2001 Future Energy Challenge
Final Report

submitted by

The University of Texas at El Paso
El Paso, TX 79968

Team Members

Casey Cook, EE
Juanpablo Gonzalez, EE
Ivan Hernandez, EE
Vanessa Melendez, EE
Angie Rodriguez, EE
Vasilios Vlahos, EE
Brenda Balderas, ME
Carlos Gabaldon, ME
Felix Ortega, ME

Faculty Advisor

Bill Diong

September 1, 2001

Return to Main Document

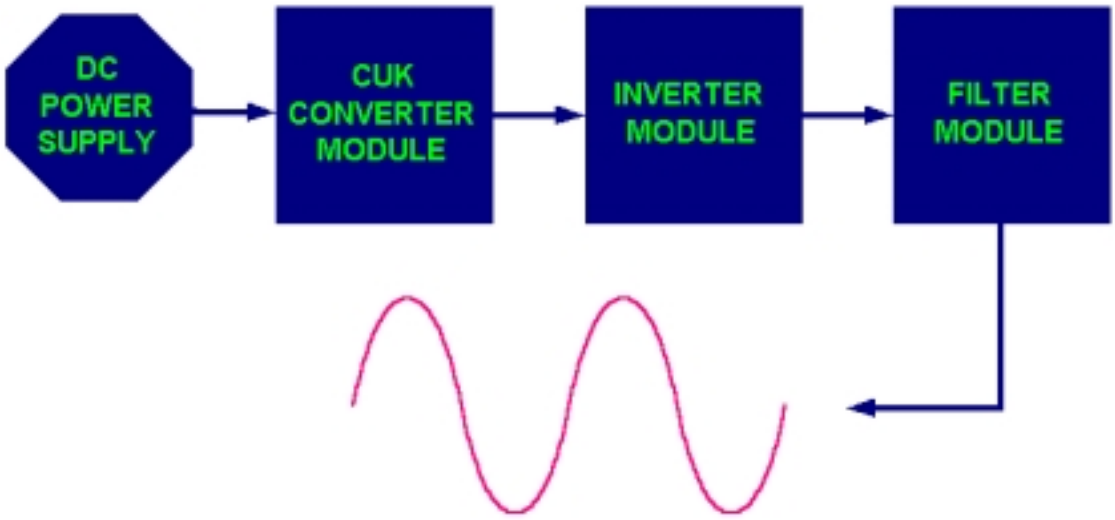
ABSTRACT

Six Electrical Engineering and three Mechanical Engineering undergraduate seniors at the University of Texas at El Paso accepted the Department of Energy's 2001 Future Energy Challenge as their Senior Project. Beginning in August 2000, the students researched various aspects of inverter design, examined the problem at hand, and developed their own prototype inverter, which they presented to the public in May 2001. This paper presents their inverter design.

INTRODUCTION

This project involves the creation of an inverter, which converts DC voltage to AC voltage. The basic motivation for this project was to provide a cost effective way of providing other sources of electricity to homes and/or businesses. The project was to design and build a low cost, high efficiency inverter capable of supplying a split-phase 120V/240V RMS load. This was to be part of a self-contained system providing power to a household from a fuel cell.

There were a total of three teams working on this project, one mechanical engineering team, and two electrical engineering teams. The mechanical side of the project, being the casing, heat sinks and other such things, was completed by the mechanical team in the first semester. Of the two electrical teams, one was responsible for the power conversion hardware and the other for the software/control. During the first semester each team worked somewhat independently of each other, and then during the second semester the teams worked on integrating the three main modules of the project. The three modules are the Cuk Converter Module, the Inverter Module and the Filtering/Sensing Module. This breakdown of the project is shown below:



The responsibilities for each of the above modules are as follows:

CVI POWER	MODULE	ELECTRICAL INGENUITY
Ivan Hernandez	Cuk Converter	Vanessa Melendez
Vasilios Vlahos	Inverter	Angie Rodriguez
Casey Cook	Filtering/Sensing	Juanpablo Gonzalez

Once the integration of the hardware and software for each module was completed, integration of the modules was completed and the project was completed.

CUK CONVERTER MODULE

THEORY

The first main module of this design is the Cuk DC-DC converter, it contains both software and hardware, and the interactions between the two are critical for obtaining the desired output.

DC-DC Converter

The DC-DC converter is one of the three main modules in this system. Switch-mode DC-DC converters are commonly used to convert an unregulated DC input into a controlled DC output at a desired voltage level. It serves as the first of the three main modules in the complete inverter system.

Possible Elimination of Module:

During the developmental stages of the project, the possibility of completely eliminating this module in the design was considered. Research was done on various PWM techniques that accomplished this including the following:

1. Digital Control Strategy by S. Funabiki
    - PWM waveforms are generated by computation of the pulse widths using real time calculations. Pulse widths are computed by sampling and predicting the fluctuating input voltage at regular time intervals

Disadvantages:

- Complicated to implement
- Involved undesired delay in implementation

2. Hysteresis Control Strategy by Kawamura and Hoft

- PWM inverters are controlled by adaptive hysteresis method with an instantaneous feedback loop. A bang-bang controller is used where the output voltage is compared with a reference sinusoidal source, and error signal is passed through a hysteresis block.

Disadvantages:

- Random behavior involved in implementation
- Inverter switching frequency depends on load parameters

Research proved that the best PWM technique to accomplish this is based on a feed-foward approach that alters the modulating signal of the PWM scheme in order to counteract variations in the DC input voltage.

Basic Proposed PWM Technique Block Diagram:

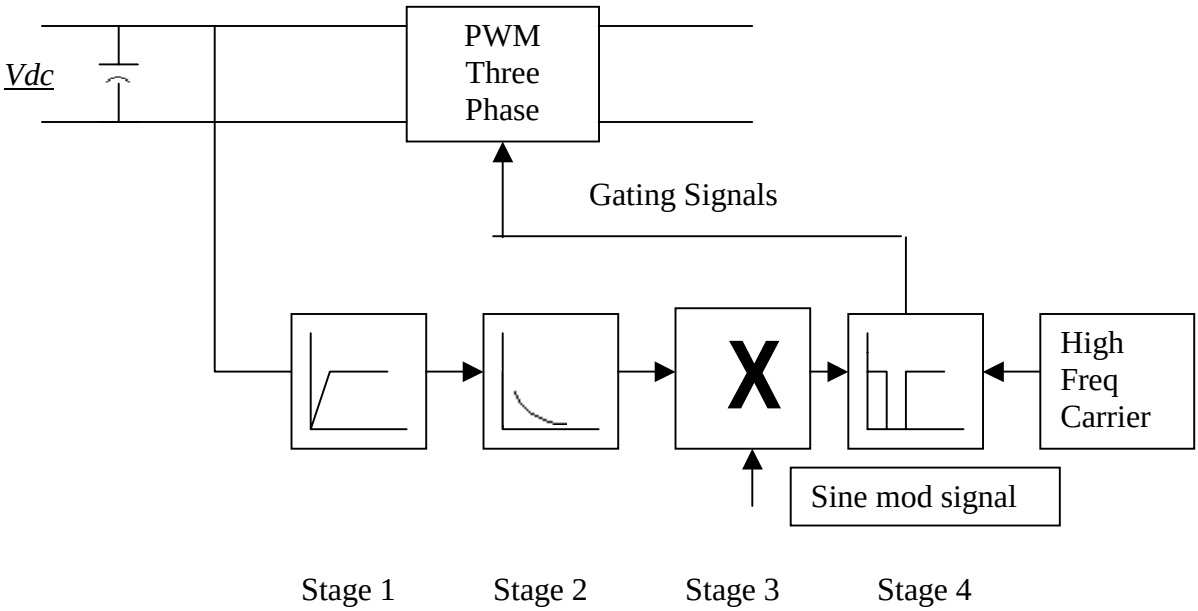


Fig. 1

Stage 1: Senses and amplifies varying component in the input signal

Stage 2: Processing stage corrects for input ripple

Stage 3: Modifies sine modulating signal

Stage 4: Comparator stage generates gating signals

Advantages:

