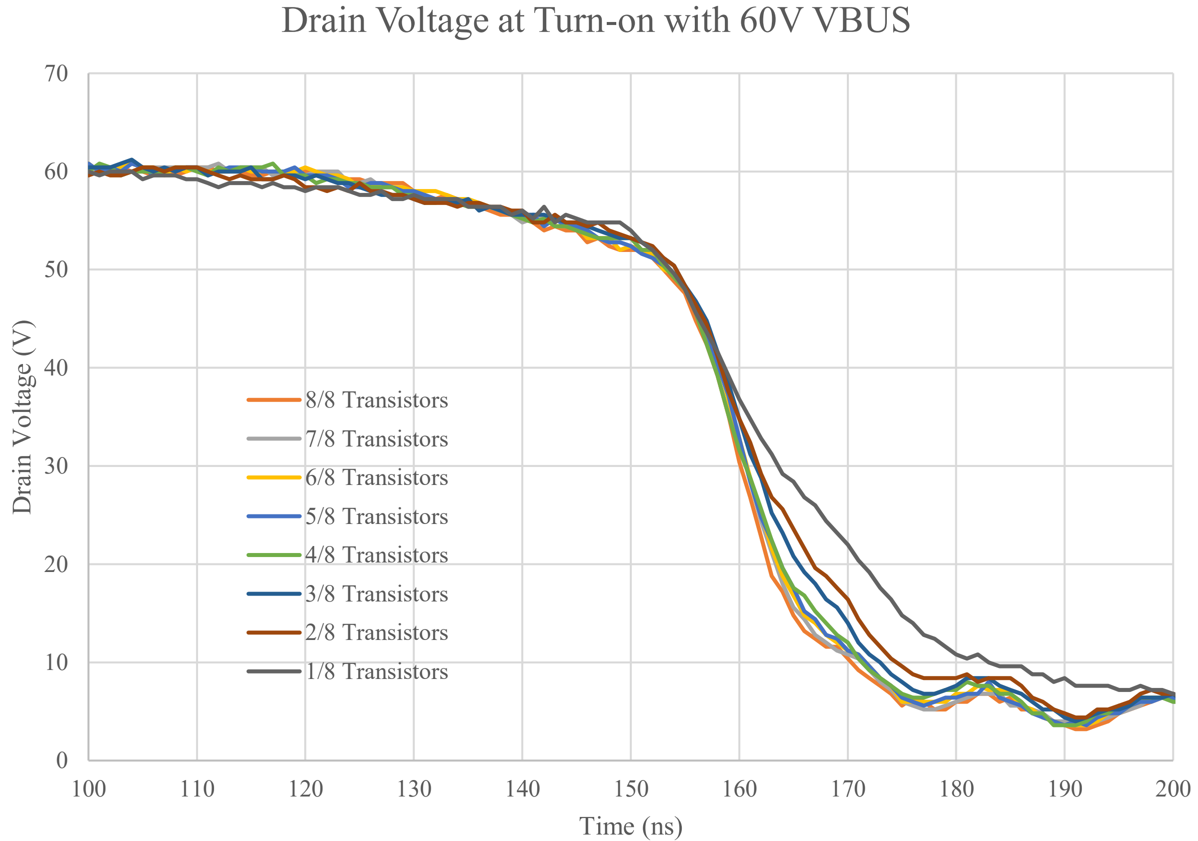


Fig. 7.28. Turn-on waveforms over various drive strengths. The load inductor current is 5 A with a nominal drain voltage of 60 V.

Looking at the measured analysis, two key performance indicators are available for turn-on based on the output voltage measured. The first is the rate of change of the voltage, or *dv/dt*, and the second is the rise and fall times. While closely related, the impact of a higher peak *dv/dt* is seen in noise immunity issues, while rise and fall times show how quickly the power MOSFET turns on. For the turn-on case, the current at turn-on was set to 5 A. The peak positive and negative *dv/dt* observed during MOSFET turn-on is shown in Fig. 7.29. For turn-on, the peak negative *dv/dt* occurs when the output falls from V_{BUS} to 0 V. The positive *dv/dt* corresponds to the small amount of ringing seen in the output immediately after turn-on in all cases except the 1/8 drive strength. The *dv/dt* was sampled with a moving window to identify the maximum change in voltage *dv/dt* over 1.6 ns. This windowing eliminates the peak *dv/dt* seen from noise in

the output voltage at steady state. The magnitude of the negative *dv/dt* decreases with reduced drive strength, resulting in a slower transition rate with decreased drive strength. The drive strength range of the gate driver provided roughly a 2:1 range of control for peak negative *dv/dt*. Fall time, shown in Fig. 7.29, remains roughly constant until the drive strength falls below 3/8 transistors active.

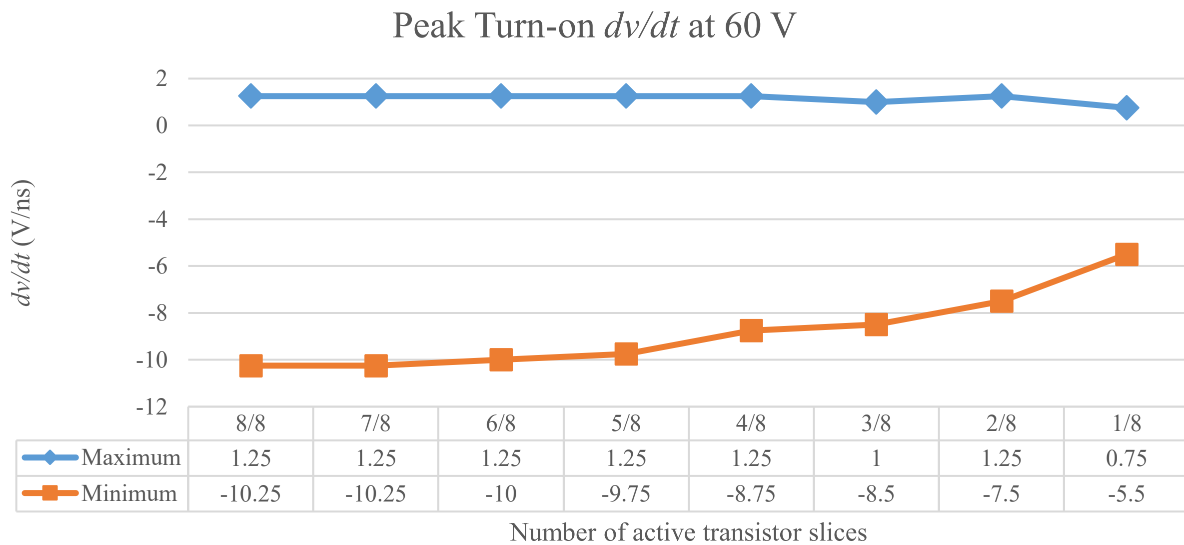
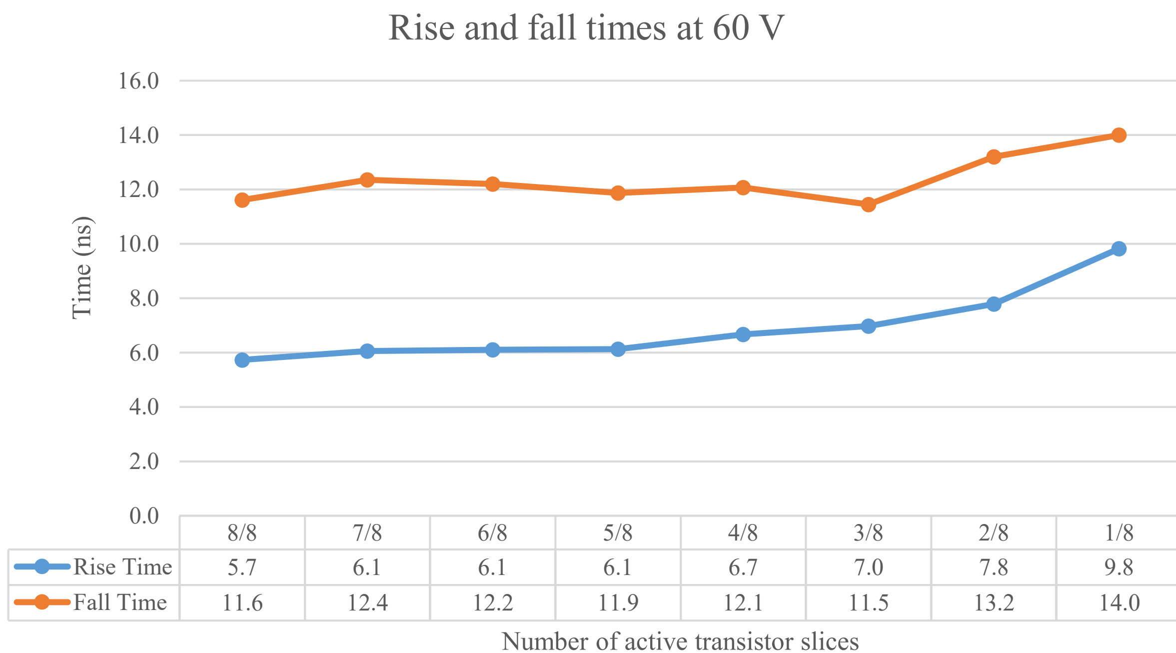


Fig. 7.29. Peak turn-on *dv/dt* while varying the drive strength. Switching conditions were a VBUS of 60 V and an inductor current of 5 A. The maximum *dv/dt* corresponds to any ringing after turn-on, and is near the observed noise level in the system. The minimum *dv/dt* is the main slew rate during switching.

At the turn-off switching edge, the rise time and *dv/dt* also play a part in determining the performance of the system. Another parameter that exists at turn-off is overshoot of the bus voltage. This overshoot is dependent on how quickly the low-side device turns off, and can result in high drain voltages. Excessive overshoot and corresponding drain-source voltages may approach the maximum V_{DS} limit of the power MOSFET, requiring further de-rating of the system or causing system damage. Decreasing the drive strength at turn-off should reduce peak *dv/dt* and decrease rise time and peak overshoot. The various rise waveforms are shown in Fig. 7.31. Starting with the rise time, Fig. 7.30 demonstrates the change in rise time as the drive

strength is varied from 8/8 transistors active down to 1/8 transistors active. The rise time begins significant change after reducing the drive strength to 4/8 slices, with the rise time increasing steadily as the drive strength is reduced further. The rise time increases by 4.1 ns from the maximum drive strength case, or a 72% increase in turn-off time. Closely related to turn-off rise times is the peak *dv/dt* observed during turn-off. While the turn-on waveform is relatively simple, turn-off includes resonant behavior after the device has completely turned off. This results in a high positive *dv/dt* as the voltage ramps from the low voltage up to the bus voltage, and a large negative *dv/dt* following the peak as the overshoot falls and rings at the natural resonant frequency. The *dv/dt* over drive strength is shown in Fig. 7.32, and the *dv/dt* magnitude demonstrates a trend with the drive strength. The rising *dv/dt* decreases from 15.5 V/ns at maximum drive strength to 8.25 V/ns, a 47 % decrease. The falling *dv/dt* corresponds to the magnitude of the ringing after switching, and shows a 66 % decrease in magnitude.



