

ECE 493 Final Project

Spring 2017

College of Engineering and Computing

Department of Electrical and Computer Engineering

ECE 493-593: Power Electronics

Mark Scott, PhD

Tianqi Wang

Mary Floren

Due-May 5,2017

# Introduction

Power Electronics focuses on the various ways to control the electrical energy through electronics. A printed circuit board (PCB) is a tool commonly used to create power electronics devices such as converters and inverters. For our final project, we designed, created, and tested a PCB for this purpose. Our PCB was tested in a laboratory as a buck converter and with a double pulse test controlled by programming a DSP. Calculations were performed that show characteristics of the board such as the efficiency of the PCB and the stray inductance present. Next, the board was modeled on PSIM and experimental results were compared to simulated and theoretical values. We also reviewed two different papers focusing on maximizing the output power of photovoltaic cells. Altogether, we gained a firm understanding of laboratory exercises, working with PCB’s, coding a digital signal processor (DSP), and modeling physical experiments in PSIM.

# Literature Review

The topic of power output is an important consideration in the world of solar power. Photovoltaic (PV) cells often do not produce the power they are capable of producing and various engineers are considering ways to maximize power output of solar arrays. There are numerous causes to the lack of full power output and some have to do with inconsistent conditions such as weather patterns and shading or alterations in the composition of the PV itself over time. Since the world is currently in need of more clean energy and less waste, finding how to achieve the maximum power output of these photovoltaic arrays is important. With inefficient systems, valuable and free renewable energy is being thrown away. Two papers, *Comparison of Photovoltaic Array Maximum Power Point Tracking Techniques* by Trishan Esram and Patrick L. Chapman and *Maximizing the Power Output of Partially Shaded Photovoltaic Plants Through Optimization of the Interconnections Among its Modules* by Luiz Fernando Lavado Villa, Damien Picault, Bertrand Raison, Seddik Bacha, and Antoine Labonne discuss different approaches to maximizing the power output of photovoltaic cells. With improvements in the power output of our solar grids we can move away from our dependance on Earth’s diminishing resources and work towards a healthier relationship with our planet.

*Comparison of Photovoltaic Array Maximum Power Point Tracking Techniques* focuses on finding the highest point, maximum power point (MPP) on a photovoltaic (PV) array’s characteristic power curve and altering conditions to move towards or stay at the highest point on the curve. The various methods discussed in the work involve sampling the current power output, finding where the system lies on the power curve and moving up and down this curve to achieve or maintain maximum power. The movement along this curve is often achieved by altering the voltage or current. When the PV has not yet reached the MPP, the slope of the change in power over time will be positive, indicating that it is still rising. At the point of MPP this slope will be zero because the curve is at a peak. After the peak, the slope will be negative.

Observing the slope behavior can give the system insight into whether to increase, maintain, or decrease the current reference voltage of which the system is basing its operations. One way to improve this process is to make the step sizes between each output power measurement dynamic. This way, as the system approaches the MPP the step sizes shrink, allowing the system to decrease fluctuation around the MPP. Another tactic, Fuzzy Logic Control, characterizes the current condition of the PV array according to a predetermined table of possible states and makes alterations based on this. Though this provides a quick way for the system to make changes, it requires precise knowledge of the system’s behavior prior to use. There are several other methods such as neural networks with internal algorithms working similarly to an API, analyzing the correlations in the ripple created when the power is switched on and off, and a sweep of the current waveform.

Each approach has multiple considerations, including the number of sensors used, how precise the approach is, and costs included. With each method it is important to consider the possibility of local maxima found on the curve that are mistaken for the MPP. This can occur when partial shading is found on the panels and can be a large detriment in the pursuit of maximum power output of the photovoltaic array.

*Maximizing the Power Output of Partially Shaded Photovoltaic Plants Through Optimization of the Interconnections Among its Modules* dives into the issue of power loss in the event of partial shading and ways to counter the losses. This paper narrows in on various topologies, or layouts, that can be used in a photovoltaic plant. Four topologies were tested and analyzed: series-parallel (SP), bridge-link (BL), totally cross tied (TCT), and honeycomb (HC). Each was observed when the four main types of shadows were acting on them. The four types of shadows used were short & wide, long & wide, short & narrow and long & narrow.

It was found that the system achieved better results when the least amount of interconnected cells were affected. Therefore the topologies that allowed for rerouting of current paths proves most effective. This way, the most modules could be used together when others are in different conditions. This means that, if possible, the modules in the sun should be linked and work at their MPP while those in the shade should be linked and work at their MPP.

The papers take different research focuses and solutions to the issue of power loss in photovoltaic arrays. One discusses the ways to find and maintain the maximum power output through heavily software-based approaches and the other proposes better topologies to fix the issue of power loss in partial shading, which is narrowing in on the hardware side of the project. Since the most important part of almost any business is to make money, the cheaper route would likely be maximizing the use of the software approaches to maintaining the MPP as opposed to physically altering the layout of the PV array. However, if a PV array is not built yet, the topology is a worthwhile consideration.

Another reason to focus on finding the MPP using approaches in the first paper is versatility. On days with varying weather, step sizes or algorithms can be changed accordingly and any system can benefit from a more efficient way to stay at maximum power output. In changing the topology, the PV array may be limiting itself when conditions do not include partial shading. For example, if the photovoltaic cell array is in a place with a wide range of seasons, such as the midwest, a topology that favors partial shading may be advantageous for 6 months of the year, however it could cause the array to miss out on valuable power for the other, sunnier 6 months of the year.

Both papers provide valuable insight into improving the power output of PV arrays, however *Comparison of Photovoltaic Array Maximum Power Point Tracking Techniques* provides a better option with a higher potential to be useful. It is important to note that each photovoltaic cell’s position is unique and the variations in conditions mean different solutions are best for different cells. Therefore, versatility is important so the system can vary when it needs. As we move forward we should look into more simple, clever solutions such as the programming algorithms that efficiently find the MPP.

# Theory of Operations

Several safety precautions played an important role in this project to ensure the safety of our board and ourselves. Lab glasses were worn while we were in the lab to protect our eyes in the event of a spark or explosion. When connecting the power source to the circuit we connected the ground wire first to ensure that was on securely and we had the power off at any time we were modifying or not using the circuit. Any time the power source, function generator, or oscilloscope was attached we double checked the links to make sure everything was on correctly and making full contact where they should. When using the function generator we made the connections and input the parameters into the function generator before setting it into output mode. Before inputting 24 V we checked out circuit connections to make sure there were no sneak paths from the oscilloscope probes and everything was properly grounded.

When preparing our circuit for the DSP testing we performed calculations of load resistance and rise time so we could control the output current of our circuit and make sure it was not too high for it to handle.

After the PCB was built it was incrementally tested so we could check individual parts. First, a voltage source was attached with an input of 5V to make sure that current ran through the board correctly before any PWM was attached. Next, a function generator was added to check the functionality. After those were confirmed we went forward with the experimental tests.

During the experiments we payed close attention to the hardware and if it smelled as if it was burning or made a loud noise we turned off the power so we could double check the wiring and consider what was happening.

U1 and U2 are isolated DC to DC converters, PDS1-S5-S12-M, CUI Inc. They boost a DC voltage of 5V with an input LC filter up to a DC voltage of 12 V with output common mode filter.

The digital pulses of PWM were coded in the state machines in DSP TMS320F28335 and these two channels went through U5 and U8 which are optical couplers, specifically SI8710 of Silicon LABS. There is galvanic isolation between U5 and U8 which provides separation between the PWM circuit and the gate drive circuit to prevent current flow between them. The gate drive IC is UCC27519. U6 and U9 are gate drivers that control when the MOSFET is turned on and off.

The Infineon IPx60R600C6s N-channel MOSFET is the power IC. This contains a low loss of 600 V/6A. In the lab, two 2 parallel resistors called the shunt resistance were used to test the current of MOSFET with an oscilloscope probe.