- Applicable to wide variety of non-sinusoidal ripple voltage signals
- No modification to power switches
- Minimum control hardware costs

Disadvantages:

- Response time limited to delay of control circuit
- Spectrum of the input DC current may be distorted
- Tolerance must be watched at a given modulation index

- High when inverter is operating at lower modulation index
- Low when inverter is operating at high modulation index
- If inverter is operated at ideal modulation index 0.8, 20% DC ripple can be tolerated

Although research proved that unregulated DC – regulated AC conversion is possible, the problem encountered was the required amount of amplification. The input power source to be used in the design is supposed to be a fuel cell. As is the case with most renewable energy sources, fuel cells aren't capable of providing high voltage ratings at given high power specs. The fuel cell to be used has a 48VDC/33A power rating. In order to accommodate the required 120V/240V RMS split-phase load at the output of the inverter, its input DC voltage needs to be at a value of 340VDC (for $240V_{RMS}$ output) assuming no losses caused by power dissipation in the circuit. This is the main reason why it was decided a DC-DC converter must be included in the design of the inverter system.

DC-DC Converter Topology:

There are several converter topologies that accomplish unregulated DC – regulated DC conversion. These include:

1. Step-down (buck) converter
2. Step-up (boost) converter
3. Step-down/step-up (buck – boost) converter
4. Cuk converter
5. Full-bridge converter

Comparing and contrasting the five topologies keeping the specifications of the project in mind led to the decision of using the fourth topology, the Cuk converter. The Cuk converter is

obtained by using the duality principle on the circuit of a buck – boost converter. Similar to the buck – boost, the Cuk provides a negative-polarity regulated output voltage with respect to the common terminal of the input voltage. The most important feature of this topology is the fact that a capacitor, instead of an inductor, is used as the primary means of storing and transferring energy from input to the output. This causes energy transfer to occur during both ON and OFF gated switch intervals. Operation is performed at almost zero ripple current at both the input and output of the circuit. These characteristics make the Cuk converter the closest to an ideal DC power supply of any topology.

Cuk Converter:

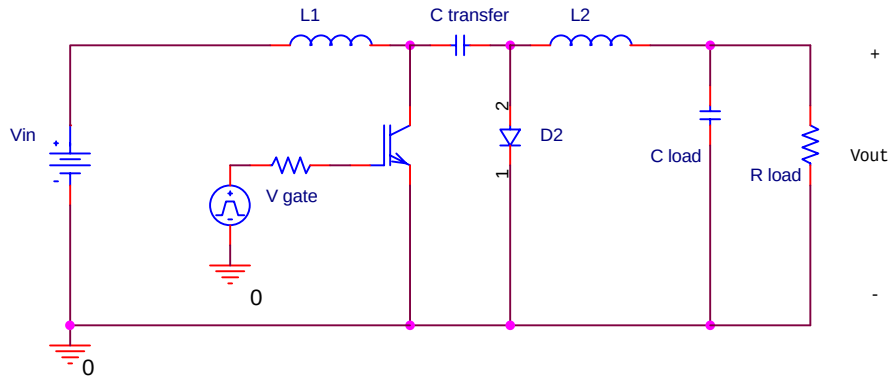


Fig. 2

Cuk Converter Operation:

Operation of the Cuk converter relies on the gating signal being fed into the switch. In reference to Fig. 2 when the switch is OFF, inductor currents of L1 and L2 decrease and flow through the diode thus charging capacitor $C_{transfer}$ with energy from both the input source and L1. This creates a boost effect at the input of the circuit. Energy stored in L2 feeds the output. When the switch is ON, the voltage across the transfer capacitor reverse biases the diode. Inductor currents of L1 and L2 increase and flow through the switch thus discharging capacitor

$C_{transfer}$ and transferring energy to the output and L2. Input/output characteristics of the circuit are described by the following equations:

$$V_{out}/V_{in} = D/(1-D)$$

$$I_{out}/I_{in} = (1-D)/D$$

As one can see, if the duty cycle D is less than or equal to .5, a buck effect occurs. If the duty cycle is greater than .5, there is a boost effect instead relating the output to the input.

Feedback Control of Cuk Converter:

In order to allow the Cuk converter to adapt to input voltage and output load variations, a feedback control has to be created. Controlling the output voltage is to be done by adjusting the duty cycle of the switch.

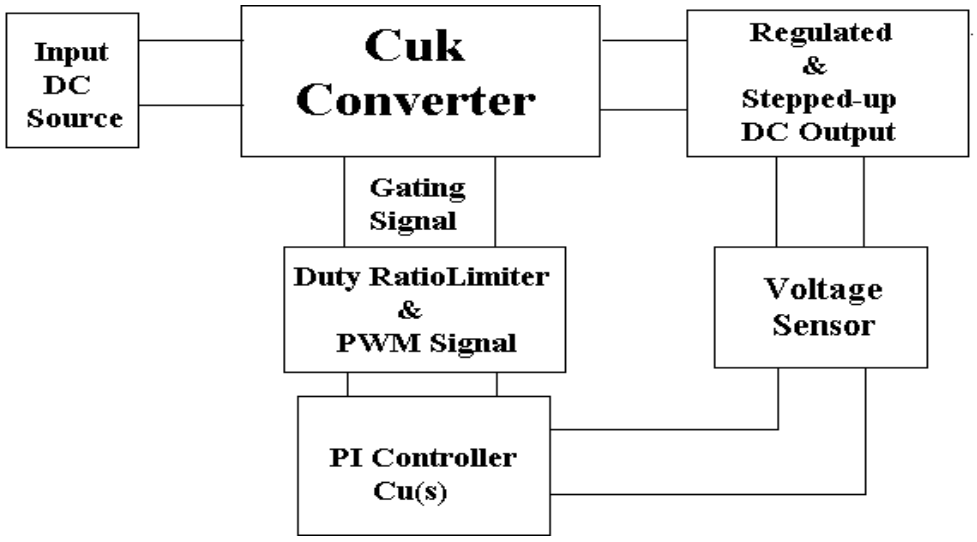


Fig. 3

The voltage at the output is sensed and this information is fed into a PI controller that is to provide the PWM control signals. This in turn varies the duty cycle of the PWM gate drive signal to the converter switch to adjust to the input voltage and output load variations. Due to the fact that the system is nonlinear, design is based on a linearized model that employs transfer