	8/8	7/8	6/8	5/8	4/8	3/8	2/8	1/8
Rise Time	5.7	6.1	6.1	6.1	6.7	7.0	7.8	9.8
Fall Time	11.6	12.4	12.2	11.9	12.1	11.5	13.2	14.0

Fig. 7.30. Rise and fall times as a function of drive strength. All situations are performed with a bus voltage of 60 V. Turn-on corresponds to a 5 A load current and to a turn-off 10 A load current.

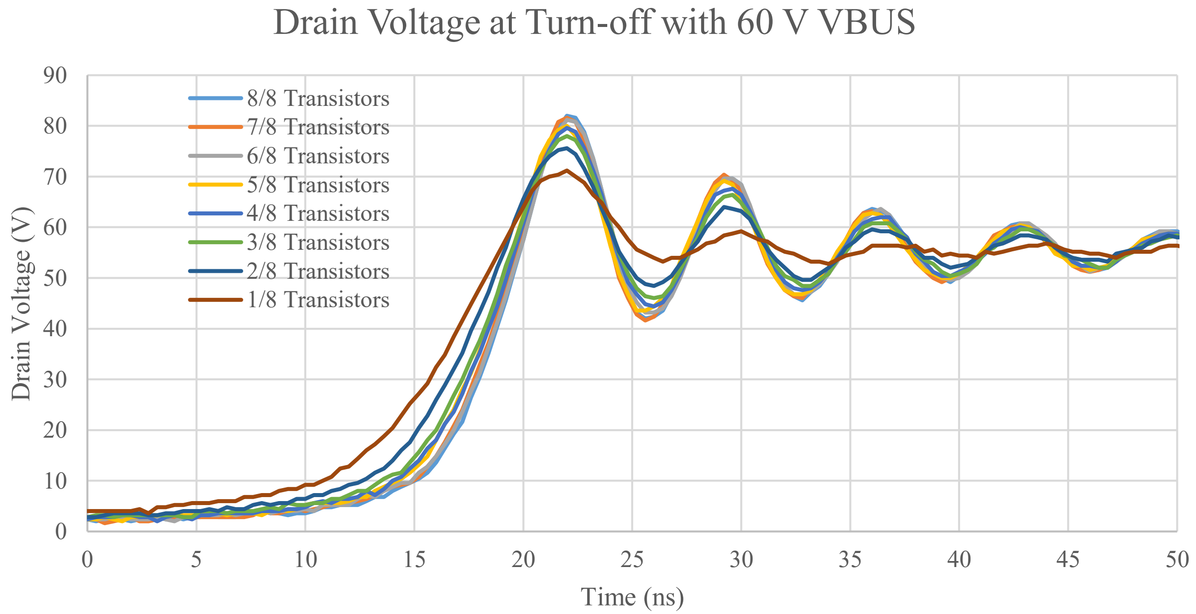
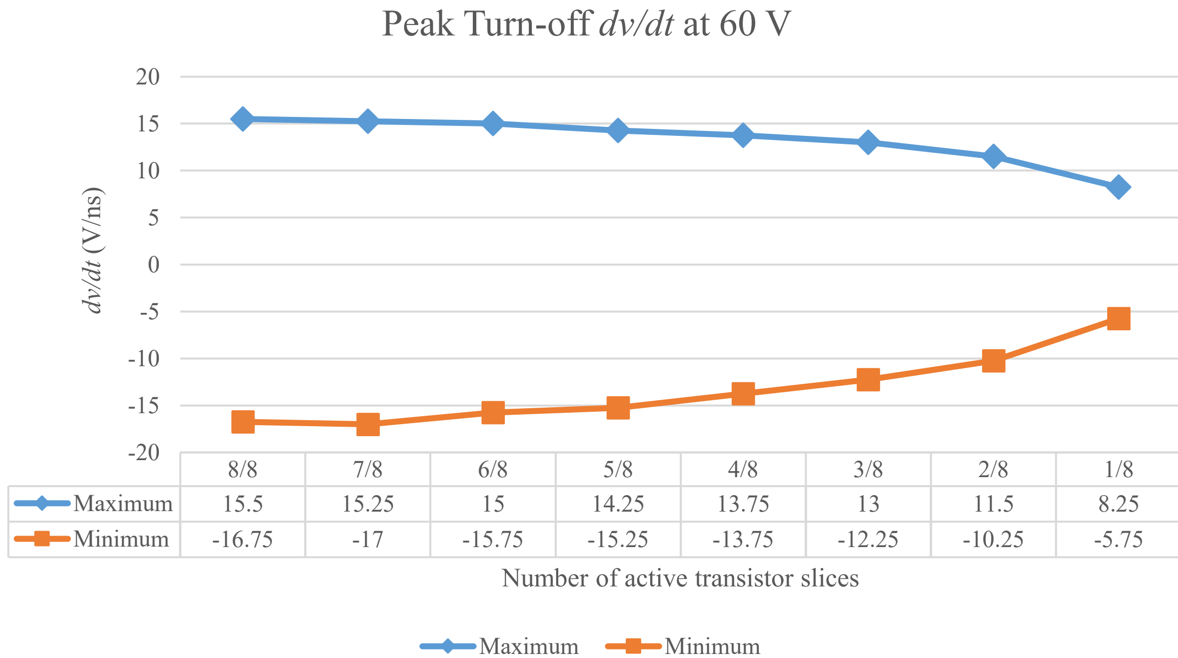


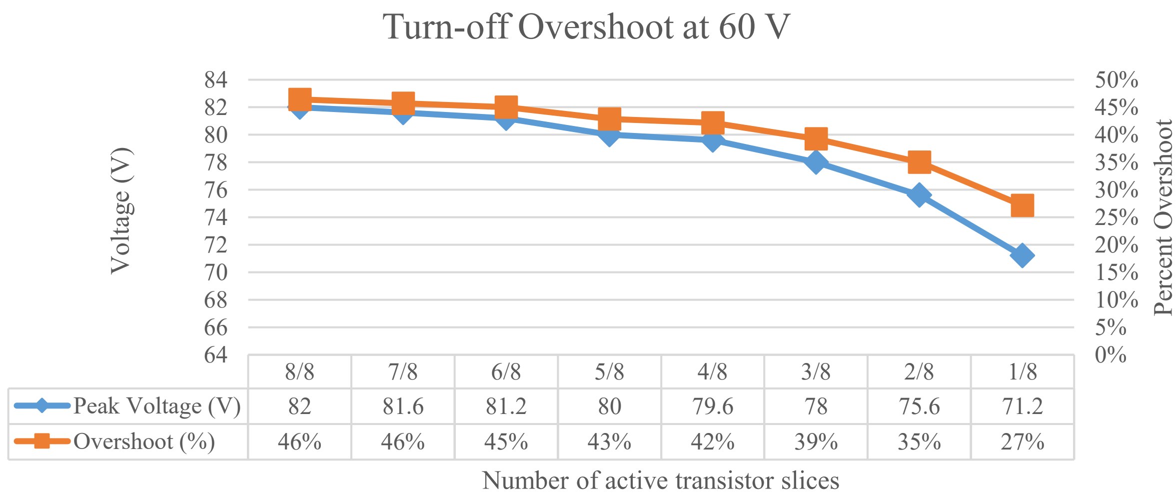
Fig. 7.31. Turn-off waveforms over various drive strengths. The load inductor current is 10 A with a nominal drain voltage of 60 V.



	8/8	7/8	6/8	5/8	4/8	3/8	2/8	1/8
Maximum	15.5	15.25	15	14.25	13.75	13	11.5	8.25
Minimum	-16.75	-17	-15.75	-15.25	-13.75	-12.25	-10.25	-5.75

Fig. 7.32. Turn-off *dv/dt* as a function of drive strength. All waveforms are with a nominal 60 V bus voltage and 10 A load inductor current. The positive *dv/dt* corresponds to the main rising voltage from turn-off, and the negative *dv/dt* matches with the falling voltage after the first drain voltage overshoot.

The overshoot of the system also has good control through using the drive strength, as is visually apparent in Fig. 7.31 and enumerated in Fig. 7.33. The percentage overshoot decreased from 46 % at full drive strength to 27 % at minimum drive strength. The voltage after the overshoot was set to the average voltage immediately after the initial ringing transients dissipated. Immediately after turn-off, the bus voltage drops to 56 V, resulting in a proportionally larger overshoot percentage. Overshoot reduction of this magnitude may be sufficient with a reasonable voltage margin.



	8/8	7/8	6/8	5/8	4/8	3/8	2/8	1/8
Peak Voltage (V)	82	81.6	81.2	80	79.6	78	75.6	71.2
Overshoot (%)	46%	46%	45%	43%	42%	39%	35%	27%

Fig. 7.33. Comparison of overshoot voltages and percent overshoot as a function of drive strength.

Conclusions

Two test regimes were examined: high temperature gate driver operation, and integrated power module operation. Multiple die showed successful operation at temperatures reaching to 450 °C to 530 °C, indicating that the presented silicon carbide gate driver is capable of useful operation over an extended temperature range. Evaluation of the integrated power module demonstrated superior performance during power MOSFET switching transitions up to 300 V

and 20 A. Evaluation of the adjustable drive strength feature show the ability to change the static drive resistance over temperature, as well as control overshoot and peak drain *dv/dt*.

Lessons Learned

Starting with known-good die is a large help when beginning packaging operations. The time and resource expensive process of packaging a bare die is at best frustrating when a die is revealed to be completely dead. Pre-screening of the devices represents a significant effort in reducing test supply waste, and a notable improvement from run 1 to run 2.

High temperature testing is fraught with challenges as temperatures exceed common material ratings. Most high temperature materials used in the gate driver validation were experimentally verified to be more capable of handling higher temperatures than indicated by their specifications. Adjusting the test setup to avoid reliance on material properties at temperature is necessary, and physical forces on wires and parts must also be considered. One standard low-temperature technique of twisting wires together was identified as a failure mode when the insulation melted.

One realization was that a standard FR-4 type PCB with an ENIG finish was a suitable substrate for fine gold ball bonding and thick aluminum wedge bonding. Minor variations in techniques allowed for the use of a low-cost, fine featured PCB for the power module integration testing, which proved advantageous.

8 Conclusions and Future Work

This work presents the design and test of a gate driver that is simple to integrate and matches the temperature requirements of future power modules. This work presents the first integrated SiC CMOS gate driver capable of driving a power MOSFET. The gate driver pushes the functional temperature limits of gate drive technology to above 530 ℃, a new high-water mark for extreme environment electronics.

The gate driver was demonstrated as integrated into a power module. The adjustable drive strength of the driver allows for new levels of gate drive control and flexibility. The control can adjust for changes in external factors without changing the circuit or connectivity, allowing for greater flexibility in direct gate-connected power modules. The demonstrated switching stability shows that the driver has sufficient drive strength and speed to control high-current and high-voltage systems in-situ.

Optimization of the output transistors show a guided approach to large layout optimization. Validation of parametrically-generated layouts allow confidence that the layouts can be manufactured, and represent a verified functionality. Automated performance evaluation over multiple operating conditions, and presentation of the measured performance in a human-understandable format are shown. A process of taking the parametric sweeps translates the test results into a device selection that can be implemented in the final design.

Testability of the gate driver for wafer-level testing is a new priority after poor yield scuttled the first run. Top-level design decisions are outlined in order to guarantee a minimum level of testability at probe. Further test refinement demonstrates a new technique for evaluating transistor health on large power transistors in-situ. This technique is used to evaluate the health

of gate driver die on a wafer and build statistical information on wafer yield with a purely digital interface.