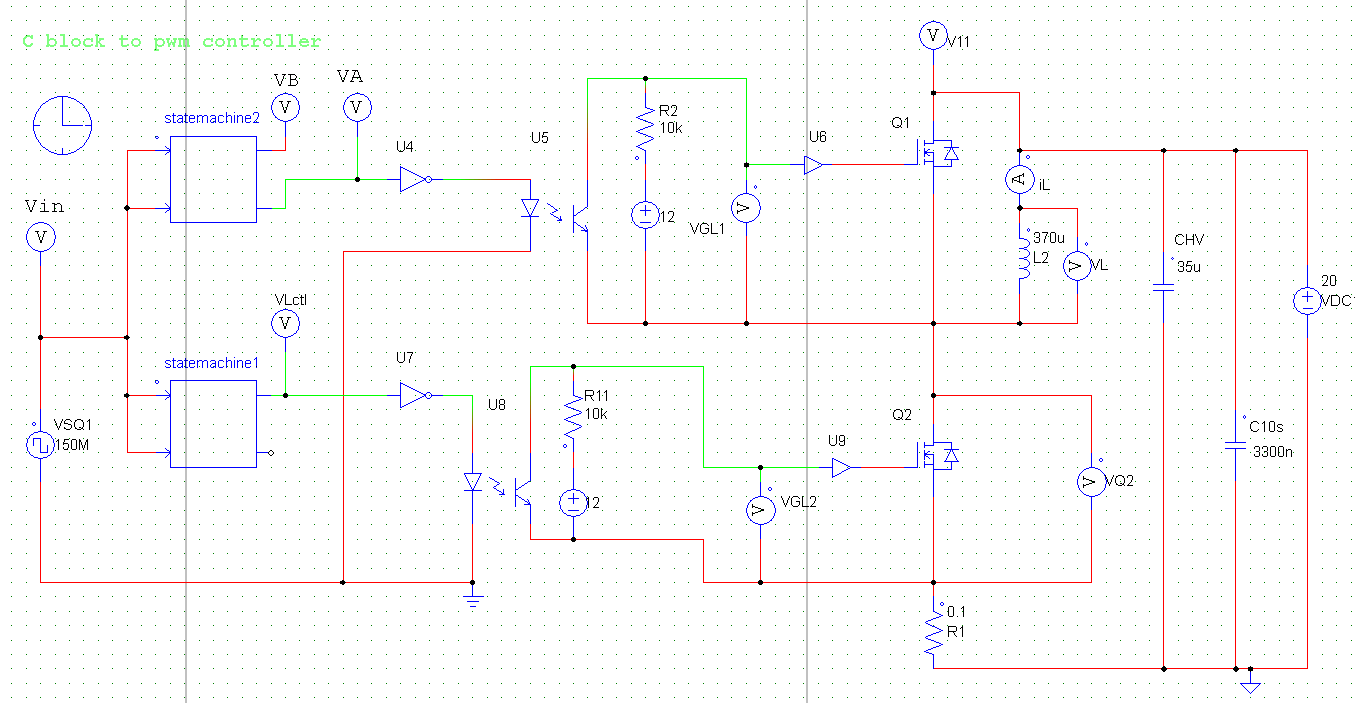


Figure 1: Schematic of the Double Pulse Test circuit.

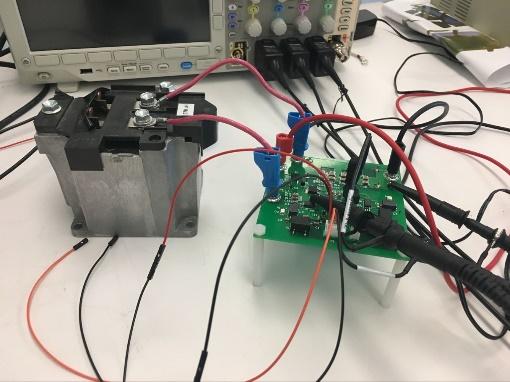


Figure 2: Laboratory setup of the Double Pulse Test.

At various points on the board, filters are created by inductors and capacitors to combat noise. A differential mode filter is formed at U1 with C1, L1 and C2. At U2 a differential mode filter is formed by C5, L3, and C6. A common mode filter at U1 is formed by L2 and at U2 a common mode filter is formed by L4.

A double pulse test procedure is used to test the switching behaviors and energy losses of the MOSFETS or IGBTs in a circuit. It shows the timing delays when the switch is turning on and off and from there it is possible to analyze the power lost in the switch or switches.

Once the inductor current is risen to the intended value, the gate momentarily switches off and back on before signaling low to lower the current back to the original value. This gives insight into the circuit’s switching capabilities because it shows the behavior when a short fall time, , occurs in relation to the proceeding on time, .

For the double pulse test, the theoretical energy losses in turn-on were slightly lower than the experimental energy losses and in turn-off the theoretical energy losses were slightly lower than in practice. The calculations for these values are seen in *Theoretical Calculations*.

The calculations showed that the energy loss in turn on was less than that in turn off. However, in DPT the loss in the switching was inverse. These observations may be due to the unideal behaviour of the parts, inexact inputs or measurements, or other various experimental alterations.

As seen in calculations below, the experimental efficiency of the buck converter was 95.4% and the experimental efficiency was 83.8%. In calculations the efficiency is assumed to be 100%, meaning the input power is equivalent to the output power. In PSIM, the simulated stray inductance hinders the efficiency of the design while in practice numerous effects impact the efficiency, such as stray inductance, temperature, operator error, imprecise equipment, and many other possible factors.

# Theoretical Calculations

There are four timing intervals involved in controlling the gate. The first time section, , refers to a high signal that allows the inductor current to climb to a specified value. is a short low signal followed by , a short high signal. The last signal, , is a long low signal allowing the current to fall back down. In order to have our circuit work with an intended current level, we had to calculate and alter , which would determine the amount of time the gate is high and allowing current to pass. Our inductor voltage was 20V the desired inductor current was 2A. The inductor used in this step was 370. Equation 1 was rearranged to produce Equation 2 which solves for the change in time necessary to produce the current change from 0A to 2A.

(1)

(2)

Inserting our values into the equation gave= 39.9.

We used and = 5because is a small interval and and should be equal to each other. The period is one second. Therefore all four stages of the period, , , and should add up to one second total and was calculated by subtracting , and from one second.

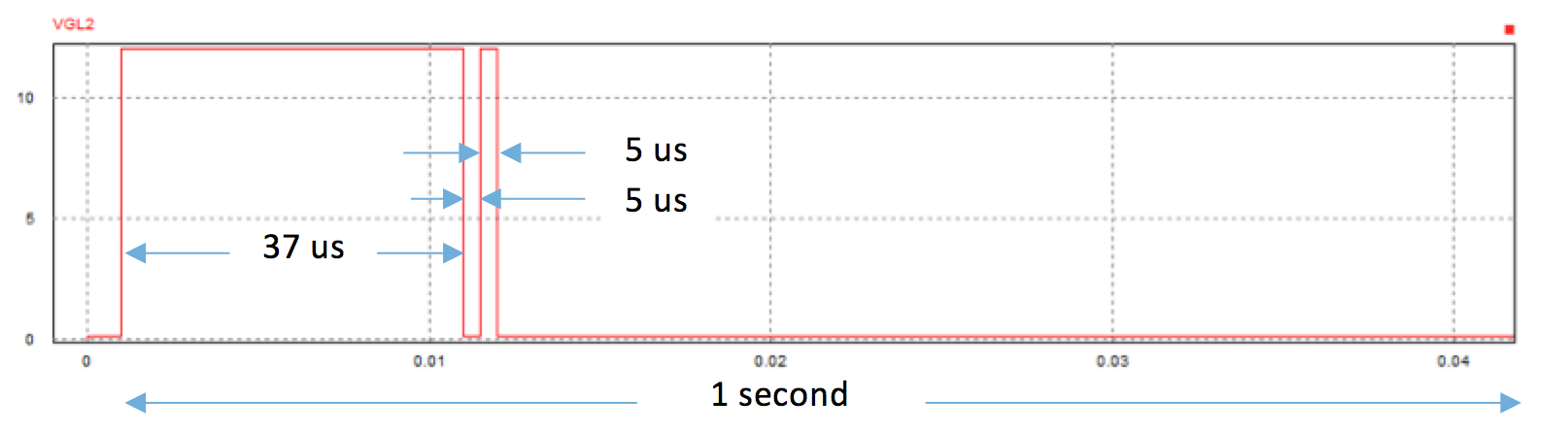


Figure 3: The DPT waveform to be designed.

These time segments were input into the DSP code. To do this, the counting was set to count-up-down mode. The DSP clock is 150 MHz and 1was used as the counting unit of the state machine. Because of this, was set to 8, was set to 1, was also set to 1 and was set to 199990 with TBPRD = 375. After this was connected to the circuit, an oscilloscope was used to observe the current waveform. The results are seen in *Figure 4 and Figure 5.*

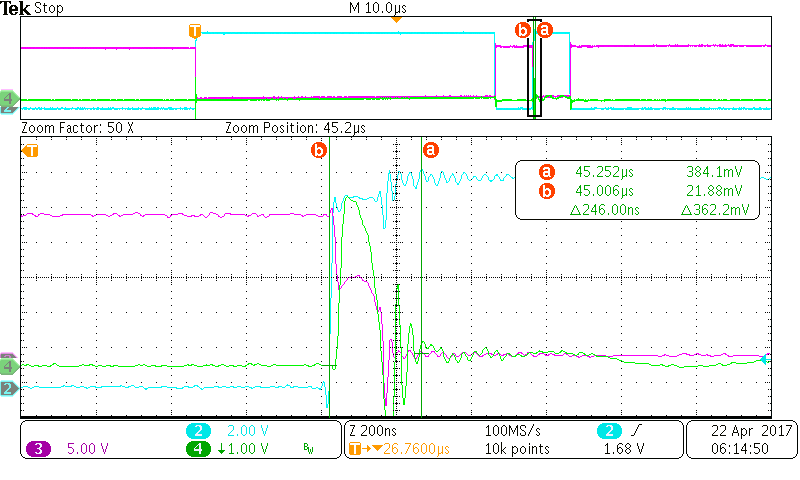


Figure 4: *Waveforms of the DPT*: Switching On Behavior.