functions for the description of the Cuk converter and the PWM scheme. The following MATLAB code computes all the parameters necessary to implement the PI control:

```
% PROGRAM THAT FIGURES KP AND KI VALUES FOR PI FEEDBACK CONTROLLER
% THE EXTENDED LINEARIZATION IS USED FOR DESIGNING THE CONTROLLER
% ANALYSIS BASED ON TRANSFER CAPACITOR VOLTAGE FEEDBACK REGULATION

% DEFINE INPUT PARAMETERS

E=20;
L1=.0002;
L2=.008;
C1=5e-6;
R=2000;
B=E/sqrt(L1);
U=.944;

% DEFINE STATE VARIABLE VALUES

% X1=I1*sqrt(L1);
% X2=V2*sqrt(C1);
% X3=I3*sqrt(L2);

% DEFINE CONVERTER PARAMETERS

W1=1/sqrt(L1*C1);
W2=1/sqrt(L2*C1);
W4=R/L2;

% EQUILIBRIUM POINTS OF AVERAGE MODEL ASSUMING CONSTANT VALUE U FOR DUTY
RATIO

Z1=((W2^2)*B*(U^2))/((W1^2)*W4*((1-U)^2));
Z2=B/(W1*(1-U))
Z3=(W2*B*U)/(W1*W4*(1-U));

% CROSSOVER FREQUENCY

A=.5*(((U^2)*(W2^2))+(((1-U)^2)*(W1^2))-((2-U)*(W4^2)));
BE=2*(W1^2)*(W4^2)*((1-U)^2);
W0=sqrt(A+(sqrt((A^2)+BE)));

% PI PARAMETERS KP AND KI

KP=((.4*W1*(W4^2)*((1-U)^2))/B)*(abs(((W1^2)*((1-U)^2))-
(W0^2)))/abs(((W4^2)*(W1^2)*((1-U)^2))-((W0^2)*(W2^2)))

KI=((W1*(W4^2)*((1-U)^2))/(4*pi*B))*((abs(((W1^2)*((1-U)^2))-
(W0^2)))/abs(((W4^2)*(W1^2)*((1-U)^2))-((W0^2)*(W2^2)))*W0)
```

Final Design:

The final design includes feedback control making it possible for the Cuk converter to be annexed to the rest of the inverter design:

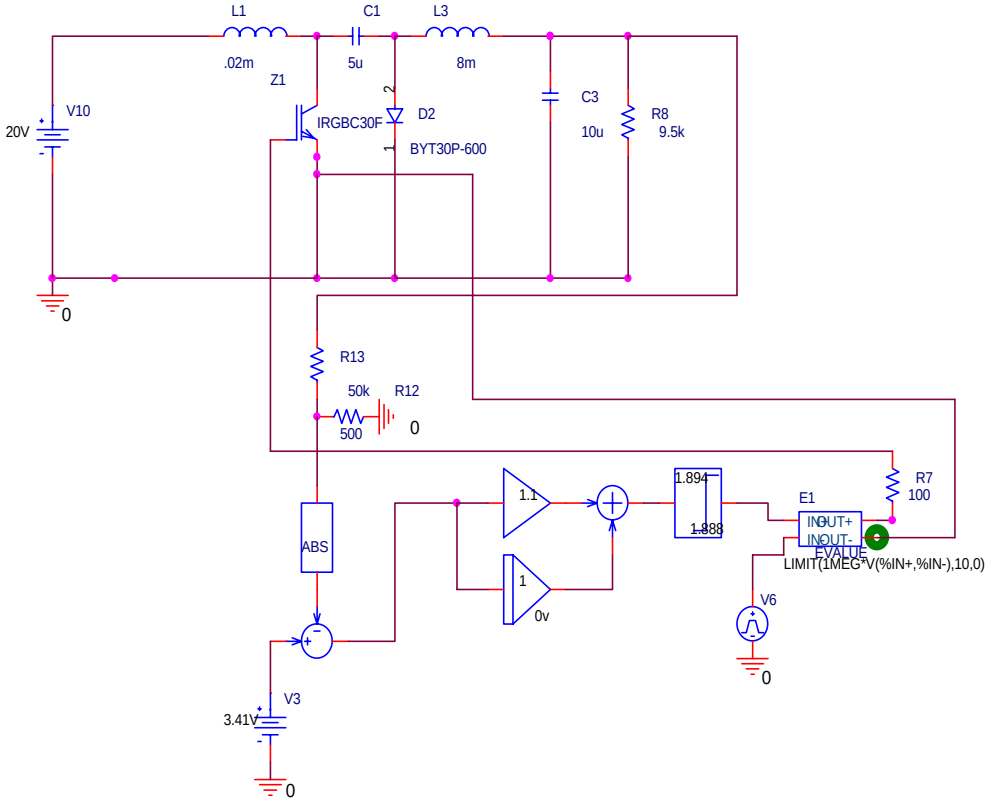


Fig. 4

The following is a table describing the effectiveness of the previous design. The load on the converter was varied and output voltage measurements were taken with and without the feedback controller.

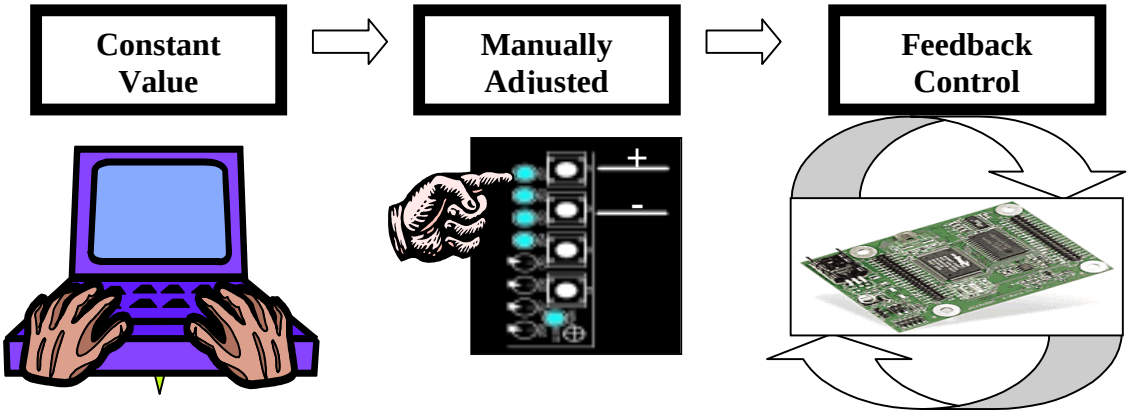
Output Voltage Regulation for Cuk Converter		
Load (Ohm)	Open Loop (V)	Closed Loop (V)
250	148	328
500	205	331
1k	254	334
1.5k	285	335
2k	343	338
2.5k	367	338
3k	408	342
3.5k	451	346
4k	483	347

From this information, it is clear that the PI controller accomplished the task of maintaining the output voltage constant (to within ± 5%) even when the load was varied. This was done by varying the duty cycle of the gating signal fed into the Cuk converter switch in order to counteract those output load changes.

SOFTWARE

In order to implement the controls for the Cuk converter, it was necessary to generate a series of pulses. The duty cycle of the pulses (defined as the ratio of the on duration to the switching time period) was required to be adjusted in order to achieve the necessary output voltage. The signals were generated within a software program using the jackrabbit microcontroller by comparing a control voltage with a constant peak repetitive waveform (sawtooth).