Future Work

As with any significant body of work, there is always more work that could be done. The gate driver itself contains multiple areas where design decisions would be made differently with knowledge gained from run 2. The logic routing in the design placed a priority on minimizing gate count and area, resulting in a higher gate driver propagation delay than preferred. Further refinement is also possible on the logic gate cell layouts, potentially resulting in a smaller logic layout area. The flip-flop used particularly stands to benefit from optimization. The addition of a second metal layer would have a tremendous impact on the design, allowing for much smaller layout areas and more efficient output transistor layouts. The pad cells also need optimization to allow for easier use with high-temperature flip-chip technologies such as sintered nano silver. Adjusting the size and spacing of the input pads for the coarser flip-chip paste is necessary to allow for an entirely flip-chip based (and, thus, wire bond-less) power module.

Further evaluation is possible of the gate driver integrated into a power module. An LTCC or DBC substrate would be used in concert with the SiC gate driver to allow for high-temperature operation unconstrained by the power module substrate. High-temperature and high voltage potting material is needed for full current and voltage demonstration of such a power module.

Bibliography

[1] H. A. Mantooth, M. D. Glover, and P. Shepherd, “Wide Bandgap Technologies and Their Implications on Miniaturizing Power Electronic Systems,” *IEEE J. Emerg. Sel. Top. Power Electron.*, vol. 2, no. 3, pp. 374–385, Sep. 2014.

[2] T. Funaki *et al.*, “Power Conversion with SiC Devices at Extremely High Ambient Temperatures,” in *2005 IEEE 36th Power Electronics Specialists Conference*, 2005, pp. 2030–2035.

[3] E. Cilio, J. Hornberger, B. McPherson, R. Schupbach, and A. Lostetter, “Design and Fabrication of a High Temperature (250 Â°C Baseplate), High Power Density Silicon Carbide (SiC) Multichip Power Module (MCPM) Inverter,” in *IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference on*, 2006, pp. 1822–1827.

[4] J. Hornberger, A. B. Lostetter, K. J. Olejniczak, T. McNutt, S. M. Lal, and A. Mantooth, “Silicon-carbide (SiC) semiconductor power electronics for extreme high-temperature environments,” in *2004 IEEE Aerospace Conference, 2004. Proceedings*, 2004, vol. 4, p. 2538–2555 Vol.4.

[5] N. Kaminski and O. Hilt, “SiC and GaN Devices - Competition or Coexistence?,” in *2012 7th International Conference on Integrated Power Electronics Systems (CIPS)*, 2012, pp. 1–11.

[6] B. J. Baliga, *Fundamentals of Power Semiconductor Devices*. Boston, MA, USA: Springer, 2008.

[7] “Analog/Mixed-Signal Semiconductor Foundry: 1.0 µm - XI10.” [Online]. Available: http://www.xfab.com/en/technology/soi/10-um-xi10.html. [Accessed: 13-Jun-2016].

[8] “Fraunhofer-Institut für Mikroelektronische Schaltungen und Systeme IMS: Detailansicht.” [Online]. Available: http://www.ims.fraunhofer.de/news/detailansicht/article/high-temperature-electronic.html. [Accessed: 13-Jun-2016].

[9] J. Valle-Mayorga *et al.*, “High-Temperature Silicon-on-Insulator Gate Driver for SiC-FET Power Modules,” *IEEE Trans. Power Electron.*, vol. 27, no. 11, pp. 4417–4424, Nov. 2012.

[10] M. A. Huque, S. K. Islam, B. J. Blalock, C. Su, R. Vijayaraghavan, and L. M. Tolbert, “Silicon-on-insulator based high-temperature electronics for automotive applications,” in *IEEE International Symposium on Industrial Electronics, 2008. ISIE 2008*, 2008, pp. 2538–2543.

[11] P. Shepherd, S. C. Smith, J. Holmes, A. M. Francis, N. Chiolino, and H. A. Mantooth, “A robust, wide-temperature data transmission system for space environments,” in *2013 IEEE Aerospace Conference*, 2013, pp. 1–13.

[12] K. Grella *et al.*, “High Temperature Characterization up to 450°C of MOSFETs and Basic Circuits Realized in a Silicon-on-Insulator (SOI) CMOS Technology,” *J. Microelectron. Electron. Packag.*, vol. 10, no. 2, pp. 67–72, Apr. 2013.

[13] “2N7640-GA Datasheet.” .

[14] J. Hornberger *et al.*, “High-temperature integration of silicon carbide (SiC) and silicon-on-insulator (SOI) electronics in multichip power modules (MCPMs),” in *2005 European Conference on Power Electronics and Applications*, 2005, p. 10 pp.-pp.P.10.

[15] N. Ericson *et al.*, “A 4H Silicon Carbide Gate Buffer for Integrated Power Systems,” *IEEE Trans. Power Electron.*, vol. 29, no. 2, pp. 539–542, Feb. 2014.

[16] A. S. Kashyap *et al.*, “Silicon carbide integrated circuits for extreme environments,” in *2013 IEEE Workshop on Wide Bandgap Power Devices and Applications (WiPDA)*, 2013, pp. 60–63.

[17] “Silicon Carbide Power MOSFET C2M Planar MOSFET 1700V 45mOhm | Wolfspeed.” [Online]. Available: http://www.wolfspeed.com/cpm2-1700-0045b. [Accessed: 14-Jun-2016].

[18] “GA100JT12-227 Datasheet.” .

[19] A. Lostetter *et al.*, “High-temperature silicon carbide and silicon on insulator based integrated power modules,” in *IEEE Vehicle Power and Propulsion Conference, 2009. VPPC ’09*, 2009, pp. 1032–1035.

[20] R. L. Greenwell, B. M. McCue, L. M. Tolbert, B. J. Blalock, and S. K. Islam, “High-temperature SOI-based gate driver IC for WBG power switches,” in *2013 Twenty-Eighth Annual IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2013, pp. 1768–1775.

[21] J.-S. Chen, S.-H. Ryu, and K. T. Kornegay, “A silicon carbide CMOS intelligent gate driver circuit,” in *The 1998 IEEE Industry Applications Conference, 1998. Thirty-Third IAS Annual Meeting*, 1998, vol. 2, pp. 963–966 vol.2.

[22] R. R. Lamichhane *et al.*, “A wide bandgap silicon carbide (SiC) gate driver for high-temperature and high-voltage applications,” in *2014 IEEE 26th International Symposium on Power Semiconductor Devices IC’s (ISPSD)*, 2014, pp. 414–417.

[23] S. Kargarrazi, L. Lanni, A. Rusu, and C.-M. Zetterling, “A monolithic SiC drive circuit for SiC Power BJTs,” in *2015 IEEE 27th International Symposium on Power Semiconductor Devices IC’s (ISPSD)*, 2015, pp. 285–288.

[24] D. T. Clark *et al.*, “High Temperature Silicon Carbide CMOS Integrated Circuits,” *Mater. Sci. Forum*, vol. 679–680, pp. 726–729, Mar. 2011.

[25] C. Li, J. S. Duster, and K. T. Kornegay, “A nonvolatile semiconductor memory device in 6H-SiC for harsh environment applications,” *IEEE Electron Device Lett.*, vol. 24, no. 2, pp. 72–74, Feb. 2003.

[26] D. B. Slater, G. M. Johnson, L. A. Lipkin, A. V. Suvorov, and J. W. Palmour, “Demonstration of a 6H-SiC CMOS technology,” in *Device Research Conference, 1996. Digest. 54th Annual*, 1996, pp. 162–163.

[27] J.-S. Chen and K. T. Kornegay, “Design of a silicon carbide CMOS power OPAMP for stable operation at elevated temperatures,” in *, Proceedings of 1997 IEEE International Symposium on Circuits and Systems, 1997. ISCAS ’97*, 1997, vol. 1, pp. 157–160 vol.1.

[28] M. D. Glover *et al.*, “A UVLO Circuit in SiC Compatible With Power MOSFET Integration,” *IEEE J. Emerg. Sel. Top. Power Electron.*, vol. 2, no. 3, pp. 425–433, Sep. 2014.

[29] P. G. Neudeck *et al.*, “Stable Electrical Operation of 6H #x2013;SiC JFETs and ICs for Thousands of Hours at 500,” *IEEE Electron Device Lett.*, vol. 29, no. 5, pp. 456–459, May 2008.

[30] D. J. Spry, P. G. Neudeck, L. Chen, D. Lukco, C. W. Chang, and G. M. Beheim, “Prolonged 500 #x00B0;C Demonstration of 4H-SiC JFET ICs With Two-Level Interconnect,” *IEEE Electron Device Lett.*, vol. 37, no. 5, pp. 625–628, May 2016.

[31] W.-K. Chen, *The VLSI Handbook, Second Edition*. CRC Press, 2016.

[32] E. . Ramsay *et al.*, “Digital and Analogue Integrated Circuits in Silicon Carbide for High Temperature Operation,” *Addit. Conf. Device Packag. HiTEC HiTEN CICMT*, vol. 2012, no. HITEC, pp. 000373–000377, Jan. 2012.

[33] “C3M0065090J 900V SiC MOSFET Power Converter | Wolfspeed.” [Online]. Available: http://www.wolfspeed.com/c3m0065090j. [Accessed: 29-Jun-2016].

[34] “STF12N120K5 Datasheet.” .

[35] “APT37M100L Datasheet.” [Online]. Available: http://www.microsemi.com/existing-parts/parts/60377#docs-specs. [Accessed: 07-Jul-2016].

[36] “IPW90R120C3 Datasheet.” [Online]. Available: http://www.infineon.com/dgdl/Infineon-IPW90R120C3-DS-v01_00-en.pdf?fileId=db3a3043183a955501185000e1d254f2. [Accessed: 07-Jul-2016].

[37] “CMF10120D Datasheet.” [Online]. Available: http://media.digikey.com/pdf/Data%20Sheets/CREE%20Power/CMF10120D.pdf. [Accessed: 07-Jul-2016].

[38] “C2M0160120D 2nd Gen Z-FET 1200-V 160-mΩ SiC MOSFET | Wolfspeed.” [Online]. Available: http://www.wolfspeed.com/c2m0160120d. [Accessed: 07-Jul-2016].

[39] "SCT2160KE | ROHM Semiconductor - ROHM Co., Ltd." [Online]. Available: http://www.rohm.com/web/global/products/-/product/SCT2160KE. [Accessed: 07-Jul-2016].

[40] "GA10JT12-247 Datasheet." [Online]. Available: http://www.genesicsemi.com/images/products_sic/sjt/GA10JT12-247.pdf. [Accessed: 07-Jul-2016].

[41] A. Domurat-Linde and E. Hoene, "Analysis and Reduction of Radiated EMI of Power Modules," in *2012 7th International Conference on Integrated Power Electronics Systems (CIPS)*, 2012, pp. 1–6.

[42] "High Power Gate Driver Board A15IDDJT22-FR4." [Online]. Available: http://www.genesicsemi.com/images/products_sic/sjt/GA15IDDJT22-FR4.pdf. [Accessed: 01-Jul-2016].

[43] P. Anthony, N. McNeill, and D. Holliday, "High-speed resonant gate driver with controlled peak gate voltage for silicon carbide MOSFETs," in *2012 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2012, pp. 2961–2968.