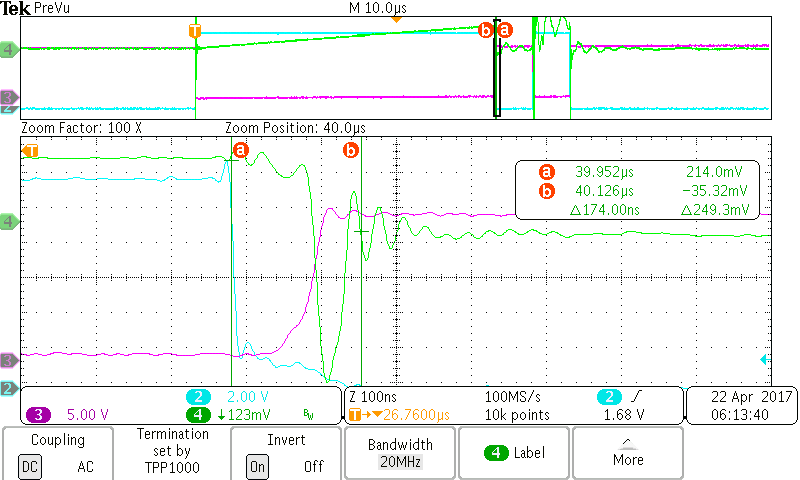


Figure 5: *Waveforms of the DPT*: Switching Off Behavior.

The waveform in *Figure 5* shows that the experimental values of and are = 39.952 and iL, MAX = . Using *Equation 3* the inductance was found to be,

(3)

With an exported CSV file from the oscilloscope, the losses in turn-on and turn-off were calculated by using the MATLAB code in *Appendix B*.

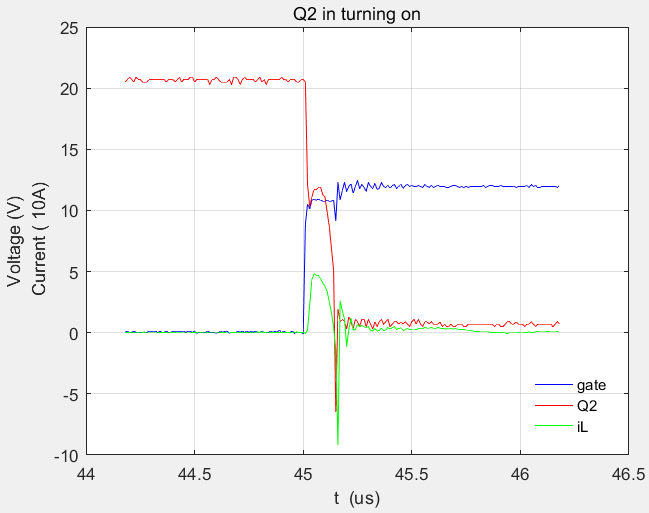


Figure 6: Plotted voltages and currents in on state with MATLAB.

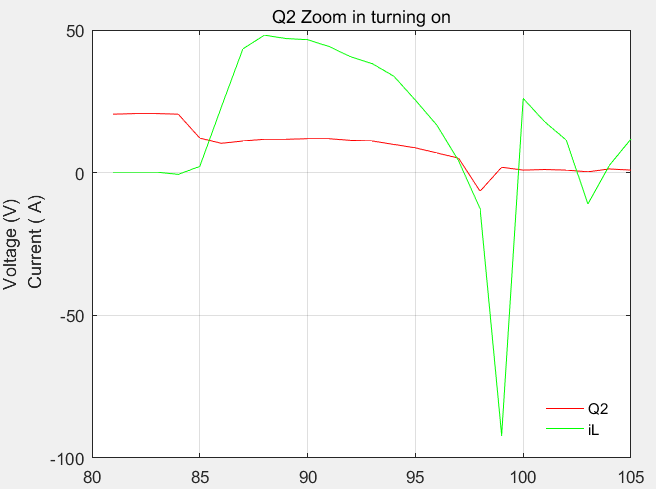


Figure 7: Zoomed in plot of voltages and currents in on state with MATLAB.

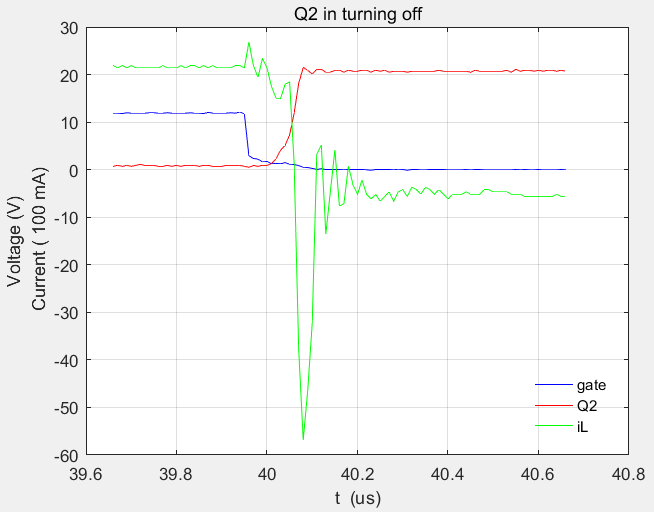


Figure 8: Plotted voltages and currents in off state with MATLAB.



Figure 9: Zoomed in plot of voltages and currents in off state with MATLAB.

The results were that the energy loss during turn-on of the power MOSFETS was = 66.44 while the energy loss during turn-off of the power MOSFETS was =1.23 .

The MOSFETs and are both the part number IPD60R380C6. According to the datasheet, , , , and . In our lab we used and .

(4)

(5)

Equation 4 and Equation 5 were used to find the theoretical energy loss during the turn-on and turn-off stages of the MOSFETS. = = J, = J.

When operating as a buck converter with , , D=0.5, and , the junction temperature of the power devices can be calculated by looking at the power losses. The thermal characteristics of IPD60R380C6 packaged in TO-252 are RDS\_ON = 340 mΩ, ton = 25 ns, and toff = 119 ns. The thermal characteristics of the built-in reverse diode are VF = 0.64 V at 25 °C and Qrr = 3.3 uC.

It was observed during the buck converter experiment that and . The following calculations were performed for the power loss of the MOSFET.

(6)

(7)

(8)

The power loss in diode was calculated using the following equations.

(9)

(10)

The MOSFET took part in the conducting in the situation of the Q1 turned off, the conduction loss was main the conduction loss of Q2.

(11)

(12)

(13)

(14)

The thermal power distribution was,

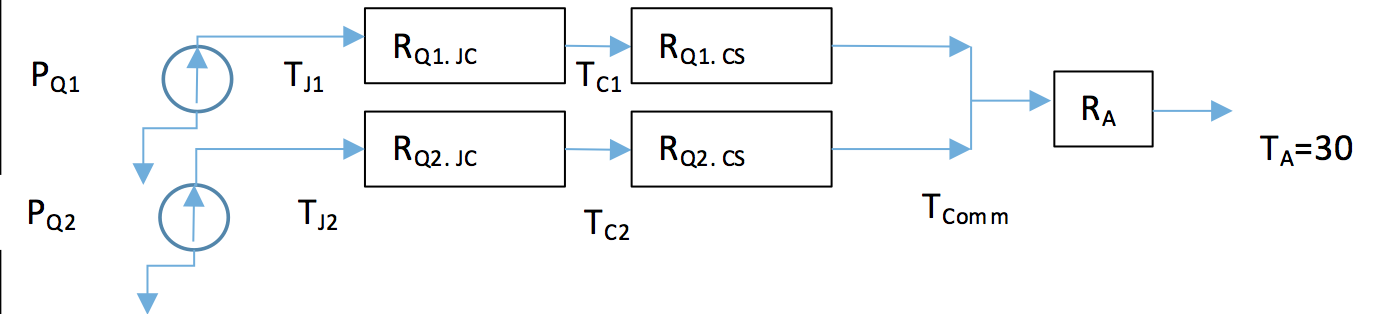


Figure 10: Outline of thermal power distribution

The ambient temperature (TA) is 30 °C and the junction temperature needs to be less than, so supposing = 1 °C/W,

(15)

(16)

If the junction temperature of power MOSFETs stayed below 65 °C when the ambient temperature is 30 °C and MOSFET in a single heatsink, the thermal impedance of the heatsink was . The benefits of two MOSFETs used in the buck converter are the decrease of the power loss.