The process of controlling the switch for the Cuk converter involved three steps as shown in the diagram below:



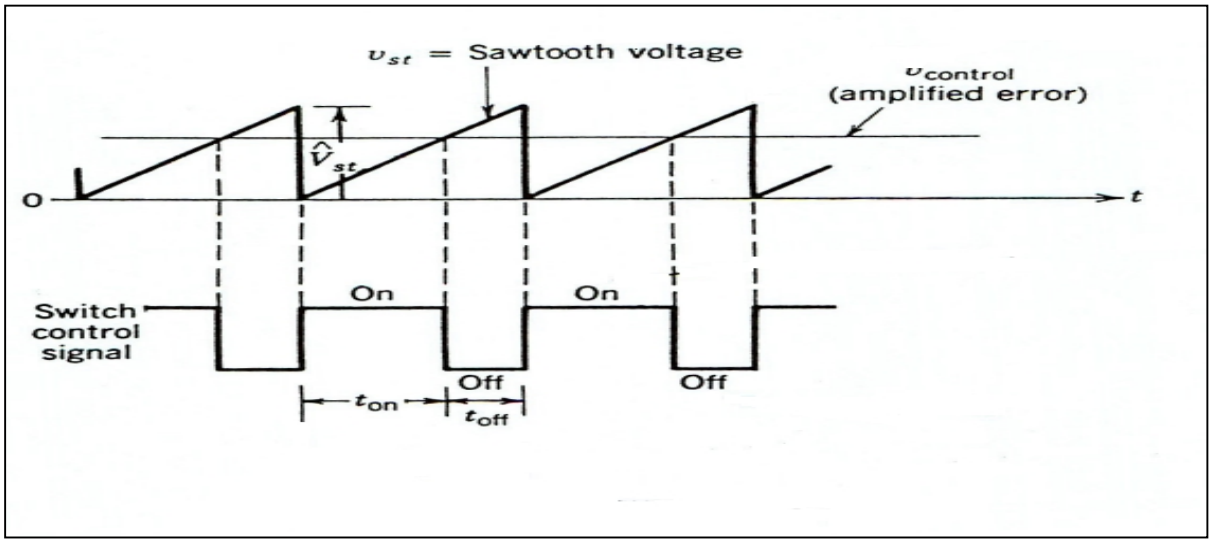
PWM Cuk converter generation of pulses – Design Process.

The first step of the design process involved creating a program generating a signal with a constant duty cycle. The percentage of the duty cycle of the signal is specified within the software before downloading it to the microcontroller. The pulse generated has a varying duty cycle from 5% to 95%. The frequency of the signal had to be modified since when integrating with the hardware part of the Cuk module a high audible noise was present. The noise was almost eliminated by increasing the frequency of the pulses generated. The final frequency achieved by the program is 7.3 kHz. (Appendix-Constant Duty Cycle for Cuk Converter Program)

The second step in the design process required creating a program that could adjust the duty cycle of the pulses by responding to manual pressing of the switches on the development board of the microcontroller. The duty cycle of the signal created is increased or decreased according to the needs of the system and it starts with a duty cycle from any given value (within 5% to 95%) by indicating the desired percentage within the software before downloading the program to the microcontroller. The final frequency of the pulses generated is also 7.3 kHz. (Appendix-Manually Adjusted Duty Cycle for Cuk Converter Program)

The third step of the process involved the creation of a control switch signal by comparing a signal level control voltage Vcontrol with a repetitive waveform. The final

frequency of the signal generated is 7.3 kHz. (Appendix-Vcontrol Dependant Duty Cycle for Cuk Converter Program).



Comparator Signals
Power electronics; Mohan, Underland, Robbins, Second edition, 1995. Wiley, Inc.

The program implemented with the feedback control involved sensing a DC signal on an analog input and varying the pulse duty cycle by comparing that DC voltage to the sawtooth signal created within the program.

The equation relating Vcontrol to the duty cycle is provided below:

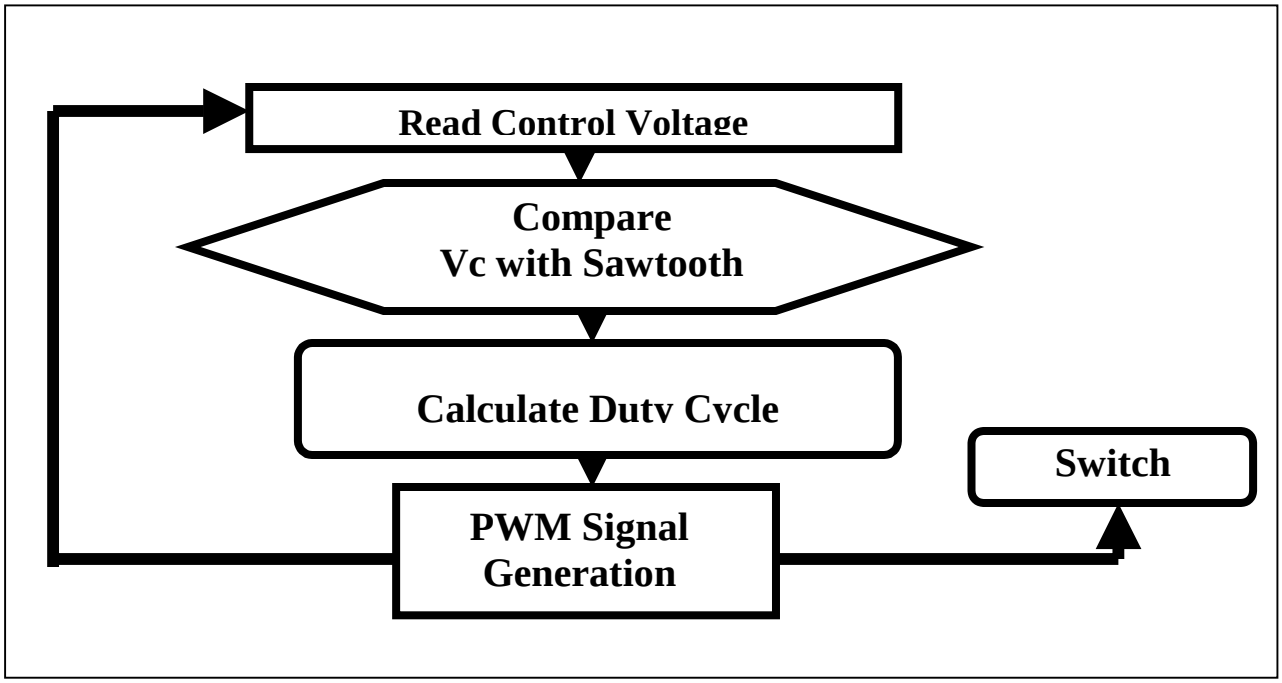
$$Vout = \frac{D}{1-D} Vin$$

$$D = \frac{Ton}{Ton + Toff} = \frac{Ton}{Tperiod} = \frac{Vout/Vin}{(Vout/Vin)+1} = \frac{Vcontrol}{Vsawtooth}$$

Finally, the equation relating the duty cycle to the value of Vcontrol is the following:

Duty cycle = ((Vcontrol)*(0.16666))

A flow chart illustrating each step involved in the PWM program for the Cuk converter is shown below:



Basically, the program reads the DC control voltage from the analog input 0. The voltage read is then compared with the sawtooth signal generated and the duty cycle is calculated. After that, the PWM signal with the required duty cycle is generated and sent to the switch from the analog output PE7 on the microcontroller. The process is then repeated reading again the control voltage and creating continuously the required PWM signal in a closed feedback loop DC-DC converter system.