[44] J. Diaz, M. A. Perez, F. M. Linera, and F. Aldana, "A new lossless power MOSFET driver based on simple DC/DC converters," in *, 26th Annual IEEE Power Electronics Specialists Conference, 1995. PESC '95 Record*, 1995, vol. 1, pp. 37–43 vol.1.

[45] I. D. de Vries, "A resonant power MOSFET/IGBT gate driver," in *Seventeenth Annual IEEE Applied Power Electronics Conference and Exposition, 2002. APEC 2002*, 2002, vol. 1, pp. 179–185 vol.1.

[46] D. J. Tschirhart and P. K. Jain, "A dual-channel current source driver for complementary switches," in *2013 Twenty-Eighth Annual IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2013, pp. 1939–1945.

[47] Z. Zhang, F. Wang, L. M. Tolbert, and B. J. Blalock, "A novel gate assist circuit for cross talk mitigation of SiC power devices in a phase-leg configuration," in *2013 Twenty-Eighth Annual IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2013, pp. 1259–1265.

[48] R. Reiner *et al.*, "Fractal structures for low-resistance large area AlGaN/GaN power transistors," in *2012 24th International Symposium on Power Semiconductor Devices and ICs (ISPSD)*, 2012, pp. 341–344.

[49] R. Reiner, F. Benkhelifa, D. Krausse, R. Quay, and O. Ambacher, "Simulation and analysis of low-resistance AlGaN/GaN HFET power switches," in *Proceedings of the 2011-14th European Conference on Power Electronics and Applications (EPE 2011)*, 2011, pp. 1–10.

[50] R. Reiner *et al.*, “Novel Layout and Packaging for Lateral, Low-Resistance GaN-on-Si Power Transistors,” in *2014 8th International Conference on Integrated Power Systems (CIPS)*, 2014, pp. 1–5.

[51] A. D. Sagneri, D. I. Anderson, and D. J. Perreault, “Optimization of Integrated Transistors for Very High Frequency DC #x2013;DC Converters,” *IEEE Trans. Power Electron.*, vol. 28, no. 7, pp. 3614–3626, Jul. 2013.

[52] D. Guilherme, N. Horta, and J. Guilherme, “Automatic layout generation of power MOSFET transistors in bulk CMOS,” in *2014 21st IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2014, pp. 606–609.

[53] “gdspy 0.8.1 : Python Package Index.” [Online]. Available: https://pypi.python.org/pypi/gdspy. [Accessed: 01-Jul-2016].

[54] *Calibre*. Mentor Graphics Corporation, 2013.

[55] *HSPICE*. Synopsys, Inc, 2012.

[56] *NumPy*. [Online].

[57] “Citing matplotlib — Matplotlib 1.5.1 documentation.” [Online]. Available: http://matplotlib.org/citing.html. [Accessed: 01-Jul-2016].

[58] W. Xiaoqing, H. Tamamoto, and K. Kinoshita, “Transistor leakage fault location with IDDQ measurement,” in *Test Symposium, 1995., Proceedings of the Fourth Asian*, 1995, pp. 51–57.

[59] S. Sunter, C. McDonald, and G. Danialy, “Contactless digital testing of IC pin leakage currents,” in *Test Conference, 2001. Proceedings. International*, 2001, pp. 204–210.

[60] S. Y. Huang, Y. H. Lin, L. R. Huang, K. H. Tsai, and W. T. Cheng, “Programmable Leakage Test and Binning for TSVs With Self-Timed Timing Control,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 32, no. 8, pp. 1265–1273, Aug. 2013.

[61] “PyVISA 1.8 : Python Package Index.” [Online]. Available: https://pypi.python.org/pypi/PyVISA. [Accessed: 01-Jul-2016].

[62] “PySide 1.2.4 : Python Package Index.” [Online]. Available: https://pypi.python.org/pypi/PySide/1.2.4. [Accessed: 01-Jul-2016].

[63] Y. A. Cengel, *Heat and Mass Transfer: Fundamentals & Applications*, Fourth edition in SI Units edition. New Delhi: McGraw Hill, 2007.

A. Appendix: Python Code Base

Python Optimization of FETs

batch.py

```
from fet_gen import *
from run_calibre import *
from run_spice import *
import numpy
title_row = ''
data_size = 0

script_run_dir = '/tmp/pcell_gen/'
calibre_run_dir = '/tmp/pcell_gen/calibre_run/'
spice_run_dir = '/tmp/pcell_gen/spice_run/'
source_tb = '/home/mbarlow/automation/pcell_gen/tb.sp'

# Stock Parameter Definitions
w       = 80.0 # Channel width per device
l       = 1.2  # Channel Length
nf      = 30   # Number of fingers in a device
m       = 15   # Multiplicity
wm      = 4.0  # Width of drain/source metal
wp      = 8.0  # Width of vertical poly cell
wc      = 400.0 # Width of metal at edge of contact
wgb     = 14.0 # Width of metal gate ties
channel = 'P'  # FET channel ('N' or 'P')

# Enable Transient Simulation
tran_en = 1


# Check to see if starting directories exist
if not os.path.exists( script_run_dir ):
  print "ERROR - Run Directory DOES NOT EXIST"
if not os.path.exists( calibre_run_dir ):
  os.makedirs( calibre_run_dir )
if not os.path.exists( spice_run_dir ):
  os.makedirs( spice_run_dir )

# Configure X sweep
x_index  = 5
x_series = numpy.arange(10, 41, 2)
xdim = len(x_series)

# Configure Y sweep
y_index  = 6
#y_series = numpy.arange(10,41,3)
y_series = numpy.arange(3.8, 4.4, 0.2)
ydim = len(y_series)

# Pre-populate array with default values
sim_data  = numpy.zeros((ydim,xdim,1,1))
phys_data = numpy.zeros((ydim,xdim,10))
phys_data[...,...,3] = w  * numpy.ones((ydim,xdim))
phys_data[...,...,4] = nf * numpy.ones((ydim,xdim))
phys_data[...,...,5] = m  * numpy.ones((ydim,xdim))
phys_data[...,...,6] = wm * numpy.ones((ydim,xdim))
phys_data[...,...,7] = wp * numpy.ones((ydim,xdim))
```

```
phys_data[...,...,8] = wgb* numpy.ones((ydim,xdim))
phys_data[...,...,9] = wc * numpy.ones((ydim,xdim))

#Populate the x series
x_array = numpy.tile(x_series, [ydim, 1])
phys_data[...,...,x_index] = x_array

#Populate the y series
y_series = y_series.reshape((ydim,1))
y_array = numpy.tile(y_series, [1, xdim])
phys_data[...,...,y_index] = y_array

progress_counter = 0.0

# Define starting range
for y in range(ydim):
  for x in range(xdim):

    # Make sure script starts each loop at the right point
    os.chdir(script_run_dir)

    # Load physical parameters for this runset
    w  = phys_data[y,x,3]
    nf = int(phys_data[y,x,4] )
    m  = int(phys_data[y,x,5] )
    wm = phys_data[y,x,6]
    wp = phys_data[y,x,7]
    wgb= phys_data[y,x,8]
    wc = phys_data[y,x,9]

    progress_counter = progress_counter + 1.0
    print str(100.0*float(progress_counter)/float(xdim*ydim))
    wm_string = str(wm).split('.')

    # Name this iteration's cell
    l_string = '_l%.1f' % l
    wm_string = '_wm%.1f' % wm
    cell_name = channel + 'FET_w'+str(w)+l_string+'_nf'+str(nf)+'_m'+str(m)+\
                wm_string+'_wp'+str(wp)+'_wgb'+str(wgb)
    cell_name = cell_name.replace('.', 'p')
    print cell_name
    # Generate layout and schematic
    top_gds = make_fet_array(w,l,nf,m,wm,wp,wgb,wc,channel,cell_name)
    #calculate area
    boundingbox = top_gds.get_bounding_box()
    boundingbox = boundingbox[1] - boundingbox[0]
    phys_data[y,x,0] = boundingbox[0]
    phys_data[y,x,1] = boundingbox[1]
    phys_data[y,x,2] = boundingbox[1] * boundingbox[0]


    # Output GDSII
    gdspy.gds_print(outfile=cell_name+'.gds',cells=[top_gds], unit=1.0e-6,
precision=1.0e-9)

    # Copy files to Calibre Run Directory
    shutil.copy(cell_name+'.gds', calibre_run_dir)
    shutil.copy(cell_name+'.sp', calibre_run_dir)
    os.chdir(calibre_run_dir)

    # Run DRC
    run_drc(calibre_run_dir + cell_name + '.gds', cell_name)
    # Run LVS
```

```
    run_lvs(calibre_run_dir + cell_name + '.sp', calibre_run_dir + cell_name + '.gds',
cell_name)
    # Run PEX
    run_pex(calibre_run_dir + cell_name + '.sp', calibre_run_dir + cell_name + '.gds',
cell_name)

    #######################
    # Copy PEX netlist to spice_run_dir, change directory
    shutil.copy(cell_name+'.pex.sp', spice_run_dir)
    shutil.copy(cell_name+'.pex.sp.pex', spice_run_dir)
    shutil.copy(cell_name+'.pex.sp.'+ cell_name + '.pxi', spice_run_dir)
    os.chdir(spice_run_dir)

    # Generate SPICE Testbench
    make_tb_file(source_tb, cell_name + '_tb.sp', cell_name+'.pex.sp', cell_name,
channel)
    # Run SPICE
    outfile = run_spice(cell_name + '_tb.sp', cell_name + '_out')

    # import SPICE data

    (new_title_row_dc, data_in_dc) = parse_spice_output(cell_name + '_out.ms0.csv')
    if tran_en == 0:
      new_title_row = new_title_row_dc
      data_in = data_in_dc
    else:
      (new_title_row_tran, data_in_tran) = parse_spice_output(cell_name +
'_out.mt0.csv')
      new_title_row = numpy.concatenate((new_title_row_dc, new_title_row_tran),
axis=0)
      data_in = numpy.hstack((data_in_dc,data_in_tran))

    if len(title_row) <1:
      title_row = new_title_row
    elif not (title_row == new_title_row).all():
    #elif not (title_row == new_title_row):
      print "ERROR - NEW TITLE ROW"

    if data_size == 0:
      data_size = data_in.shape
      sim_data.resize((ydim,xdim,data_size[0], data_size[1]), refcheck=False)
    elif data_size != data_in.shape:
      print "ERROR - DATA MATRIX SIZES DON\'T AGREE"

  #  sim_data.resize((x+1,data_size[0], data_size[1]), refcheck=False)
    sim_data[y,x]=data_in

print title_row
#print sim_data

os.chdir(script_run_dir)
print "saving simulation data"
numpy.save('sim_data', sim_data)
print "saving physical data"
numpy.save('phys_data', phys_data)
```

fet_gen.py

This file is not included due to extensive use of confidential PDK information.