The first task in the buck converter portion of the lab was to design the inductor. The VIN was 20 V, switching frequency was 50 kHz with duty cycle 0.5, IOUT was 2 A, and desired inductance, L, was 150 uH. The experimental core was Toroids R 50.0 × 30.0 × 20.0, Series/Type: B64290L0082. The parameters are in Table 1.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Material | AL value | μi | Ordering code | Magnetic characteristics | | | |
|  | nH |  |  | Σl/A mm –1 | Le mm | Ae mm2 | Ve mm3 |
| N87 | 4460 | 2200 | B64290L0082X087 | 0.62 | 120.4 | 195.7 | 23560 |

Table 1: Characteristics and ordering codes

Le: Effective magnetic path length

Ae: Effective magnetic cross section

We calculated the number of turns to create the 150 uH inductor using u0=4pi\*10-7; ui = 2200; Bs = 490 mT, at 25 oC.

(17)

The saturation current was then calculated using the following equations.

(18)

(19)

, the resistance of the load was 5 . This was calculated by considering that VIN = 20 V and IOUT = 2 A. To find the output voltage in a buck converter, *Equation 20* and *Equation 21* were used.

V = IR (20)

(21)

Using these, we get and gives =5.

The J3 and J2 ports of the PCB were connected to the inductor in series with , which was in parallel with the capacitor 100 F.

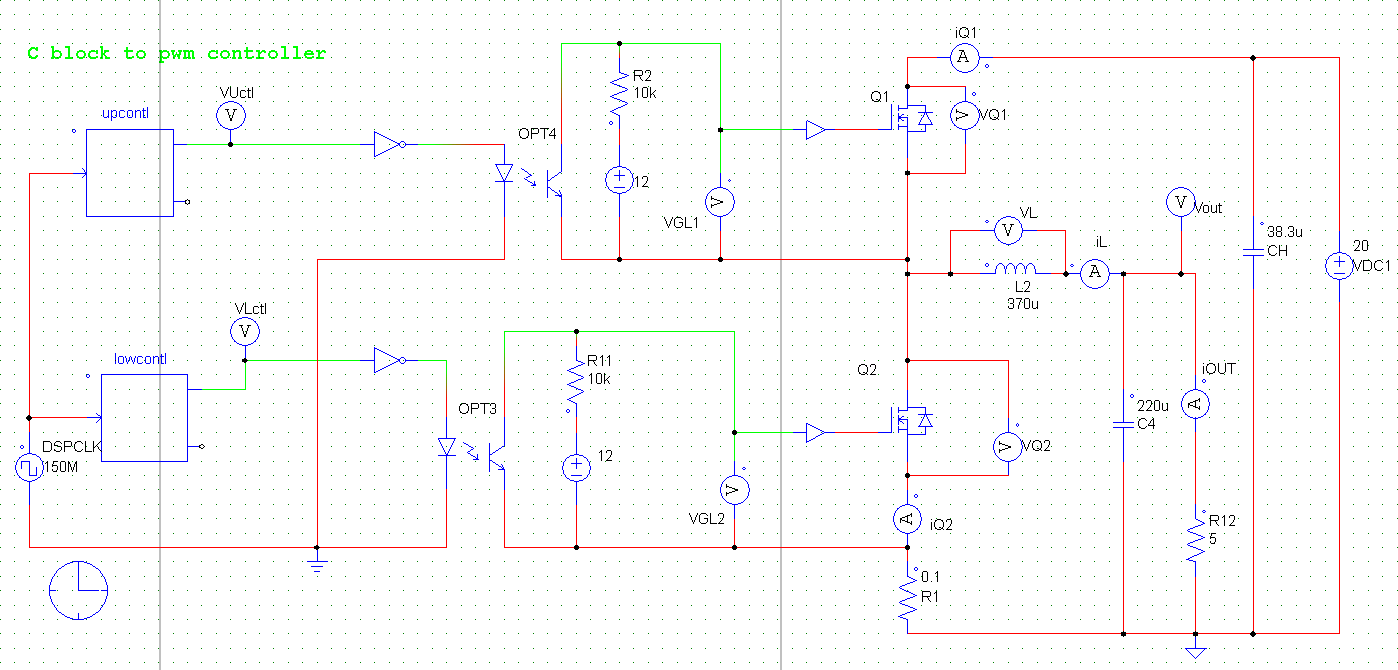


Figure 11: Schematic of transforming the PCB to a buck converter circuit.

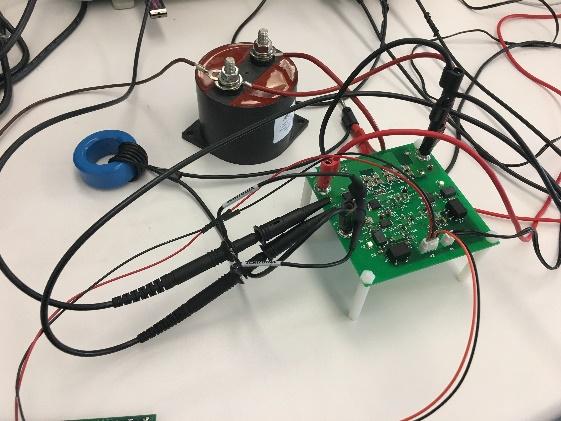


Figure 12: Setup of buck circuit in the lab.

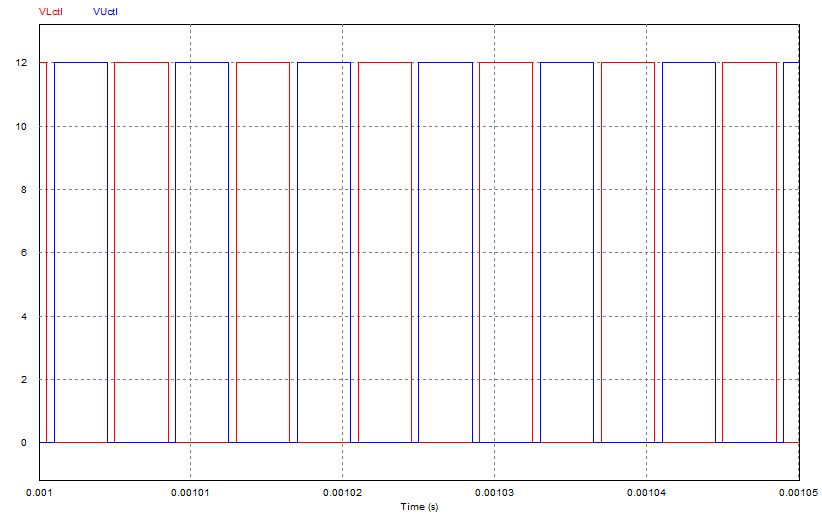


Figure 13: PWM waveforms to control MOSFETs of buck converter.



Figure 14: Waveforms displayed on the oscilloscope.

The power input and output curves were recorded in *Figure 15*. The pink wave is the input current, the green wave is the output current and the light green wave is the voltage stress on the MOSFET. The averages of the input and output currents were IOUT = 1.97 A and IIN = 0.968 A, as displayed on the oscilloscope graphs. VOUT was 8.61 V and VIN was 19.85 V as measured by the multimeter. The ripple of the output current was .

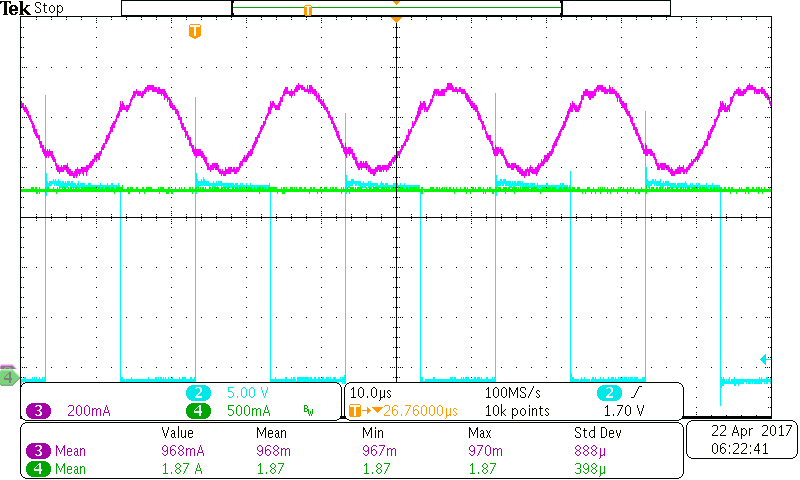


Figure 15: IOUT, IIN and QDS waveforms of buck converter, L= 154 uH.

The power efficiency was,

(22)

The peak to peak current in the inductor was 2.38 A in the experiment and its period was 2\*10.05= 20.10 corresponding to 50 kHz PWM of the DSP output in *Figure 16*. The current wave showed the buck converter working in CCM.

Theoretically, since the mean current of the inductor was 1.87 A and this is over half of the peak to peak current in the inductor, 1.19 A, the converter was operating in CCM.