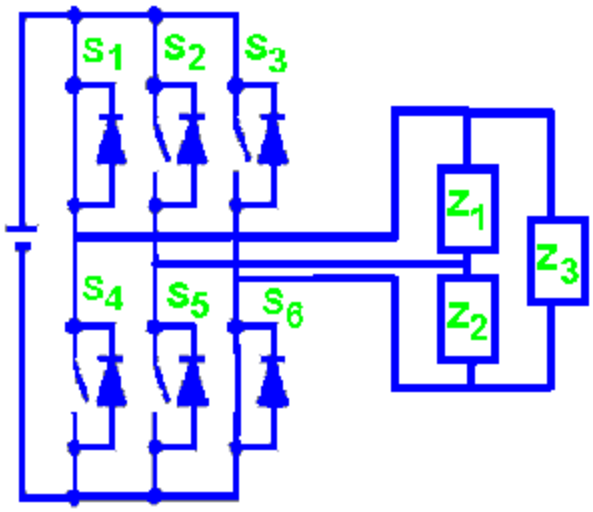
INVERTER MODULE

THEORY

The second main module of this design is the inverter. The purpose of this module is to convert the output of the Cuk converter, which is DC, into AC. This AC output is then fed into a filter in order for a clean sine-wave to be obtained. The module contains both software and hardware, and the interactions between the two are critical for obtaining the desired output.

The inverter is what is called a three-wire single phase configuration. It contains three legs of two switches each, resembling a three-phase topology, leading to a dual output across the

power terminals, resembling a split-single phase load. The following figure indicates the topology used:



The first and third legs of the design comprise the differential control mode. Its purpose is to ensure that the voltage across Z3 remains always constant and equal to 240 Vrms. During the first half-cycle of operation, switches S1 and the inverse of S3 are turned on, thus connecting the input DC voltage to the connected loads, with current flowing through it. During the last half-cycle of operation, the previous two switches are now turned off, while switches S3 and the inverse of S1 are turned on. This causes current to flow in the opposite direction than the previous half-cycle, completing one full cycle.

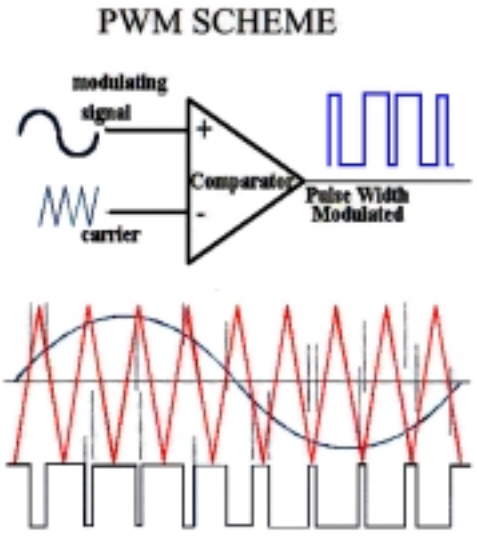
The middle leg of the inverter comprises the common control mode. Its purpose is to ensure that impedances Z1 and Z2 see exactly 120Vrms each, regardless of their values. Assuming that these two loads are always balanced would make this control mode unnecessary, but in reality these two may not always be the same. In that case, the common mode control will force both of the loads to have the same potential difference across their terminals.

Both of these control modes are essential for the proper operation of the inverter, capable of delivering power to any type of load, efficiently and effectively. All the necessary adjustments required to take place so that the inverter can accommodate for differences in loads are done instantaneously through the micro-controller, which will adjust its program appropriately in order to satisfy the different load conditions present.

The switching scheme employed is the Bipolar Pulse Width Modulation (PWM). The micro-controller produces switching pulses resembling the one below:

Note that the pulses have varying duty cycles, ranging from very small at the beginning and end of the sequence, to very large in the middle of it. Moreover, the amplitude of this output signal varies from +Vmax to –Vmax during any transitions between adjacent pulses. The benefits of this switching scheme as opposed to a simple square wave scheme (pulses with constant fifty-percent duty cycle) are that the output closely resembles a sine wave (called modified sine wave) and it is much easier to filter, requiring smaller and cheaper filter components.

The theory behind bipolar PWM can be explained as follows: two sine waves and a triangular wave are created. The two sine waves have different frequencies. One wave will be used to control the differential mode legs and the other the common mode leg. The sine waves and triangular wave are compared and two pulse signals are generated, one from each sine wave. The pulse is determined by Vtri > Vsin. This comparison generates a pulse whose duty cycle varies from approximately 3% to 98%. The following figure is representative of this process, but an external comparator is not used, instead the comparison is done in software using the micro-controller.



HARDWARE

The hardware part of the inverter is composed of power electronics components interconnected in such a way so as to realize the circuit mentioned earlier.

The components used in the design of this system fall in one of the following two categories: they are either power semiconductor switches or their role is as secondary supportive components. The former are the most important part of the inverter. Without the switches no inversion would take place, and DC would not be converted into AC. The role of the latter is to facilitate the operation of the switches, and ensure that they will function effectively and efficiently throughout their operating life.

SWITCHES

The switch of choice for this application is International Rectifier's IRG4PC40UD. The switch is an IGBT (Insulated-Gate Bipolar Transistor) and has many benefits compared to other power devices such as MOSFETS or BJTs, because it combines the best characteristics of both of these commonly used power semiconductor devices. It resembles a MOSFET in the sense that it is a voltage controlled device, and also a BJT in the sense that it has two back-to-back p-n junctions (NPN) thus having superior current handling capability. It is a rugged device, capable of withstanding voltages up to 600V and currents up to 40A. In addition, it has a built-in fast recovery diode, useful when inductive loads and stray inductances are present in the circuit, and has one of the lowest on-state voltage drops available in the market today (1.72V).

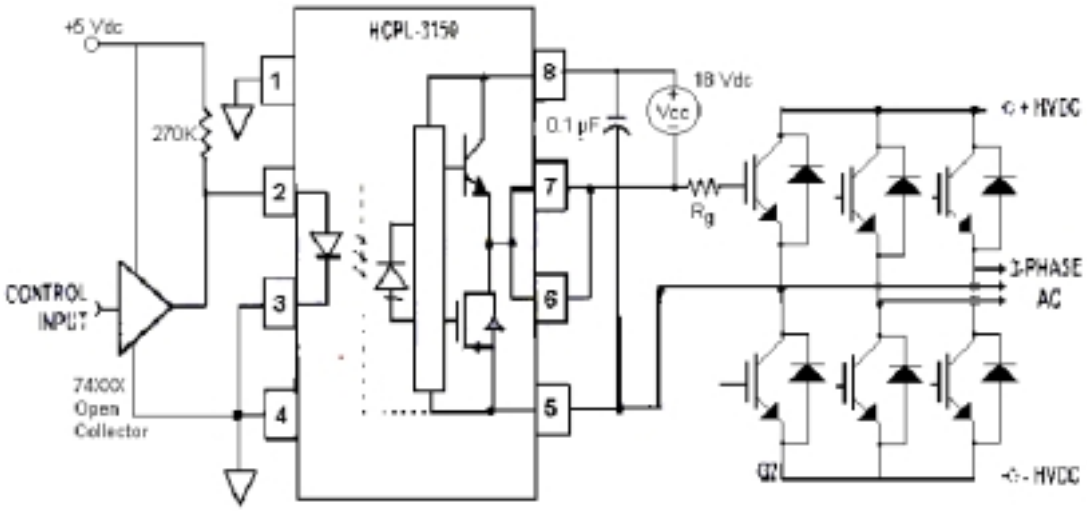
There are two types of losses taking place in semiconductor devices, stemming from different processes and mechanisms: static and dynamic. Static losses occur when the device is in the on state. In this state, the device behaves as a small resistor, having an associated internal resistance that dissipates some of the power flowing through it. The IRG4PC40UD has an on-state voltage drop of 1.72 V associated with it during normal operation. Multiplying the above with the value of the instantaneous current flowing through it will give the static losses of the device. On the other hand dynamic losses occur when the switch transitions from the on to the off state, or vice versa. During that period of time, the current decreases from some constant value to zero, while the voltage instantaneously increases from a very small value to the full Vcc