run_calibre.py

```
  #!/usr/bin/python

import os
import shutil
import subprocess
drc_include_file = '/mscad/foundry/raytheon/raysic/Calibre/DRC/calibre.drc'
lvs_include_file = '/mscad/foundry/raytheon/raysic/Calibre/PEX/calibre.pex'
pex_include_file = '/mscad/foundry/raytheon/raysic/Calibre/PEX/calibre.pex'


def run_drc(filename, cell_name):
  print "running Calibre DRC"
#   run_dir = "calibre_run"
#   # Check to see if we are in a "run_dir" directory
#   if os.getcwd().find(run_dir) == -1:
#     # change (or make) calibre run dir
#     if not os.path.exists(run_dir):
#       os.makedirs(run_dir)
#     shutil.copy(filename, run_dir)
#     os.chdir(run_dir)
#   # Build calibre run file
  run_file = cell_name + '.rules.drc'
  summary_file = cell_name + '.sum.drc'
  f = open(run_file, 'w')
  f.write('LAYOUT SYSTEM GDS\n')
  f.write('LAYOUT PATH \"'+filename+'\"\n')
  f.write('LAYOUT PRIMARY "' + cell_name + '"\n')
  f.write('include \"/mscad/foundry/raytheon/raysic/Calibre/DRC/calibre.drc\"\n')
  f.write('DRC SUMMARY REPORT '+summary_file+' REPLACE\n')

  f.close()
  # run calibre
  #print run_file
  #print subprocess.call(["ls"])
  calibre_drc_logfile = open('calibre_drc_stdout.txt', 'w')

  print subprocess.call(["calibre", "-drc", run_file], stdout=calibre_drc_logfile)
  calibre_drc_logfile.close()

  # evaluate results
  f = open( summary_file, 'r')
  # Run through file looking for results line
  for line in f:
    if line.find('TOTAL DRC Results Generated') != -1:
      subline = line.split(':')
      drc_status = int(subline[1])

  os.chdir('..')
  return drc_status

def run_lvs(netlist_file, gds_file, cell_name):
  print "running Calibre LVS"
```

```
  #run_dir = "calibre_run"
  lvs_results = "last.lvs"
#   # Check to see if we are in a "run_dir" directory
#   if os.getcwd().find(run_dir) == -1:
#     # change (or make) calibre run dir
#     if not os.path.exists(run_dir):
#       os.makedirs(run_dir)
#     shutil.copy(netlist_file, run_dir)
#     shutil.copy(gds_file, run_dir)
#     os.chdir(run_dir)
  # Build calibre run file
  run_file = cell_name + '.rules.lvs'
  summary_file = cell_name + '.sum.lvs'
  svdb_file = cell_name + '.svdb'
  pex_netlist_file = cell_name + '.pex.sp'
  f = open(run_file, 'w')
  f.write('LAYOUT SYSTEM GDS\n')
  f.write('LAYOUT PATH "'+gds_file+'"\n')
  f.write('LAYOUT PRIMARY "' + cell_name + '"\n')
  f.write('SOURCE SYSTEM SPICE\n')
  f.write('SOURCE PATH "'+netlist_file+'"\n')
  f.write('SOURCE PRIMARY "' + cell_name + '"\n')
  f.write('MASK SVDB DIRECTORY "'+ svdb_file +'" QUERY XRC\n')
  #f.write('LVS REPORT "last.lvs"\n')
  #f.write('PEX NETLIST  ' + pex_netlist_file + ' HSPICE 1 SOURCENAMES\n')
  #f.write('LVS RECOGNIZE GATES NONE\n')
  f.write('VIRTUAL CONNECT COLON NO\n')
  f.write('VIRTUAL CONNECT REPORT NO\n')
  f.write('include \"' + pex_include_file + '\"\n')
  f.close()
  # run calibre
  calibre_lvs_logfile = open('calibre_lvs_stdout.txt','w')
  retval =  subprocess.call(["calibre", "-lvs", "-hier", '-spice '+svdb_file
+'/'+cell_name+'.sp', run_file], stdout=calibre_lvs_logfile)
  calibre_lvs_logfile.close()

  if retval != 0:
    return retval

  f = open(lvs_results, 'r')
  correct_count = 0
  incorrect_count = 0
  for line in f:
    if line.find("CORRECT") >= 0:
      correct_count = correct_count + 1
    if line.find("INCORRECT") >= 0:
      correct_count = correct_count - 1
      incorrect_count = incorrect_count + 1

  if incorrect_count == 0 and correct_count == 0:
    # Something is wrong with LVS execution (error)
    print "LVS EXECUTION ERROR"
    return -1
  elif incorrect_count > 0:
    # LVS ran successfully, and the schematic != layout
    print "LVS INCORRECT", incorrect_count
    lvs_status = incorrect_count
  elif correct_count > 0 and incorrect_count == 0:
    # LVS ran successfully, and schematic and layout match
    print "LVS CORRECT"
    lvs_status = 0
  else:
    # Catch-all error
```

```
    print "SCRIPT PARSING ERROR"
    lvs_status = -1

  return lvs_status

def run_pex(netlist_file, gds_file, cell_name):
  print "running Calibre PEX"
  run_dir = "calibre_run"
  lvs_results = "last.lvs"
#   # Check to see if we are in a "run_dir" directory
#   if os.getcwd().find(run_dir) == -1:
#     # change (or make) calibre run dir
#     if not os.path.exists(run_dir):
#       os.makedirs(run_dir)
#     shutil.copy(netlist_file, run_dir)
#     shutil.copy(gds_file, run_dir)
#     os.chdir(run_dir)
  # Build calibre run file
  run_file = cell_name + '.rules.pex'
  summary_file = cell_name + '.sum.pex'
  svdb_file = cell_name + '.svdb'
  pex_netlist_file = cell_name + '.pex.sp'
  f = open(run_file, 'w')
  f.write('LAYOUT SYSTEM GDS\n')
  f.write('LAYOUT PATH "'+gds_file+'"\n')
  f.write('LAYOUT PRIMARY "' + cell_name + '"\n')
  f.write('SOURCE SYSTEM SPICE\n')
  f.write('SOURCE PATH "'+netlist_file+'"\n')
  f.write('SOURCE PRIMARY "' + cell_name + '"\n')
  f.write('MASK SVDB DIRECTORY "'+ svdb_file +'" QUERY XRC\n')
  f.write('include \"' + pex_include_file + '\"\n')
  f.write('PEX NETLIST  "' + pex_netlist_file + '" HSPICE 1 SOURCENAMES\n')
  #f.write('LVS RECOGNIZE GATES NONE\n')
  f.write('VIRTUAL CONNECT COLON NO\n')
  f.write('VIRTUAL CONNECT REPORT NO\n')
  f.close()
  # run calibre, assuming the LVS extraction has happened
  calibre_pex_logfile = open('calibre_pex_stdout.txt','w')
  retval = subprocess.call(["calibre", "-xrc", "-pdb", "-rcc", "-turbo 1", run_file],
stdout=calibre_pex_logfile)
  if retval != 0:
    return retval
  calibre_pex_logfile.write('************* Starting Export *************\n')

  retval = subprocess.call(["calibre", "-xrc", "-fmt", "-all", run_file],
stdout=calibre_pex_logfile)
  if retval != 0:
    return retval
  calibre_pex_logfile.close()
```

Run_spice.py

```
#!/usr/bin/python

#This script runs a HSPICE testbench, and loads the results into a numpy array
import csv, numpy, subprocess

def parse_spice_output(output_file):
  with open(output_file, 'r') as f:
    reader = csv.reader(f, delimiter=',')
    rowlength = 0
    numrows = 0
    for row in reader:
      if len(row) > 1 and row[1].isalpha():
        title_row = row
        rowlength = len(row)
        data = numpy.zeros((1,rowlength))

      elif len(row) > 1:
        numrows = numrows + 1
        data.resize((numrows, rowlength), refcheck=False)
        data[numrows-1] = row

  return( title_row, data)

def run_spice(run_file, out_file):
  print "Running HSPICE"
  hspice_logfile = open('hspice_stdout.txt', 'w')
  retval = subprocess.call(["hspice", '-i '+run_file, '-mp', '-o '+out_file],
stdout=hspice_logfile)
  hspice_logfile.close()

  print retval

def make_tb_file(source_tb, destination_tb, include_file, cell_name, channel):

  if channel == 'P':
    symbol = '-'
  else:
    symbol = ''

  with open(destination_tb, 'w') as outfile:
    with open(source_tb, 'r') as infile:
      for line in infile:
        if line.find('*NETLIST*') >= 0:
          #Insert Netlist
          outfile.write('.include '+include_file+'\n')
          #outfile.write('rg 1 3 200\n')
          outfile.write('rg 1 3 20\n')
          outfile.write('vg 3 0 dc '+symbol+'15 PULSE (0 '+symbol+'15 0 1n 1n 1 2)\n')
          #outfile.write('vg 3 0 dc '+symbol+'12 PULSE (0 '+symbol+'12 0 1n 1n 1
2)\n')
          if channel == 'P':
            #outfile.write('vs 0 2 dc 1.0\n')
            outfile.write('vs 0 2 dc 15\n')
          else:
            outfile.write('vs 2 0 dc 15\n')
          outfile.write('xi0 1 0 2 '+cell_name+'\n')
        else:
          outfile.write(line)
```

```
# bigarray = numpy.zeros((1,1,1))
# title_row = ''
# data_size = 0
# for x in range(10):
#   (new_title_row,data_in) = parse_spice_output('calibre_run/spice/outfile.ms0.csv')
#   if len(title_row) <1:
#     title_row = new_title_row
#   elif title_row != new_title_row:
#     print "ERROR - NEW TITLE ROW"
#
#   if data_size == 0:
#     data_size = data_in.shape
#     bigarray.resize((x,data_size[0], data_size[1]), refcheck=False)
#   elif data_size != data_in.shape:
#     print "ERROR - DATA MATRIX SIZES DON\"T AGREE"
#
#   bigarray.resize((x+1,data_size[0], data_size[1]), refcheck=False)
#   bigarray[x]=data_in

# print bigarray
```

Process_data.py

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
# Ident index          0        1       2      3    4     5    6     7     8      9
phys_data_ident = [ 'xsize', 'ysize', 'area', 'w', 'nf', 'm', 'wm', 'wp', 'wgb', 'wc'
]
axis_labels = [ 'ERROR 0', 'ERROR 1', 'ERROR 2', r'FET Width ($\mu$m)', 'Number of
Fingers',\
                'Multiplicity', r'Metal Width ($\mu$m)', r'Poly Width ($\mu$m)',\
                r'Gate Bus Width ($\mu$m)', r'End Contact Width ($\mu$m)']

root_dir = '/tmp/pcell_gen/' #default dir for latest run is /tmp/pcell_gen
#root_dir = 'results/pfet_w_v_m/' #default dir for latest run is /tmp/pcell_gen
phys_data = np.load(root_dir + 'phys_data.npy')
sim_data = np.load(root_dir + 'sim_data.npy')

# Identify if data includes transient measurements
sim_dim=sim_data.shape
print sim_dim
if sim_dim[3] < 6: #No transient data
  print "No transient simulation data detected"
  tran_en = 0 #disable transient plotting
else:
  tran_en = 1 #enable

# Identify Axes and generate series text
series_string = ''
for index in range(3,10):
  if phys_data[0,0,index] != phys_data[0,1,index]:
    x_index = index
    series_string = series_string + phys_data_ident[index] + '(' +
str(np.amin(phys_data[...,...,index])) \
                    + '-' + str(np.amax(phys_data[...,...,index])) + ') '
  elif phys_data[0,0,index] != phys_data[1,0,index]:
    y_index = index
    series_string = series_string + phys_data_ident[index] + '(' +
str(np.amin(phys_data[...,...,index])) \
                    + '-' + str(np.amax(phys_data[...,...,index])) + ') '
  else:
    series_string = series_string + phys_data_ident[index] + str(phys_data[0,0,index])
+ ' '

print x_index, y_index, series_string

y_axis = phys_data[...,...,y_index]
ymin = np.min(y_axis)
ymax = np.max(y_axis)
#x_axis =np.multiply(phys_data[...,...,5] , phys_data[...,...,3])
x_axis =phys_data[...,...,x_index]
xmin = np.min(x_axis)
xmax = np.max(x_axis)


xlabel = axis_labels[x_index]
ylabel = axis_labels[y_index]