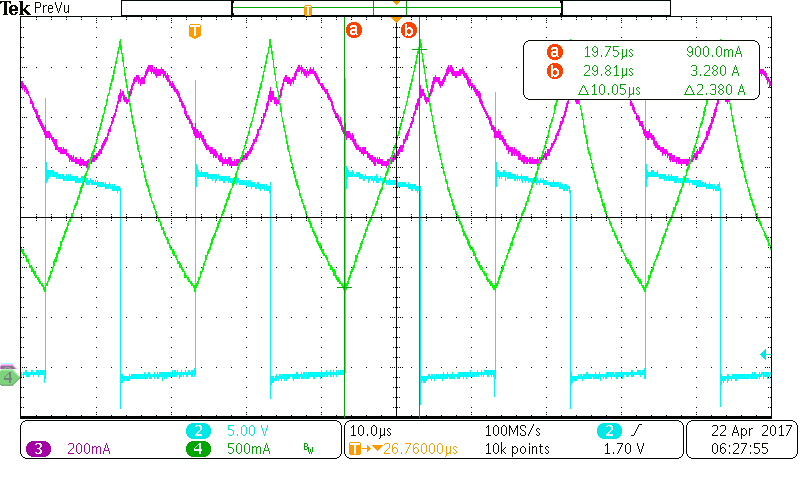


Figure 16: Buck converter waveforms.

Green: inductor current, pink: output current, light green: voltage stress.

The theoretical calculation of the peak to peak current in inductor was,

(23)

(24)

Using 2.38 A, the efficiency of the inductor was calculated.

(25)

According to *Equation 18*,

(26)

With the max peak current of inductor,

(27)

Since the calculated result of, the inductor was not saturated during the test. Therefore the real value of the inductance in the buck converter was operated with going through current.

# Simulation Results

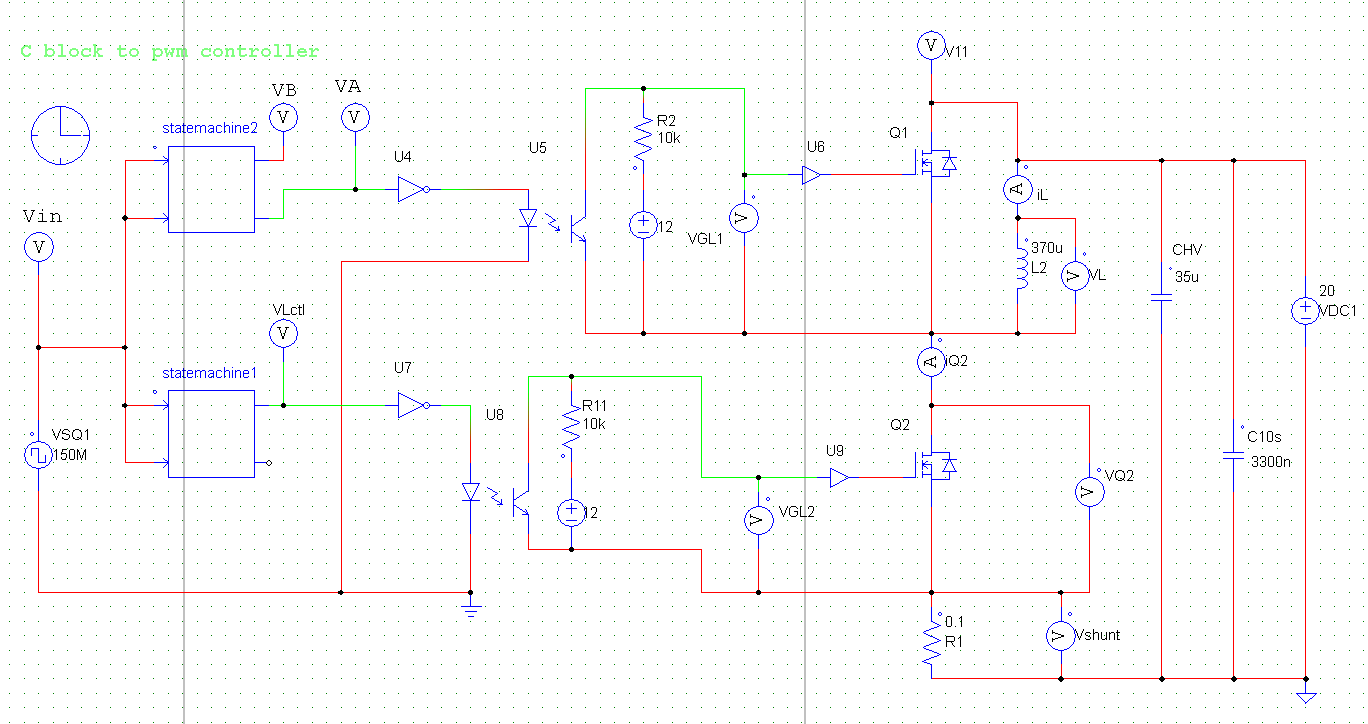


Figure 17: Schematic of PSIM simulation with no stray inductance.

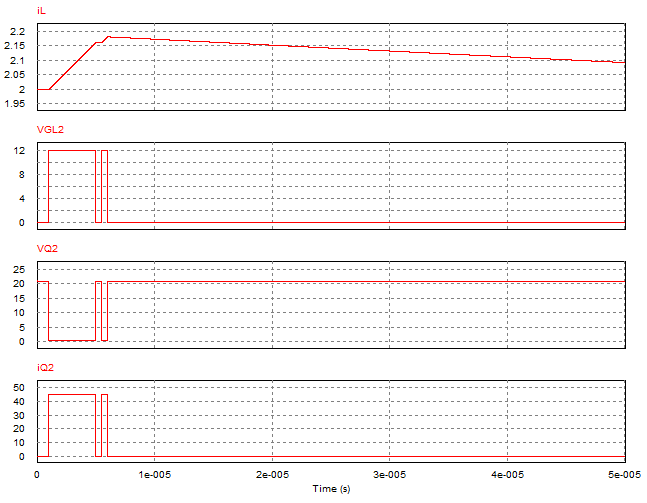


Figure 18: PSIM simulation output waveforms.

The inductance of a flat or ribbon wire (rectangular cross section), can be found using the calculator or the formula given below.

The stray inductances in the PCB were compounded of three parts: connecting wire, routing traces of the HVDC- and Lcom and loops of Q2 with HVDC. The stray inductance caused by shunting resistance in the PCB was induced by a trace of width 3and length 30, which created inductance of 21.04, and another trace of width 10 and length 65, which created inductance of 40.25. The loop inductance was similar to a conduction wire with loops 100 long and a diameter of 1, which created an inductance of 102.19. This was calculated by the PCB inductance calculator 8. Therefore, the stray inductance was reasonable at the total value of 200for the PCB.

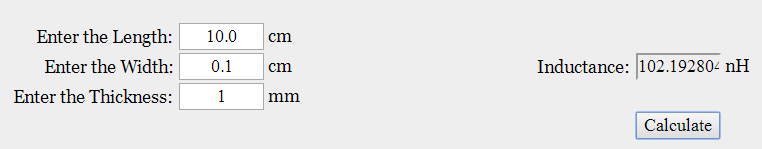
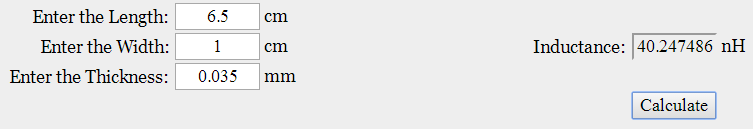
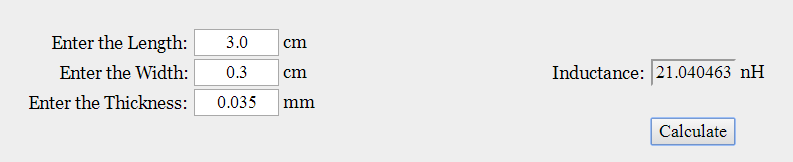


Figure 19: The stray inductance of PCB calculated by the PCB calculator.