seen across the device (and vice versa). Plotting the changes in current and voltage with respect to time, one would see that their intersection forms a curve. The area under the curve, which can be obtained by integrating the current and voltage waveforms with respect to time, represents the dynamic losses occurring in the switch. The behavior of a system in which power losses are important can be significantly improved by selecting devices having as low a voltage drop as possible (thus minimizing static losses) and ensuring that fast transitions between the on and off states (or vice versa) are taking place. This means that the area under the current-voltage curve will be as small as possible, thus yielding low dynamic losses. For more information, consult the Appendix.

Due to the delicate nature of the devices and the commonly observed degradation of this type of devices associated with high operating temperatures, generic TO-247AC package heat-sinks were utilized in order to provide temporary thermal relief and dissipate the excess heat to the surrounding ambient environment.

GATE DRIVERS:

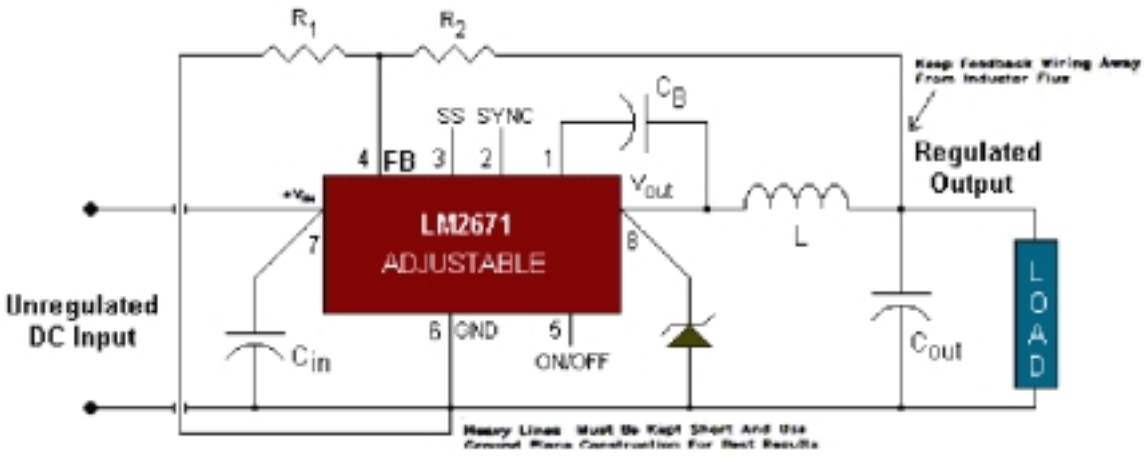
In order for the switches to be properly turned on and off following the instructions of the micro-controller, a gate driver circuit is needed. Agilent's HCPL 3150 opto-coupler (also known as an opto-isolator) was used due to its low cost and high durability. The diagram below indicates the configuration of this device and its interconnection to the rest of the circuit.



The opto-coupler is the interface between the control and the power circuit, effectively isolating the two and protecting both from any disturbances that if occurred on one side could spread and affect the other and possibly cause irreparable damage. The most important thing to notice from the above diagram is the floating DC power supply connected across the 0.1 μf capacitor between terminals 5 and 8. The amplitude of this power supply determines the amplitude of the output PWM signal, originally received from the micro-controller. Moreover, since this power supply is always referenced with respect to the emitter of the IGBT, it will always be able to turn the IGBT on, no matter what the voltage drop at that reference point is, because the difference between the gate and emitter will always be Vcc (thus sufficiently high) to turn the device on or off.

VOLTAGE REGULATOR CHIP

The floating power supply across terminals 5 and 8 of the opto-coupler mentioned above, is realized using the LM2671 step down voltage regulator. Its configuration is shown in the following figure:



The purpose of the external capacitors and inductors is to store energy temporarily while transferring it to the output and to minimize the output ripple as much as possible. The two external resistors R1 and R2 determine the gain of this device, given by

$$Vout = Vref + (1 + \frac{R2}{R1})$$

Where Vref = 1.21V. Moreover, Cin = Cout = 120 μF, 35V aluminum electrolytic capacitors. Cin's purpose is for prevention of large voltage transients appearing on the input, while Cout's purpose is reduction of ripple at the output. In addition, the C_B capacitor's purpose is to develop the necessary voltage to turn the switch gate inside the chip on fully. A 0.01 μF, 50 V ceramic capacitor is used as C_B. The inductor L1 is a 33 μH whose purpose is to store energy during the intervals when the internal switch is turning on and off. Finally, the Catch Diode D1, is a 1N5821 30V, 3A diode clamping the output.

For this design, the input voltage is ranging from 20V to 24V (representing the battery bank) and will be stepped down to 18V as required by the manufacturer of the opto-couplers (this will be the chip's Vcc) in order to successfully gate the IGBTs.

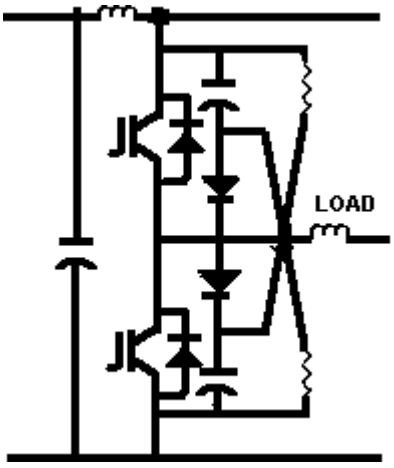
SNUBBER CIRCUIT

When inductive loads are present in a circuit, and the current flowing in them is abruptly shut off, there is still some stray inductance present in the circuit. This excess energy stored in the inductors must be dissipated somewhere, usually across the power semiconductor switch. When this happens, one can examine the voltage across one of these switches and notice the generation of a large-amplitude spike across the terminals of this device. Since the amplitude of the spike is proportional to the stray inductance of the circuit, the spike could easily exceed a few hundred volts in magnitude and thus force the device to function outside its safe operating limits. Excessive spiking can cause damage to the device and negatively affect the quality of the overall system. Therefore there is a great need for if not eliminating spiking, then to minimize it as much as possible.