(x,y,z) = phys_data.shape
height = np.tile(np.resize(phys_data[...,...,1],(x,y,1)),(1,1,6))
area = np.tile(np.resize(phys_data[...,...,2],(x,y,1)),(1,1,6))
max_current = -1 * sim_data[...,...,...,0]
```

```
if tran_en == 1:
  t_on = 1000000000 * sim_data[...,...,...,5]
fet_w =
np.tile(np.resize(np.multiply(np.multiply(phys_data[...,...,3],phys_data[...,...,4]),p
hys_data[...,...,5]),(x,y,1)),(1,1,6))

#number of plots
if tran_en == 1:
  n=4
else:
  n=2

# Processed data container
(a,b,c,d) = sim_data.shape
plot_data = np.zeros([n,a,b,c])
filename = []
suptitle = []
cb_label = []

# Subtitles (model runs)
#subtitles = str(sim_data[0,0,counter,3])+r'$^\circ$C'

# Data Series Mapping

# Figure 1 - Current Normalized by Height
plot_data[0] = 1000.0 * max_current / height
filename.append('max_current.png')
suptitle.append('Maximum Current (Normalized by Height)')
cb_label.append('Current (A/mm)')

# Figure 2 - Current Normalized by Area
plot_data[1] = 1000000.0 * max_current / area
filename.append('current_density.png')
suptitle.append('Maximum Current (Normalized by Area)')
cb_label.append('Current (A/mm^2)')

if tran_en == 1:
  # Figure 3 - Turn-on time
  plot_data[2] = 1000000000 * sim_data[...,...,...,5]
  filename.append('turn_on_time.png')
  suptitle.append('Turn-on speed')
  cb_label.append('Time to 90% Imax (ns)')

  # Figure 4 - Figure of Merit
  plot_data[3] = np.divide(sim_data[...,...,...,5], fet_w )
  filename.append('fom.png')
  suptitle.append('Figure of Merit (ton*Imax)')
  cb_label.append('Time to 90% Imax (ns)')

# Plotting Loop Core
for pn in range(n):
  fig2, axes = plt.subplots(nrows=3, ncols=2, figsize=(10,10))
  fig2.subplots_adjust( hspace=0.4 )
  # Load dataset
  data = plot_data[pn]
  # Set colormap to be consistent across all plots
  minval = np.amin(data)
  maxval = np.amax(data)
  zscale = np.linspace(minval, maxval, 50)

  # Iterate through each plot
  counter_index = 0
  for ax in axes.flat:
```

```
    im = ax.contourf(x_axis, y_axis, data[...,...,counter_index], zscale)
    #
    ax.set_title(str(sim_data[0,0,counter_index,3])+r'$^\circ$C')
    ax.set_xlabel(xlabel)
    ax.set_ylabel(ylabel)
    ax.set_ylim(ymin, ymax)
    ax.set_xlim(xmin, xmax)
    (xi,yi) = np.unravel_index(np.argmax(data[...,...,counter_index]),
(data[...,...,0]).shape)
    x = str(x_axis[xi,yi])
    y = str(y_axis[xi,yi])
    value = str(data[xi,yi,counter_index])
    ax.annotate('max = '+value+' at \n'+phys_data_ident[x_index] +' = '+x+' and
'+phys_data_ident[y_index] +' = '+y, xy=(x,y), xycoords='data',
                xytext=(0,0), textcoords='offset points',
horizontalalignment='center')
    counter_index = counter_index + 1

  fig2.subplots_adjust(right=0.8)
  cbar_ax = fig2.add_axes([0.85, 0.15, 0.05, 0.7])
  ax_cb = fig2.colorbar(im, cax=cbar_ax)
  ax_cb.set_label(cb_label[pn])
  fig2.suptitle(suptitle[pn]+'\n'+series_string, fontsize=20)
  plt.savefig(filename[pn] )
# END of loop
```

Gate Driver Test Interface

Gate_driver.py

```
import sys
from PySide import QtGui
from gpib.sources import tekAFG3022B
from gpib.scopes import tektds
import numpy as np
import visa
import time
import csv

from gd_gui_ui import Ui_Dialog
# Equipment Use:
#   15V Power Supply (Not connected
#   2x Tek 3022 AFG
#   3-4 channel scope (4 preferable, 2 is workable)

awg1_visa_address = "GPIB::11::INSTR"
awg2_visa_address = "GPIB::12::INSTR"
awg1_visa_address = "USB0::0x0699::0x0347::C031330::INSTR"
awg2_visa_address = "USB0::0x0699::0x0347::C031326::INSTR"

class wf_db:
    def __init__(self):
        self.tdi = ''
        self.tclk = ''
        self.ten = ''

class MainControl(QtGui.QWidget):

    def __init__(self, parent=None):
        QtGui.QWidget.__init__(self, parent)

        self.ui = Ui_Dialog()
        self.ui.setupUi(self)

        #Populate Fields
        self.ui.testClkFreq.setText("100")
        self.ui.testRunLength.setText("10")
        self.ui.testVoltage.setText("15")
        self.ui.testRunType.addItem("Default")

        #Connect function calls
        self.ui.buttonInitVisa.clicked.connect(self.init_VISA)
        self.ui.buttonRunTest.clicked.connect(self.run_test)
        self.ui.buttonSaveData.clicked.connect(self.getData)
        self.ui.testDataNfet.setText("00")
        self.ui.testDataPfet.setText("00")
        self.ui.testDataCtrl.setText("1")
        self.ui.testRunType.currentIndexChanged.connect(self.preset_test_data)



    def preset_test_data(self):
        combo_value = self.ui.testRunType.currentText()
        if combo_value == "Gate - Drain":
            print "gd test"
            self.ui.testDataNfet.setText("00")
```

```
            self.ui.testDataPfet.setText("00")
            self.ui.testDataCtrl.setText("1")
        elif combo_value == "Gate - Source":
            print "gs test"
            self.ui.testDataNfet.setText("FF")
            self.ui.testDataPfet.setText("FF")
            self.ui.testDataCtrl.setText("4")
        else:
            print "unidentified, no change"

    '''
    Builds the waveforms for each AWG channel
    '''
    def build_waveforms(self,serial_data, test_time, clk_freq, test_enable):


        # Quick and dirty timebase approximation
        if clk_freq >= 100e3:
            timebase = 100e-9
        elif clk_freq >= 10e3:
            timebase = 1e-6
        elif clk_freq >= 1e3:
            timebase = 10e-6
        elif clk_freq >= 100:
            timebase = 100e-6
        elif clk_freq >10:
            timebase = 1e-3
        else:
            timebase = 1e-2

        quarter_period = int(1/(clk_freq*4*timebase))
        test_length = int(test_time / (timebase * 1e6))

        #generate binary string from initial hex data
        bin_data = bin(int(serial_data, 16))[2:].zfill(20)

        # Initial state: 10 counts of nothing to stabilize system
        initial_wait = 2
        wf = wf_db()
        wf.tdi  = np.zeros(initial_wait, dtype=np.int)
        wf.tclk = np.zeros(initial_wait, dtype=np.int)
        wf.ten  = np.zeros(initial_wait, dtype=np.int)

        # Clock in data
        for x in range(len(bin_data)):
            current_bit = bin_data[x]
            wf.tdi  = np.append(wf.tdi,  int(current_bit) * np.ones(quarter_period *
4, dtype=np.int))
            wf.tclk = np.append(wf.tclk, np.ones(quarter_period * 2, dtype=np.int))
            wf.tclk = np.append(wf.tclk, np.zeros(quarter_period * 2, dtype=np.int))
            wf.ten  = np.append(wf.ten,  np.zeros(quarter_period * 4, dtype=np.int))

        # Perform Test
        hold_before_test = 8 * quarter_period # Wait for things to settle
        wf.tclk = np.append(wf.tclk, np.ones(hold_before_test, dtype=np.int))
        wf.tdi  = np.append(wf.tdi,  np.zeros(hold_before_test, dtype=np.int))
        wf.ten  = np.append(wf.ten,  np.zeros(hold_before_test, dtype=np.int))

        # Deactivate tristate buffers
        wf.tclk = np.append(wf.tclk, np.ones(test_length, dtype=np.int))
        wf.tdi  = np.append(wf.tdi,  np.zeros(test_length, dtype=np.int))
        wf.ten  = np.append(wf.ten,  test_enable * np.ones(test_length, dtype=np.int))
```

```
        # Capture Data
        wf.tclk = np.append(wf.tclk, np.zeros(quarter_period * 2, dtype=np.int))
        wf.tclk = np.append(wf.tclk, np.ones(quarter_period * 6, dtype=np.int))
        wf.tdi  = np.append(wf.tdi,  np.zeros(quarter_period * 8, dtype=np.int))
        wf.ten  = np.append(wf.ten,  test_enable * np.ones(quarter_period * 2,
dtype=np.int))
        wf.ten  = np.append(wf.ten,  np.zeros(quarter_period * 6, dtype=np.int))

        # Shift out data
        for x in range(len(bin_data)-1):
            wf.tdi  = np.append(wf.tdi,  np.zeros(quarter_period * 4, dtype=np.int))
            wf.tclk = np.append(wf.tclk, np.ones(quarter_period * 2, dtype=np.int))
            wf.tclk = np.append(wf.tclk, np.zeros(quarter_period * 2, dtype=np.int))
            wf.ten  = np.append(wf.ten,  np.zeros(quarter_period * 4, dtype=np.int))

        # Wait after test
        hold_after_test = 2 # Wait for things to settle
        wf.tclk = np.append(wf.tclk, np.zeros(hold_after_test, dtype=np.int))
        wf.tdi  = np.append(wf.tdi,  np.zeros(hold_after_test, dtype=np.int))
        wf.ten  = np.append(wf.ten,  np.zeros(hold_after_test, dtype=np.int))

        #scale waveforms to maximum values
        max_val = 16382
        wf.tclk = max_val * wf.tclk
        wf.tdi  = max_val * wf.tdi
        wf.ten  = max_val * wf.ten

        return [wf, timebase]

    def run_test(self):

        [clock_freq, test_time, supply_voltage, site_string, test_data, test_enable] =
self.validate_forms()
        print test_data



        [wf, samplerate] = self.build_waveforms(test_data, test_time, clock_freq,
test_enable )
        print "length of tclk: ", len(wf.tclk)
        print "sample rate: ", samplerate

        vdd = supply_voltage # Vdd for internal calculations

        self.ui.fileName.setText(site_string)

        voffset = vdd - 10 # Power Supply Offset
        if voffset < 0: # If less than 10V, then an offset isn't necessary
            voffset = 0
        vlow = -voffset # Negative supply for function generator
        vhigh = vdd - abs(vlow)
        # GPIB power up supplies
        # Set +25V channel to +15V
        # Set +6V channel to Vdd-10V