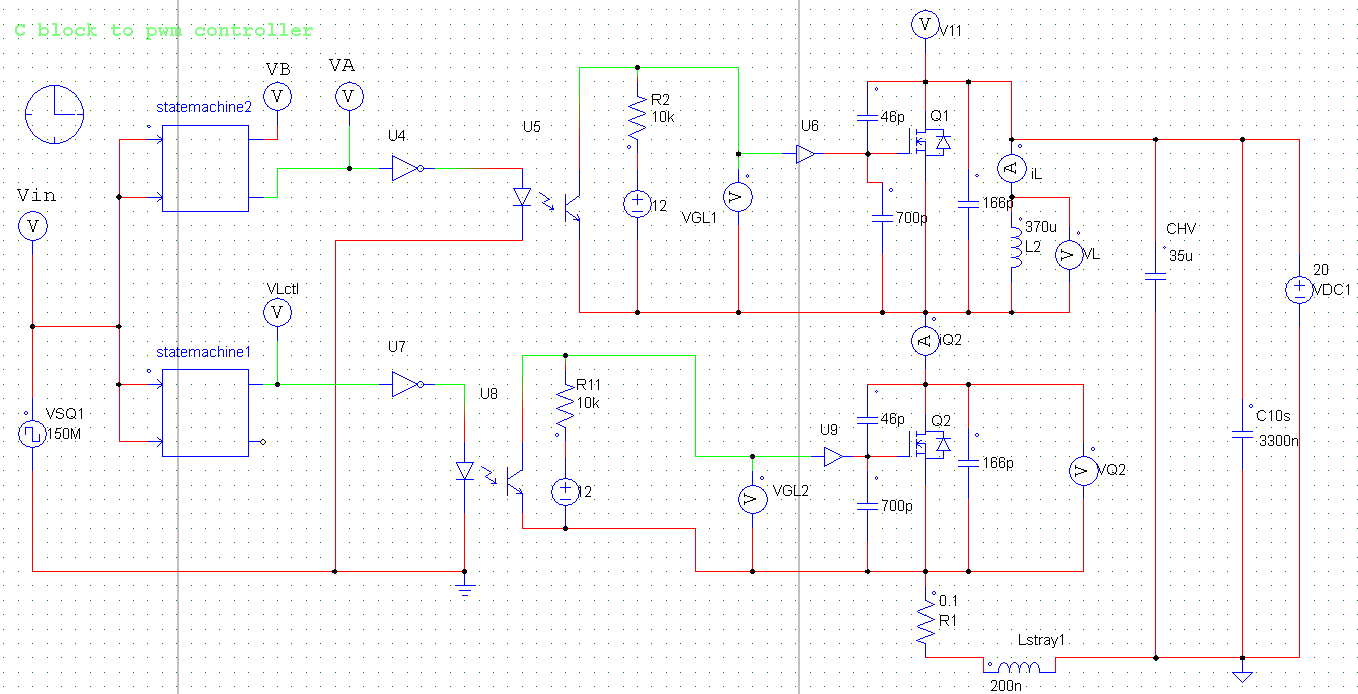


Figure 20: PSIM simulation with stray inductor of 200 nH

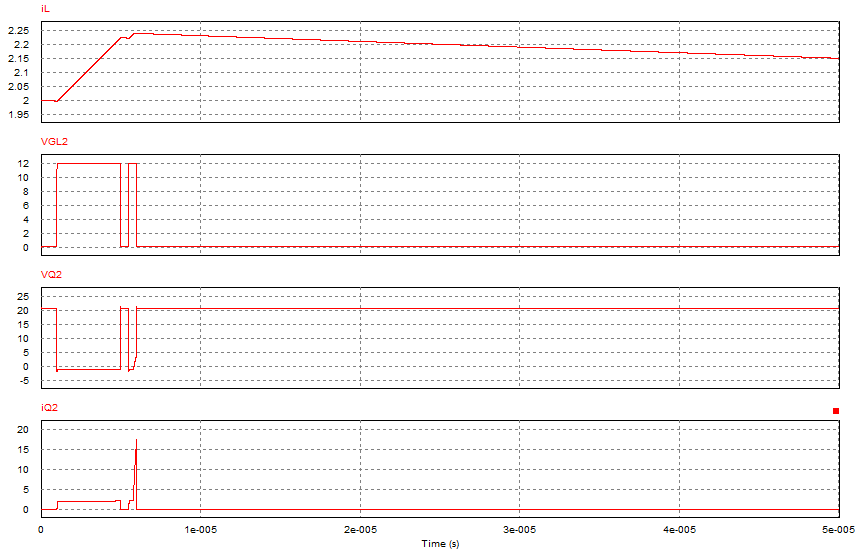


Figure 21: PSIM output waveforms.

When there was stray inductance in the PCB, the waveform of the current through Q2 was limited to overcome the negative voltage created by the stray inductance in the conducting state of Q2. In the second test pulse the stray inductor was not discharged totally, so there was a sharp increasing in the conducting current in *Figure 21*. there was the same response in VQ2 waveform.

# 

The initial conditions of the buck converter were: VIN = 20 V, IOUT = 2 A and L = 154 uH and frequency 50 kHz, shown in *Figure 22*.

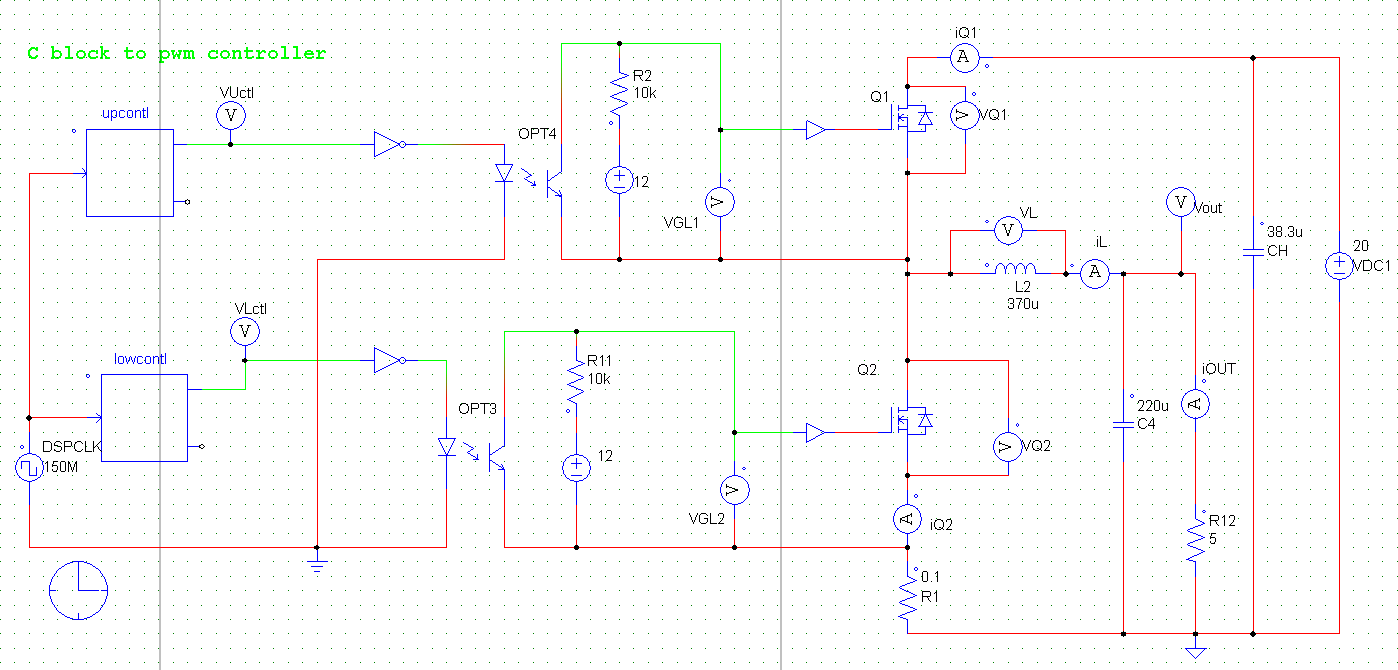


Figure 22: PSIM schematic of simulated buck converter on the PCB.

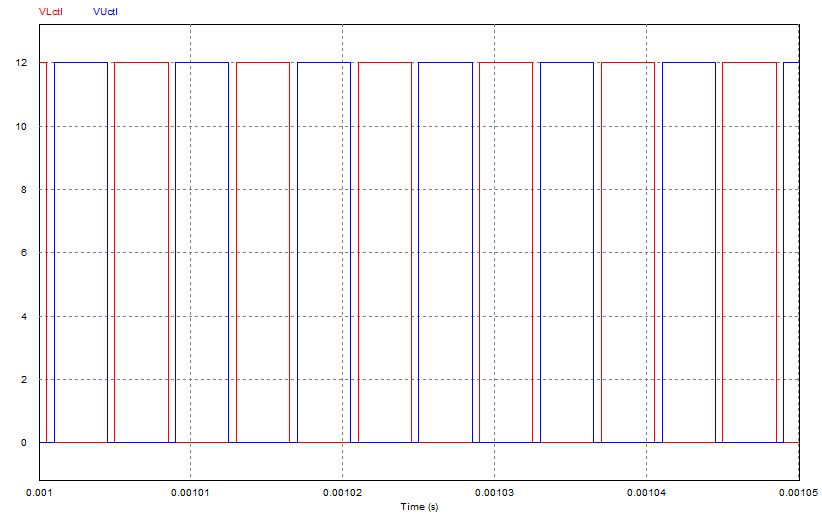


Figure 23: PSIM waveforms.