As a means for protecting semiconductor devices from spikes of this sort, circuit designers implement snubber circuits. A snubber circuit's purpose is to reduce the di/dt and dv/dt for a particular device of interest. In practice, instead of the trapped energy being dissipated across the terminals of the switch (and seen as a spike on the scope), it is now safely dissipated in the snubber circuit instead. There are many different topologies that can be found and utilized, each one recommended for specific applications and having different advantages and disadvantages, some of which are outlined below:

	Decoupling Capacitor	**Discharge Resistance Decoupling Capacitor**	**RCD Snubber Clamp Circuit**	**RCD Charge Discharge Snubber Circuit**
Advantages	Low snubber losses Favorably effects turn-off and turn-on voltage stresses Module compatible caps effective limiting voltage transients	Low snubber losses Reduces voltage turn-off and favorably effects turn-on Faster switching	Low snubber losses Reduces turn-off voltage overshoots and favorably affects turn-on No oscillation in DC bus	Reduces turn-off voltage overshoots Could reduce turn off losses No oscillation in DC bus
Disadvantages	Severe voltage-current oscillations in the DC bus	Less effective protection High recovery voltage spikes due to snubber diode	More components	More components Increased turn-on losses Complicated component selection

The one found suitable for this design is the RCD Snubber Clamp Circuit configuration in the following figure:

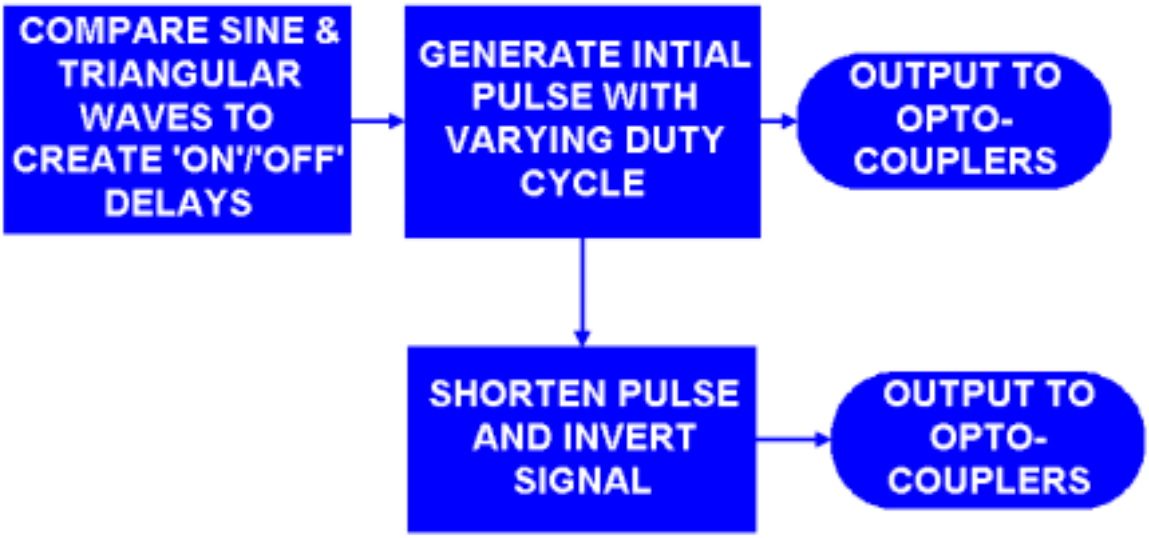


It is composed of the ultra-fast recovery diode HFA50PA60C, capable of withstanding currents up to 25A per leg, two polypropylene capacitors of 0.27μF and 0.47μf in parallel, both rated at 400VDC, and finally a 6.8Ohm, 2W power resistor. Whenever the voltage across the device the snubber is supposed to protect exceeds the rail voltage, the snubber diode will become forward biased and charge the polypropylene capacitors. When the abnormality has been eliminated, the excess energy stored in those capacitors will be safely dissipated through the power resistor and not across the switch. Care must be taken when designing a snubber circuit in order to select parts that will not cause any oscillations in the DC bus. Fast and ultra-fast recovery diodes are ideal for this type of applications. Moreover, the snubber capacitors must be made out of polypropylene and the resistors must have minimum inductance and be recommended by the manufacturer for snubber applications. To demonstrate the significance of this add-on circuit, simulations are included of the implemented circuit with stray inductance of 1 mH. The first printout shows the voltage across one of the switches without snubbers, while the second printout is for a properly designed snubber circuit placed across that same switch.

SOFTWARE

As mentioned earlier in this report the pulses to control the switching scheme of the inverter were generated using the BL1800 (Jackrabbit) Micro-controller. The way this was done was by comparing the values of the triangular wave and the sine wave as mentioned earlier in the PWM theory and creating “on” and “off” states. Since the micro-controller is programmed in

Dynamic C (a form of C programming specifically for the Jackrabbit), these states were generated using loops. The loops decreased or increased the on time and off time respectively to create a varying duty cycle. A copy of the program used may be found in the Appendix. Not much more can be said about the software side because following the theory one can see how the generation of these pulses was implemented. The following flow diagram illustrates how this program basically works.



FILTERING/SENSING

Low-Pass Output Filter

The Output Filter design is based on two main properties of the inverter switching scheme. These are the Frequency Modulation Index and the Amplitude Modulation Index. These are usually chosen to maximize operating efficiency and make the selection of filter components easier.

The Frequency Modulation Index (m_f) determines the location of the harmonics present in the output. Lower mf values correspond to harmonics present at frequencies closer to the fundamental, while higher values "push" these harmonics farther away. Hence, higher switching frequencies are desirable when designing the output filter. In addition to filtering considerations,

switching frequencies should be chosen to be outside the audible spectrum in order to avoid creating unwanted noise. This design employs a switching frequency of 20,100Hz, a fundamental frequency of 60Hz, and hence a Frequency Modulation Index of 335.

The Amplitude Modulation Index determines the amplitude of the harmonic components relative to the fundamental. Higher values of m_a correspond to higher harmonic amplitudes and hence higher Total Harmonic Distortion (THD). Generally, m_a values are kept less than 1 in order to ensure that the inverter is operating in the linear range. Based upon the voltages available from the CUK converter stage of this inverter, the m_a is assumed to vary from 0.8 to 1.0.

The locations and amplitudes of the first three harmonics are tabulated below. For the purposes of designing the output filter, it is assumed that sufficient attenuation of the first three harmonics will imply sufficient attenuation of the remaining harmonics. A steep attenuation slope in the stop-band should make this possible.

$m_f = 335$ $m_a = .8$

Frequency	Amplitude
60Hz ,fundamental	169.7 V (120Vrms)
19,980 Hz ,fundamental * ($m_f - 2$)	46.6V
20,100 Hz ,fundamental * m_f	173.6V
20,220 Hz, fundamental * ($m_f + 2$)	46.6V

Total Harmonic Distortion is calculated by $THD = \sqrt{\sum_{i=2}^{n} H_i^2}$. For THD < 5%, the following must be true: $.05 * (169.7) = 8.485V$ and using the first three harmonics, $8.485 \geq \sqrt{(A^2} + (3.72A)^2 + A^2)$, A = 2.132. The first three harmonics must be reduced as tabulated below.

Frequency (Hz)	Attenuation (V)	Attenuation (dB)
19,980	-44.6	-33
20,100	-165.7	-45
20,220	-44.6	-33

Higher order Butterworth filters exhibit flat pass-bands, and steep attenuation in the stop-band. Existing MATLAB functions will provide the necessary order of a Butterworth filter with desired attenuation at a given frequency. The following equation can be solved to determine the cutoff (3dB) frequency for a low-pass Butterworth filter of the order determined by MATLAB.

$$Attenuation = 10\log[1+(\frac{\omega}{\omega_c})^{2n}]dB$$

For n = 4, the 3dB frequency (ωc) is 5504.2 Hz.