        #Configure AWG Channel 1 - Test Clock
        self.awg1_obj.resetArbEMEM(self.awg1, len(wf.tclk))
        self.awg1_obj.writeArbTrace(self.awg1, wf.tclk.tolist())
        self.awg1_obj.copyArbMem(self.awg1, "EMEM", "USER1")

        self.awg1_obj.setShape(self.awg1, 1, "USER1")
        self.awg1_obj.setZout(self.awg1, 1, "INF")
        self.awg1_obj.setVout(self.awg1, 1, vlow, vhigh)
```

```
        print "length of tclk: ", len(wf.tclk)
        self.awg1_obj.setFreq(self.awg1, 1, 1.0/(samplerate*len(wf.tclk)))

        #Configure AWG Channel 2 - Test Enable
        self.awg1_obj.resetArbEMEM(self.awg1, len(wf.ten))
        self.awg1_obj.writeArbTrace(self.awg1, wf.ten.tolist())
        self.awg1_obj.copyArbMem(self.awg1, "EMEM", "USER2")

        self.awg1_obj.setShape(self.awg1, 2, "USER2")
        self.awg1_obj.setZout(self.awg1, 2, "INF")
        self.awg1_obj.setVout(self.awg1, 2, vlow, vhigh)
        self.awg1_obj.setFreq(self.awg1, 2, 1.0/(samplerate*len(wf.ten)))

        #Configure AWG Channel 3 - Test Data Input
        self.awg2_obj.resetArbEMEM(self.awg2, len(wf.tdi))
        self.awg2_obj.writeArbTrace(self.awg2, wf.tdi.tolist())
        self.awg2_obj.copyArbMem(self.awg2, "EMEM", "USER3")

        self.awg2_obj.setShape(self.awg2, 1, "USER3")
        self.awg1_obj.setZout(self.awg2, 1, "INF")
        self.awg2_obj.setVout(self.awg2, 1, vlow, vhigh)
        self.awg2_obj.setFreq(self.awg2, 1, 1.0/(samplerate*len(wf.tdi)))




        self.awg1_obj.setTrigger(self.awg1, "EXT", "POS", 1)
        self.awg2_obj.setTrigger(self.awg2, "EXT", "POS", 1)

        self.awg1_obj.setBurst(self.awg1, 1, "ON", 1, "TRIG")
        self.awg1_obj.setBurst(self.awg1, 2, "ON", 1, "TRIG")
        self.awg1_obj.setBurst(self.awg2, 1, "ON", 1, "TRIG")

        self.awg1_obj.setOutState(self.awg1, 1, "ON")
        self.awg1_obj.setOutState(self.awg1, 2, "ON")
        self.awg1_obj.setOutState(self.awg2, 1, "ON")


    def validate_forms(self):
        print "Running validate_forms"
        # Clock Frequency Validation

        clock_freq = self.ui.testClkFreq.text()
        try:
            clock_freq = float(clock_freq)
        except:
            print "ERROR: Could not convert clock frequency to a float"
            return [-1, -1, -1, "Error", 0, 0]
        clock_freq = clock_freq * 1000
        if clock_freq < 1 or clock_freq > 2.5e6:
            print "ERROR: Clock frequency out of range"
            clock_freq = -1

        # Test Time Validation
        test_time = self.ui.testRunLength.text()
        try:
            test_time = float(test_time)
        except:
            print "ERROR: Could not convert test time to a float"
            return [clock_freq, -1, -1, "Error", 0, 0]
        if test_time < 1 or test_time > 1e6:
            print "ERROR: Test time out of range"
```

```
            test_time = -1

        # Supply Voltage
        supply_voltage = self.ui.testVoltage.text()
        try:
            supply_voltage = float(supply_voltage)
        except:
            print "ERROR: Could not convert supply voltage to a float"
            return [clock_freq, test_time, -1, "Error", 0, 0]
        if supply_voltage < 0 or supply_voltage > 20:
            print "ERROR: supply voltage out of range"
            supply_voltage = -1

        # Site Identification
        row = self.ui.waferSiteRow.currentText()
        column = self.ui.waferSiteColumn.currentText()
        wafer = self.ui.waferSelection.currentText()
        site_string = "R" + row + "C" + column + "_" + wafer
        print "site has not been validated, only processed"

        # Data Packet creation
        # No validation!!!
        test_data = self.ui.testDataPfet.text() + self.ui.testDataCtrl.text() +
self.ui.testDataNfet.text()

        # Test enable
        if self.ui.testData.isChecked():
            test_enable = 1
        else:
            test_enable = 0

        return [clock_freq, test_time, supply_voltage, site_string, test_data,
test_enable ]

    def init_VISA(self):

        #Initialize AWG1 - AFG3022B
        try:
            self.awg1 = visa.instrument(awg1_visa_address)
            self.awg1_obj = tekAFG3022B()
            self.awg1_obj.setupSource(self.awg1)
        except:
            print "Error initializing AWG1"
            self.awg1 = 'unavailable'

        #Initialize AWG2 - AFG3022B
        try:
            self.awg2 = visa.instrument(awg2_visa_address)
            self.awg2_obj = tekAFG3022B()
            self.awg2_obj.setupSource(self.awg2)
        except:
            print "Error initializing AWG2"
            self.awg2 = 'unavailable'

        try:
            self.oscope1 = visa.instrument("USB0::0x0699::0x0401::C002333::INSTR")
            #self.oscope1 = visa.instrument("TCPIP0::130.184.163.236::inst0::INSTR")
            #self.oscope1 = visa.instrument("TCPIP0::mso4034-
04wt59.uark.edu.ddns.uark.edu::inst0::INSTR")
            self.oscope1_obj = tektds()
            self.oscope1_obj.setupScope(self.oscope1)
        except:
            print "Error initializing Scope 1"
```

```
            self.oscope1 = 'unavailable'

        self.awg1_obj.setBurst(self.awg1, 1, "ON", 1, "TRIG")
        self.awg1_obj.setBurst(self.awg1, 2, "ON", 1, "TRIG")
        self.awg1_obj.setTrigger(self.awg1, "EXT", "POS", 1)

        self.awg1_obj.setOutState(self.awg1, 1, "OFF")
        self.awg1_obj.setOutState(self.awg1, 2, "OFF")

        self.awg2_obj.setBurst(self.awg2, 1, "ON", 1, "TRIG")
        self.awg2_obj.setTrigger(self.awg2, "EXT", "POS", 1)

        self.awg2_obj.setOutState(self.awg2, 1, "OFF")

    def getData(self):
        print "Grabbing Oscilloscope Data"
        if self.oscope1 != 'unavailable':
            self.oscope1_obj.readValues(self.oscope1)

        output_file="out%s.csv" %(time.time())
        output_file = self.ui.fileName.text() + output_file
        file_handle = open(output_file,'wb')
        datawriter = csv.writer( file_handle)
        datawriter.writerow(['Gate Driver test data, v1.5'])
        datawriter.writerow(["clock_freq", "test_time", "supply_voltage",
"site_string", "test_data", "test_enable" ])
        datawriter.writerow(self.validate_forms())
        if self.oscope1 != 'unavailable':
            row = ['time1',].append( self.oscope1_obj.active_channels)

            #datawriter.writerow('time1', scope.channels)

            for i in
range(len(self.oscope1_obj.getTimesCh(ch=self.oscope1_obj.active_channels[0]))):

                row = [
self.oscope1_obj.getTimesCh(ch=self.oscope1_obj.active_channels[0])[i] ]
                for channel in self.oscope1_obj.active_channels:
                    row.append(self.oscope1_obj.getValuesCh(ch=channel)[i])
                datawriter.writerow( row )

        file_handle.close()
        print "Finished Writing Data"


if __name__ == '__main__':
    app = QtGui.QApplication(sys.argv)
    convert = MainControl()
    convert.show()
    sys.exit(app.exec_())
```

gd_gui_ui.py

```
 # Form implementation generated from reading ui file
'C:\Users\mbarlow\workspace_PyDev\GPIB_Control\src\projects_bic\gd_gui.ui'
#
# Created: Sat May 30 14:10:38 2015
#      by: pyside-uic 0.2.14 running on PySide 1.1.2
#
# WARNING! All changes made in this file will be lost!

from PySide import QtCore, QtGui

class Ui_Dialog(object):
    def setupUi(self, Dialog):
        Dialog.setObjectName("Dialog")
        Dialog.resize(513, 687)
        self.formLayout_3 = QtGui.QFormLayout(Dialog)

self.formLayout_3.setFieldGrowthPolicy(QtGui.QFormLayout.AllNonFixedFieldsGrow)
        self.formLayout_3.setObjectName("formLayout_3")
        self.label_16 = QtGui.QLabel(Dialog)
        font = QtGui.QFont()
        font.setPointSize(10)
        font.setWeight(75)
        font.setBold(True)
        self.label_16.setFont(font)
        self.label_16.setAlignment(QtCore.Qt.AlignCenter)
        self.label_16.setObjectName("label_16")
        self.formLayout_3.setWidget(0, QtGui.QFormLayout.SpanningRole, self.label_16)
        self.label_13 = QtGui.QLabel(Dialog)
        self.label_13.setObjectName("label_13")
        self.formLayout_3.setWidget(1, QtGui.QFormLayout.LabelRole, self.label_13)
        self.waferSelection = QtGui.QComboBox(Dialog)
        self.waferSelection.setObjectName("waferSelection")
        self.waferSelection.addItem("")
        self.waferSelection.addItem("")
        self.waferSelection.addItem("")
        self.formLayout_3.setWidget(1, QtGui.QFormLayout.FieldRole,
self.waferSelection)
        self.label_14 = QtGui.QLabel(Dialog)
        self.label_14.setObjectName("label_14")
        self.formLayout_3.setWidget(2, QtGui.QFormLayout.LabelRole, self.label_14)
        self.waferSiteColumn = QtGui.QComboBox(Dialog)
        self.waferSiteColumn.setObjectName("waferSiteColumn")
        self.waferSiteColumn.addItem("")
        self.waferSiteColumn.addItem("")
        self.waferSiteColumn.addItem("")
        self.waferSiteColumn.addItem("")
        self.formLayout_3.setWidget(2, QtGui.QFormLayout.FieldRole,
self.waferSiteColumn)
        self.label_15 = QtGui.QLabel(Dialog)
        self.label_15.setObjectName("label_15")
        self.formLayout_3.setWidget(3, QtGui.QFormLayout.LabelRole, self.label_15)
        self.waferSiteRow = QtGui.QComboBox(Dialog)
        self.waferSiteRow.setObjectName("waferSiteRow")
        self.waferSiteRow.addItem("")
        self.waferSiteRow.addItem("")
        self.waferSiteRow.addItem("")
        self.waferSiteRow.addItem("")
        self.waferSiteRow.addItem("")
        self.waferSiteRow.addItem("")
```