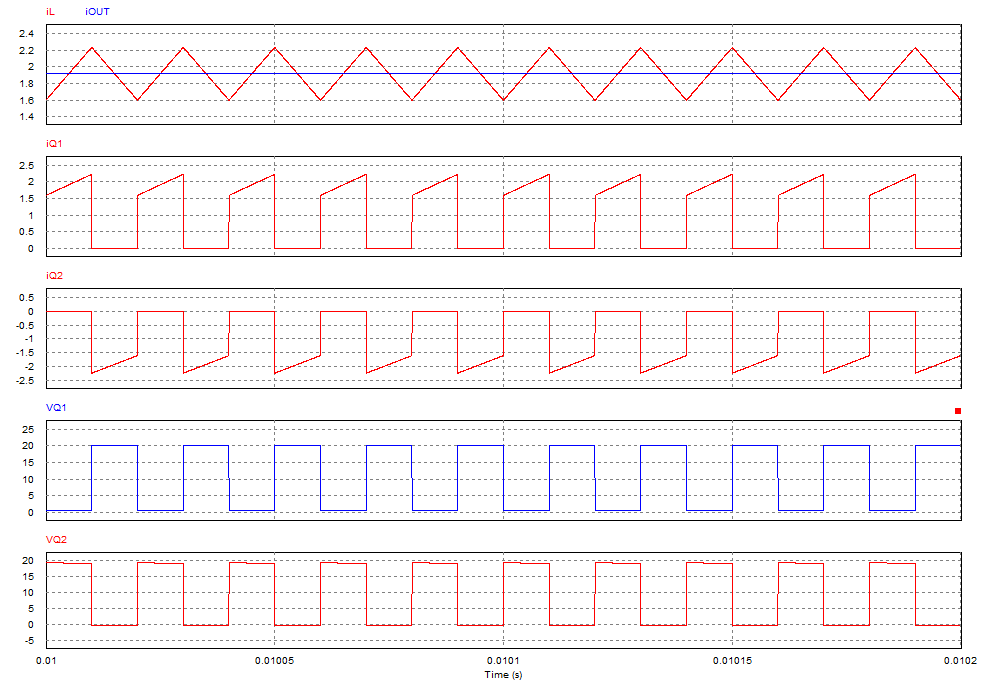


Figure 24: Output waveforms of PSIM simulation with stray inductor 200 nH.

There were the simulation results of the buck converter of PCB with load resistance of 5. The output current was 1.91 A. The power efficiency was,

(28)

# Conclusion

We produced a PCB that could act as a buck converter. The various experiments on this printed circuit board indicate that experimental results as well as simulated results vary from theoretical results. This is due to attributed unaccounted for in calculations, such as stray inductance. The double pulse test allowed us to better understand the switching behaviors of the MOSFET and developed our understanding of the characteristics of current going through components in a circuit. If we were to repeat this project a main objective would be to try to reduce stray inductance in the board and see if we could produce a PCB that could handle higher voltage levels. Overall, our PCB was a successful first step into the world of Power Electronics.

Reflection

This project was useful in many ways. It included topics from topics we learned all throughout the semester both in class and in labs and that helped synthesize everything. It was difficult enough to be a challenge but not too difficult that the learning curve way too high to be attainable. Having something so content-heavy and directly related to lectures helped to learn the material, especially since we witnessed concepts firsthand instead of simply reading .

Stricter deadlines may help improve this project in the future to break it up into smaller, less intimidating sections. Also, requiring that every student complete this project would allow for more class time to be focused on discussing the project instructions and hold ups. However, the help outside of class was very valuable and made the project that much more effective.

**Acknowledgements**

We are grateful for Dr.Scott, Mr. Wilson Guo, Mr. Hassan Hassan and Mr. Egor Iuravin. Thank you for providing discussions and helping us throughout our project.

**References**

[1] Mark Scott, ECE 493 lab guide, 2017.

[2] *TMS320x2833x, 28322x Enhanced Pulse Width Modulator (ePWM) Module*, Texas Instruments, Dallas, TX, Literature Number: SPRUG04A, (Rev. July 2009)

[3] *Texas Instruments C2000 Teaching Materials: Module 7: F2833x PWM, Capture and QEP*, Texas Instruments, Dallas, TX.

[4] TMS320F28335, Digital Signal Controllers (DSCs) Data Manual, Texas Instruments, Dallas, TX, Literature Number: SPRS439M (Rev. Aug. 2012)

[5] *Ferrites and accessories: Toroids R50.0 x 30.0 x 20.0,* EPCOS, Munich, Germany, Sept. 2006

[6] *Ferrites and accessories: SIFERRIT material N87,* EPCOS, Munich, Germany, Sept. 2006.

[7] PCB inductance calculator,<http://chemandy.com/calculators/flat-wire-inductor-calculator.htm>

**Appendix A**

**DSP: controlling PWM of DPT**

// The code written by Dr. Mark Scott.

// Some code modified to ECE DPT project by Tianqi Wang

// 2017-4-10

#include "DSP28x\_Project.h"

typedef struct{

volatile struct EPWM\_REGS \*EPwmRegHandle;

Uint16 EPwm\_CMPA\_Direction;

Uint16 EPwm\_CMPB\_Direction;

Uint16 EPwmTimerIntCount;

Uint16 EPwmMaxCMPA;

Uint16 EPwmMinCMPA;

Uint16 EPwmMaxCMPB;

Uint16 EPwmMinCMPB;

}EPWM\_INFO;void InitEPwm1Example(void);interrupt void epwm1\_isr(void);

EPWM\_INFO epwm1\_info;EPWM\_INFO epwm2\_info;EPWM\_INFO epwm3\_info;

#define EPWM1\_TIMER\_TBPRD 2000

#define EPWM1\_MAX\_CMPA 1950

#define EPWM1\_MIN\_CMPA 50

#define EPWM1\_MAX\_CMPB 1950

#define EPWM1\_MIN\_CMPB 50

#define EPWM2\_TIMER\_TBPRD 2000

#define EPWM2\_MAX\_CMPA 1950

#define EPWM2\_MIN\_CMPA 50

#define EPWM2\_MAX\_CMPB 1950

#define EPWM2\_MIN\_CMPB 50

#define EPWM3\_TIMER\_TBPRD 2000

#define EPWM3\_MAX\_CMPA 950

#define EPWM3\_MIN\_CMPA 50

#define EPWM3\_MAX\_CMPB 1950

#define EPWM3\_MIN\_CMPB 1050

#define EPWM\_CMP\_UP 1

#define EPWM\_CMP\_DOWN 0

Uint16 EPWM1\_PERIOD = 375;Uint16 EPWM1\_CMPA = 375;Uint16 EPWM1\_CMPB = 375;Uint16 ECET1=7; Uint16 ECET2=0; Uint16 ECET3=0; Uint16 ECET4=199880; int StateMachine = 0;int CounterA = 0;

void main(void){

InitSysCtrl();

InitEPwm1Gpio();InitEPwm2Gpio(); InitEPwm3Gpio();

DINT; InitPieCtrl(); IER = 0x0000; IFR = 0x0000;

InitPieVectTable(); EALLOW; PieVectTable.EPWM1\_INT = &epwm1\_isr; EDIS;

EALLOW; SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC = 0;

EDIS; InitEPwm1Example(); EALLOW; SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC = 1;

EDIS; IER |= M\_INT3; PieCtrlRegs.PIEIER3.bit.INTx1 = 1; PieCtrlRegs.PIEIER3.bit.INTx2 = 1;

PieCtrlRegs.PIEIER3.bit.INTx3 = 1; EINT; ERTM;

for(;;){asm("NOP");}

}

interrupt void epwm1\_isr(void){

switch(StateMachine){

case 1:

EPwm1Regs.AQCTLA.bit.CAU = 1; EPwm1Regs.AQCTLA.bit.CAD = AQ\_NO\_ACTION;

if(CounterA++ >= ECET1){StateMachine=2;CounterA = 0; } break;

case 2:

EPwm1Regs.AQCTLA.bit.CAU = 0; EPwm1Regs.AQCTLA.bit.CAD = AQ\_NO\_ACTION;

if(CounterA++ >= ECET2){StateMachine=3; CounterA = 0; } break;

case 3:

EPwm1Regs.AQCTLA.bit.CAU = 1; EPwm1Regs.AQCTLA.bit.CAD = AQ\_NO\_ACTION;

if(CounterA++ >= ECET3){StateMachine=4;CounterA = 0; } break;

case 4:

EPwm1Regs.AQCTLA.bit.CAU = 0;EPwm1Regs.AQCTLA.bit.CAD = AQ\_NO\_ACTION;

if(CounterA++ >= ECET4){ StateMachine=1; CounterA = 0; } break;

default:StateMachine=1;CounterA = 0; }

EPwm1Regs.ETCLR.bit.INT = 1; PieCtrlRegs.PIEACK.all = PIEACK\_GROUP3;

}

void InitEPwm1Example(){

EPwm1Regs.TBPRD = EPWM1\_PERIOD; EPwm1Regs.TBPHS.half.TBPHS = 0x0000;

EPwm1Regs.TBCTR = 0x0000; EPwm1Regs.CMPA.half.CMPA = EPWM1\_CMPA;

EPwm1Regs.CMPB = EPWM1\_CMPB; EPwm1Regs.TBCTL.bit.CTRMODE = TB\_COUNT\_UPDOWN;

EPwm1Regs.TBCTL.bit.PHSEN = TB\_DISABLE; EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB\_DIV1;

EPwm1Regs.TBCTL.bit.CLKDIV = TB\_DIV1; EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC\_SHADOW;

EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC\_SHADOW; EPwm1Regs.CMPCTL.bit.LOADAMODE = CC\_CTR\_ZERO;

EPwm1Regs.CMPCTL.bit.LOADBMODE = CC\_CTR\_ZERO; EPwm1Regs.AQCTLA.bit.CAU = 1;

EPwm1Regs.AQCTLA.bit.CAD = 0; EPwm1Regs.AQCTLB.bit.CBU = 1;

EPwm1Regs.AQCTLB.bit.CBD = 0;

EPwm1Regs.ETSEL.bit.INTSEL = ET\_CTR\_PRD; EPwm1Regs.ETSEL.bit.INTEN = 1;

EPwm1Regs.ETPS.bit.INTPRD = ET\_1ST; epwm1\_info.EPwm\_CMPA\_Direction = EPWM\_CMP\_UP;

epwm1\_info.EPwm\_CMPB\_Direction = EPWM\_CMP\_DOWN; epwm1\_info.EPwmTimerIntCount = 0;

epwm1\_info.EPwmRegHandle = &EPwm1Regs; epwm1\_info.EPwmMaxCMPA = EPWM1\_MAX\_CMPA;

epwm1\_info.EPwmMinCMPA = EPWM1\_MIN\_CMPA; epwm1\_info.EPwmMaxCMPB = EPWM1\_MAX\_CMPB;

epwm1\_info.EPwmMinCMPB = EPWM1\_MIN\_CMPB;

}

**Appendix B**

**MATLAB: Calculation of Losses in Turn On and Turn Off of MOSFET**

clear all;

y = load( 'C:\Users\Administrator\Desktop\tek0002ALL.csv');

fs= 1/0.00000001; % Sample Interval 0.0000000100

f=50000;

tt=1:length(y);

figure (10);

plot(1000000\*y(tt,1),y(tt,2),'b',1000000\*y(tt,1),y(tt,3),'r',1000000\*y(tt,1),100\*y(tt,4),'g'); grid on;

xlabel('t (us)');ylabel({'Voltage (V)';' Current ( 100 mA)'});

legend('gate','Q2','iL','location','southeast');legend('boxoff');

title('Q2 in turning off')

Nleft\_tm = 28; % t-start

Nright\_tm = 48; % t-end

figure(20);

nn = Nleft\_tm:Nright\_tm;

plot(nn,y(nn,3),'r',nn,100\*y(nn,4),'g');

Energy\_off=(10/fs).\*abs(y(nn,2)).\*abs(y(nn,4));

Energy\_off\_total = sum(Energy\_off)

xlabel('');ylabel({'Voltage (V)';' Current ( 100 mA)'});

legend('Q2','iL','location','southeast');legend('boxoff');

title('Q2 Zoom in turning off'); grid on;

**Appendix C**

**DSP: Controlling PWM of Buck Converter**

// The code written by Dr. Mark Scott.

// Some code modified to ECE DPT project by Tianqi Wang

// 2017-4-9

#include "DSP28x\_Project.h" // Device Headerfile and Examples Include File

typedef struct {

volatile struct EPWM\_REGS \*EPwmRegHandle;

Uint16 EPwm\_CMPA\_Direction;Uint16 EPwm\_CMPB\_Direction;Uint16 EPwmTimerIntCount;

Uint16 EPwmMaxCMPA; Uint16 EPwmMinCMPA;Uint16 EPwmMaxCMPB;Uint16 EPwmMinCMPB;}EPWM\_INFO;void InitEPwm1Example(void);interrupt void epwm1\_isr(void);

EPWM\_INFO epwm1\_info;EPWM\_INFO epwm2\_info;EPWM\_INFO epwm3\_info;

#define EPWM1\_TIMER\_TBPRD 2000

#define EPWM1\_MAX\_CMPA 1950

#define EPWM1\_MIN\_CMPA 50

#define EPWM1\_MAX\_CMPB 1950

#define EPWM1\_MIN\_CMPB 50

#define EPWM2\_TIMER\_TBPRD 2000

#define EPWM2\_MAX\_CMPA 1950

#define EPWM2\_MIN\_CMPA 50

#define EPWM2\_MAX\_CMPB 1950

#define EPWM2\_MIN\_CMPB 50

#define EPWM3\_TIMER\_TBPRD 2000

#define EPWM3\_MAX\_CMPA 950

#define EPWM3\_MIN\_CMPA 50

#define EPWM3\_MAX\_CMPB 1950

#define EPWM3\_MIN\_CMPB 1050

#define EPWM\_CMP\_UP 1

#define EPWM\_CMP\_DOWN 0

Uint16 EPWM1\_PERIOD = 1500;Uint16 EPWM1\_CMPA = 825;Uint16 EPWM1\_CMPB = 675;

Uint16 EPWM1\_CLK\_DIV = 0;

void main(void){InitSysCtrl(); InitEPwm1Gpio(); InitEPwm2Gpio(); InitEPwm3Gpio();

DINT; InitPieCtrl(); IER = 0x0000; IFR = 0x0000;

InitPieVectTable(); EALLOW; PieVectTable.EPWM1\_INT = &epwm1\_isr; EDIS; EALLOW;

SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC = 0; EDIS; InitEPwm1Example(); EALLOW;

SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC = 1; EDIS;

IER |= M\_INT3; PieCtrlRegs.PIEIER3.bit.INTx1 = 1; PieCtrlRegs.PIEIER3.bit.INTx2 = 1;

PieCtrlRegs.PIEIER3.bit.INTx3 = 1;

EINT; ERTM; for(;;){asm("NOP");}

}

interrupt void epwm1\_isr(void){EPwm1Regs.TBPRD = EPWM1\_PERIOD;

EPwm1Regs.CMPA.half.CMPA = EPWM1\_CMPA; EPwm1Regs.CMPB = EPWM1\_CMPB;

EPwm1Regs.TBCTL.bit.CLKDIV = EPWM1\_CLK\_DIV;EPwm1Regs.ETCLR.bit.INT = 1;

PieCtrlRegs.PIEACK.all = PIEACK\_GROUP3;

}

void InitEPwm1Example(){ EPwm1Regs.TBPRD = EPWM1\_PERIOD;

EPwm1Regs.TBPHS.half.TBPHS = 0x0000; EPwm1Regs.TBCTR = 0x0000;

EPwm1Regs.CMPA.half.CMPA = EPWM1\_CMPA;EPwm1Regs.CMPB = EPWM1\_CMPB;

EPwm1Regs.TBCTL.bit.CTRMODE = TB\_COUNT\_UPDOWN; EPwm1Regs.TBCTL.bit.PHSEN = TB\_DISABLE;

EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB\_DIV1; EPwm1Regs.TBCTL.bit.CLKDIV = TB\_DIV1;

EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC\_SHADOW; EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC\_SHADOW;

EPwm1Regs.CMPCTL.bit.LOADAMODE = CC\_CTR\_ZERO; EPwm1Regs.CMPCTL.bit.LOADBMODE = CC\_CTR\_ZERO;

EPwm1Regs.AQCTLA.bit.CAU = 1; EPwm1Regs.AQCTLA.bit.CAD = 0;

EPwm1Regs.AQCTLB.bit.CBU = 0; EPwm1Regs.AQCTLB.bit.CBD = 1;

EPwm1Regs.ETSEL.bit.INTSEL = ET\_CTR\_PRD; EPwm1Regs.ETSEL.bit.INTEN = 1;

EPwm1Regs.ETPS.bit.INTPRD = ET\_1ST; epwm1\_info.EPwm\_CMPA\_Direction = EPWM\_CMP\_UP;

epwm1\_info.EPwm\_CMPB\_Direction = EPWM\_CMP\_DOWN; epwm1\_info.EPwmTimerIntCount = 0;

epwm1\_info.EPwmRegHandle = &EPwm1Regs; epwm1\_info.EPwmMaxCMPA = EPWM1\_MAX\_CMPA;

epwm1\_info.EPwmMinCMPA = EPWM1\_MIN\_CMPA;

epwm1\_info.EPwmMaxCMPB = EPWM1\_MAX\_CMPB;epwm1\_info.EPwmMinCMPB = EPWM1\_MIN\_CMPB;

}