```
        self.waferSiteRow.addItem("")
        self.formLayout_3.setWidget(3, QtGui.QFormLayout.FieldRole, self.waferSiteRow)
        self.label_17 = QtGui.QLabel(Dialog)
        self.label_17.setObjectName("label_17")
        self.formLayout_3.setWidget(4, QtGui.QFormLayout.LabelRole, self.label_17)
        self.fileName = QtGui.QLineEdit(Dialog)
        self.fileName.setObjectName("fileName")
        self.formLayout_3.setWidget(4, QtGui.QFormLayout.FieldRole, self.fileName)
        self.label_18 = QtGui.QLabel(Dialog)
        font = QtGui.QFont()
        font.setPointSize(10)
        font.setWeight(75)
        font.setBold(True)
        self.label_18.setFont(font)
        self.label_18.setObjectName("label_18")
        self.formLayout_3.setWidget(5, QtGui.QFormLayout.SpanningRole, self.label_18)
        self.label_19 = QtGui.QLabel(Dialog)
        self.label_19.setObjectName("label_19")
        self.formLayout_3.setWidget(6, QtGui.QFormLayout.LabelRole, self.label_19)
        self.testClkFreq = QtGui.QLineEdit(Dialog)
        self.testClkFreq.setObjectName("testClkFreq")
        self.formLayout_3.setWidget(6, QtGui.QFormLayout.FieldRole, self.testClkFreq)
        self.label_20 = QtGui.QLabel(Dialog)
        self.label_20.setObjectName("label_20")
        self.formLayout_3.setWidget(7, QtGui.QFormLayout.LabelRole, self.label_20)
        self.testRunLength = QtGui.QLineEdit(Dialog)
        self.testRunLength.setObjectName("testRunLength")
        self.formLayout_3.setWidget(7, QtGui.QFormLayout.FieldRole,
self.testRunLength)
        self.label_24 = QtGui.QLabel(Dialog)
        self.label_24.setObjectName("label_24")
        self.formLayout_3.setWidget(8, QtGui.QFormLayout.LabelRole, self.label_24)
        self.testVoltage = QtGui.QLineEdit(Dialog)
        self.testVoltage.setObjectName("testVoltage")
        self.formLayout_3.setWidget(8, QtGui.QFormLayout.FieldRole, self.testVoltage)
        self.label_21 = QtGui.QLabel(Dialog)
        self.label_21.setObjectName("label_21")
        self.formLayout_3.setWidget(9, QtGui.QFormLayout.LabelRole, self.label_21)
        self.testRunType = QtGui.QComboBox(Dialog)
        self.testRunType.addItem("")
        self.testRunType.addItem("")
        self.testRunType.setObjectName("testRunType")
        self.formLayout_3.setWidget(9, QtGui.QFormLayout.FieldRole, self.testRunType)
        self.line_5 = QtGui.QFrame(Dialog)
        self.line_5.setFrameShape(QtGui.QFrame.HLine)
        self.line_5.setFrameShadow(QtGui.QFrame.Sunken)
        self.line_5.setObjectName("line_5")
        self.formLayout_3.setWidget(13, QtGui.QFormLayout.SpanningRole, self.line_5)
        self.label_22 = QtGui.QLabel(Dialog)
        font = QtGui.QFont()
        font.setPointSize(10)
        font.setWeight(75)
        font.setBold(True)
        self.label_22.setFont(font)
        self.label_22.setObjectName("label_22")
        self.formLayout_3.setWidget(14, QtGui.QFormLayout.LabelRole, self.label_22)
        self.buttonInitVisa = QtGui.QPushButton(Dialog)
        self.buttonInitVisa.setObjectName("buttonInitVisa")
        self.formLayout_3.setWidget(15, QtGui.QFormLayout.LabelRole,
self.buttonInitVisa)
        self.buttonRunTest = QtGui.QPushButton(Dialog)
        self.buttonRunTest.setObjectName("buttonRunTest")
```

```
        self.formLayout_3.setWidget(16, QtGui.QFormLayout.LabelRole,
self.buttonRunTest)
        self.buttonSaveData = QtGui.QPushButton(Dialog)
        self.buttonSaveData.setObjectName("buttonSaveData")
        self.formLayout_3.setWidget(17, QtGui.QFormLayout.LabelRole,
self.buttonSaveData)
        self.label = QtGui.QLabel(Dialog)
        self.label.setObjectName("label")
        self.formLayout_3.setWidget(10, QtGui.QFormLayout.LabelRole, self.label)
        self.gridLayout_2 = QtGui.QGridLayout()
        self.gridLayout_2.setObjectName("gridLayout_2")
        self.label_2 = QtGui.QLabel(Dialog)
        self.label_2.setObjectName("label_2")
        self.gridLayout_2.addWidget(self.label_2, 0, 0, 1, 1)
        self.label_3 = QtGui.QLabel(Dialog)
        self.label_3.setObjectName("label_3")
        self.gridLayout_2.addWidget(self.label_3, 0, 1, 1, 1)
        self.label_4 = QtGui.QLabel(Dialog)
        self.label_4.setObjectName("label_4")
        self.gridLayout_2.addWidget(self.label_4, 0, 2, 1, 1)
        self.testDataPfet = QtGui.QLineEdit(Dialog)
        self.testDataPfet.setMaxLength(2)
        self.testDataPfet.setObjectName("testDataPfet")
        self.gridLayout_2.addWidget(self.testDataPfet, 1, 0, 1, 1)
        self.testDataCtrl = QtGui.QLineEdit(Dialog)
        self.testDataCtrl.setMaxLength(1)
        self.testDataCtrl.setObjectName("testDataCtrl")
        self.gridLayout_2.addWidget(self.testDataCtrl, 1, 1, 1, 1)
        self.testDataNfet = QtGui.QLineEdit(Dialog)
        self.testDataNfet.setMaxLength(2)
        self.testDataNfet.setObjectName("testDataNfet")
        self.gridLayout_2.addWidget(self.testDataNfet, 1, 2, 1, 1)
        self.formLayout_3.setLayout(10, QtGui.QFormLayout.FieldRole,
self.gridLayout_2)
        self.testData = QtGui.QCheckBox(Dialog)
        self.testData.setObjectName("testData")
        self.formLayout_3.setWidget(11, QtGui.QFormLayout.FieldRole, self.testData)

        self.retranslateUi(Dialog)
        QtCore.QMetaObject.connectSlotsByName(Dialog)

    def retranslateUi(self, Dialog):
        Dialog.setWindowTitle(QtGui.QApplication.translate("Dialog", "Dialog", None,
QtGui.QApplication.UnicodeUTF8))
        self.label_16.setText(QtGui.QApplication.translate("Dialog", "Site Selection",
None, QtGui.QApplication.UnicodeUTF8))
        self.label_13.setText(QtGui.QApplication.translate("Dialog", "Wafer", None,
QtGui.QApplication.UnicodeUTF8))
        self.waferSelection.setItemText(0, QtGui.QApplication.translate("Dialog",
"W30", None, QtGui.QApplication.UnicodeUTF8))
        self.waferSelection.setItemText(1, QtGui.QApplication.translate("Dialog",
"W31", None, QtGui.QApplication.UnicodeUTF8))
        self.waferSelection.setItemText(2, QtGui.QApplication.translate("Dialog",
"W36", None, QtGui.QApplication.UnicodeUTF8))
        self.label_14.setText(QtGui.QApplication.translate("Dialog", "Column Number",
None, QtGui.QApplication.UnicodeUTF8))
        self.waferSiteColumn.setItemText(0, QtGui.QApplication.translate("Dialog",
"1", None, QtGui.QApplication.UnicodeUTF8))
        self.waferSiteColumn.setItemText(1, QtGui.QApplication.translate("Dialog",
"2", None, QtGui.QApplication.UnicodeUTF8))
        self.waferSiteColumn.setItemText(2, QtGui.QApplication.translate("Dialog",
"3", None, QtGui.QApplication.UnicodeUTF8))
```

```
        self.waferSiteColumn.setItemText(3, QtGui.QApplication.translate("Dialog",
"4", None, QtGui.QApplication.UnicodeUTF8))
        self.label_15.setText(QtGui.QApplication.translate("Dialog", "Row Number",
None, QtGui.QApplication.UnicodeUTF8))
        self.waferSiteRow.setItemText(0, QtGui.QApplication.translate("Dialog", "1",
None, QtGui.QApplication.UnicodeUTF8))
        self.waferSiteRow.setItemText(1, QtGui.QApplication.translate("Dialog", "2",
None, QtGui.QApplication.UnicodeUTF8))
        self.waferSiteRow.setItemText(2, QtGui.QApplication.translate("Dialog", "3",
None, QtGui.QApplication.UnicodeUTF8))
        self.waferSiteRow.setItemText(3, QtGui.QApplication.translate("Dialog", "4",
None, QtGui.QApplication.UnicodeUTF8))
        self.waferSiteRow.setItemText(4, QtGui.QApplication.translate("Dialog", "5",
None, QtGui.QApplication.UnicodeUTF8))
        self.waferSiteRow.setItemText(5, QtGui.QApplication.translate("Dialog", "6",
None, QtGui.QApplication.UnicodeUTF8))
        self.waferSiteRow.setItemText(6, QtGui.QApplication.translate("Dialog", "7",
None, QtGui.QApplication.UnicodeUTF8))
        self.testRunType.setItemText(0, QtGui.QApplication.translate("Dialog", "Gate -
Drain", None, QtGui.QApplication.UnicodeUTF8))
        self.testRunType.setItemText(1, QtGui.QApplication.translate("Dialog", "Gate -
Source", None, QtGui.QApplication.UnicodeUTF8))
        self.label_17.setText(QtGui.QApplication.translate("Dialog", "Filename
Prefix", None, QtGui.QApplication.UnicodeUTF8))
        self.label_18.setText(QtGui.QApplication.translate("Dialog", "Test
Parameters", None, QtGui.QApplication.UnicodeUTF8))
        self.label_19.setText(QtGui.QApplication.translate("Dialog", "Clock Frequency
(kHz)", None, QtGui.QApplication.UnicodeUTF8))
        self.label_20.setText(QtGui.QApplication.translate("Dialog", "Test Time (µs)",
None, QtGui.QApplication.UnicodeUTF8))
        self.label_24.setText(QtGui.QApplication.translate("Dialog", "Supply Voltage",
None, QtGui.QApplication.UnicodeUTF8))
        self.label_21.setText(QtGui.QApplication.translate("Dialog", "Preset Test
Type", None, QtGui.QApplication.UnicodeUTF8))
        self.label_22.setText(QtGui.QApplication.translate("Dialog", "VISA Commands",
None, QtGui.QApplication.UnicodeUTF8))
        self.buttonInitVisa.setText(QtGui.QApplication.translate("Dialog", "Initialize
VISA Devices", None, QtGui.QApplication.UnicodeUTF8))
        self.buttonRunTest.setText(QtGui.QApplication.translate("Dialog", "Run Test",
None, QtGui.QApplication.UnicodeUTF8))
        self.buttonSaveData.setText(QtGui.QApplication.translate("Dialog", "Save
Results", None, QtGui.QApplication.UnicodeUTF8))
        self.label.setText(QtGui.QApplication.translate("Dialog", "Sent Data", None,
QtGui.QApplication.UnicodeUTF8))
        self.label_2.setText(QtGui.QApplication.translate("Dialog", "PFET Gates",
None, QtGui.QApplication.UnicodeUTF8))
        self.label_3.setText(QtGui.QApplication.translate("Dialog", "Output Drive",
None, QtGui.QApplication.UnicodeUTF8))
        self.label_4.setText(QtGui.QApplication.translate("Dialog", "NFET Gates",
None, QtGui.QApplication.UnicodeUTF8))
        self.testData.setText(QtGui.QApplication.translate("Dialog", "Don\'t Activate
Test Enable", None, QtGui.QApplication.UnicodeUTF8))
```