The 4th order Low-Pass Butterworth Transfer Function is

$$H(s) = \frac{1}{s^4 + 2.613s^3 + 3.41372s^2 + 2.613s + 1}$$

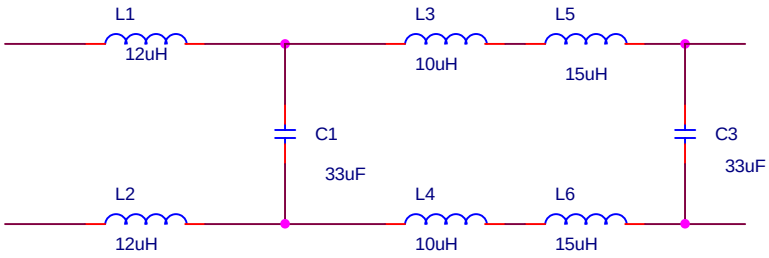
The general 4th order Low-Pass Transfer Function is

$$H(s) = \frac{\frac{1}{C_2L_2C_4L_4}}{s^4 + s^3(\frac{R_s}{L_2} + \frac{1}{R_lC_4}) + s^2(\frac{1}{C_2L_2} + \frac{1}{C_4L_4} + \frac{1}{C_2L_4} + \frac{Rs}{Rl}(\frac{1}{C_4L_2}) + s(\frac{1}{L_2C_2C_4Rl} + \frac{2Rs}{L_2L_4C_4} + \frac{1}{L_4C_2C_4Rl}) + (\frac{Rs}{Rl}+1)(\frac{1}{L_2L_4C_2C_4})]}$$

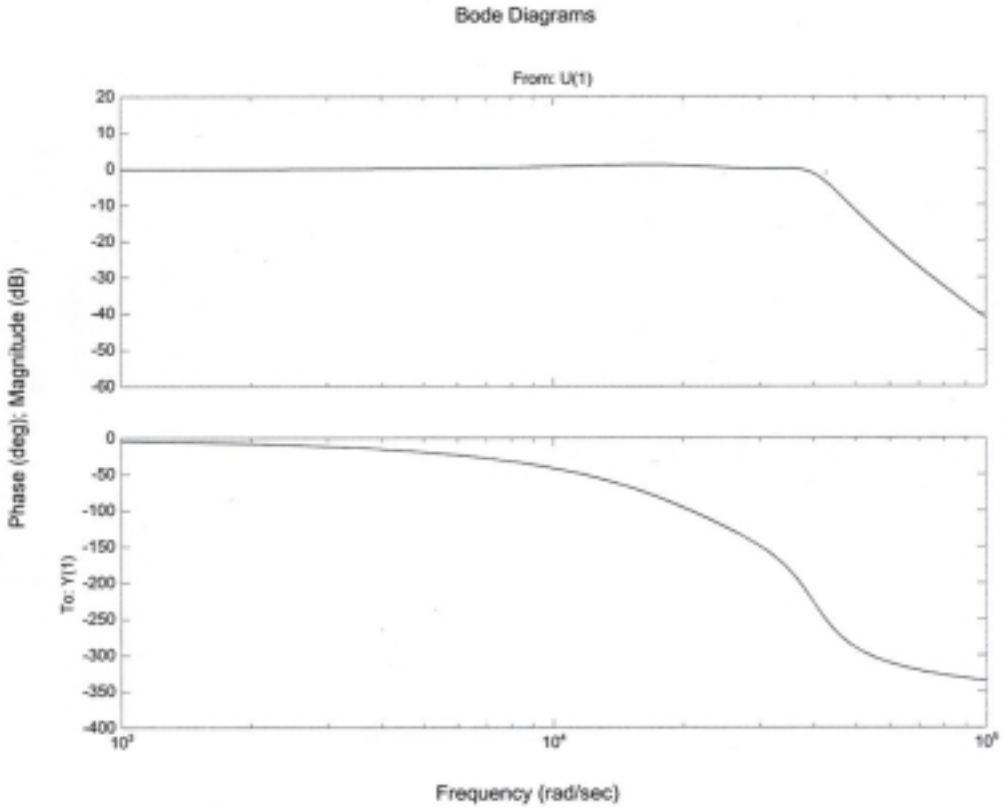
An approximation to a Low-Pass Butterworth filter is determined with a spread-sheet by setting these two equations equal and making several assumptions. These assumptions are:

1. C2=C4 – the capacitances are equal, and chosen from available component values.
2. Higher order terms dominate the behavior of the polynomial.

The final realizable component values are C2=C4=33uF, L2=26uH, L4=50uH.
The Bode plot of the filter with these values shows that this filter very closely approximates Butterworth behavior (flat pass-band, steep roll-off in the stop-band),
and provides sufficient attenuation at the desired harmonic frequencies.



Output Filter



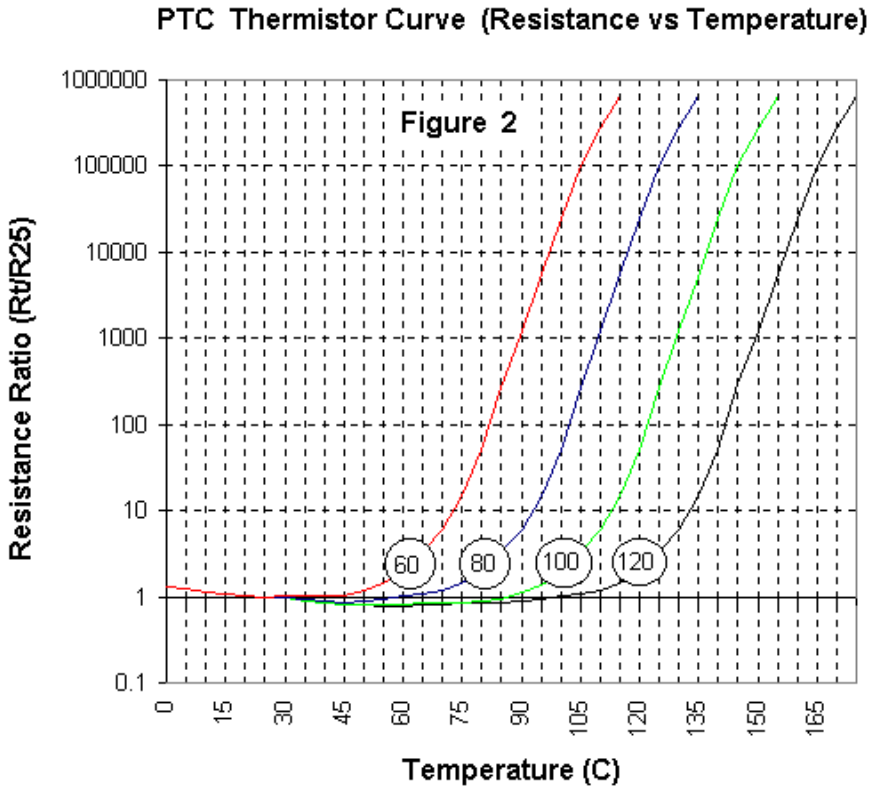
Output Sensing

The output voltage is floating AC, and the micro-controller is ground referenced making the use of an isolation device necessary. This is accomplished for each output by a PCB mountable 115/220V (primary)/5V(secondary)-60Hz transformer. The secondary is ground referenced. The 5V AC signal is converted to a DC value by a Maxim RMS to DC converter. The DC signal simplifies hardware and software because it does not necessitate a Phase Locked Loop (PLL) for comparison to the reference signal. The DC output is adjustable based on the inverter AC output, and for this purpose a filtered 60Hz 120VRMS inverter output corresponds to a 2.7VDC sensed output.

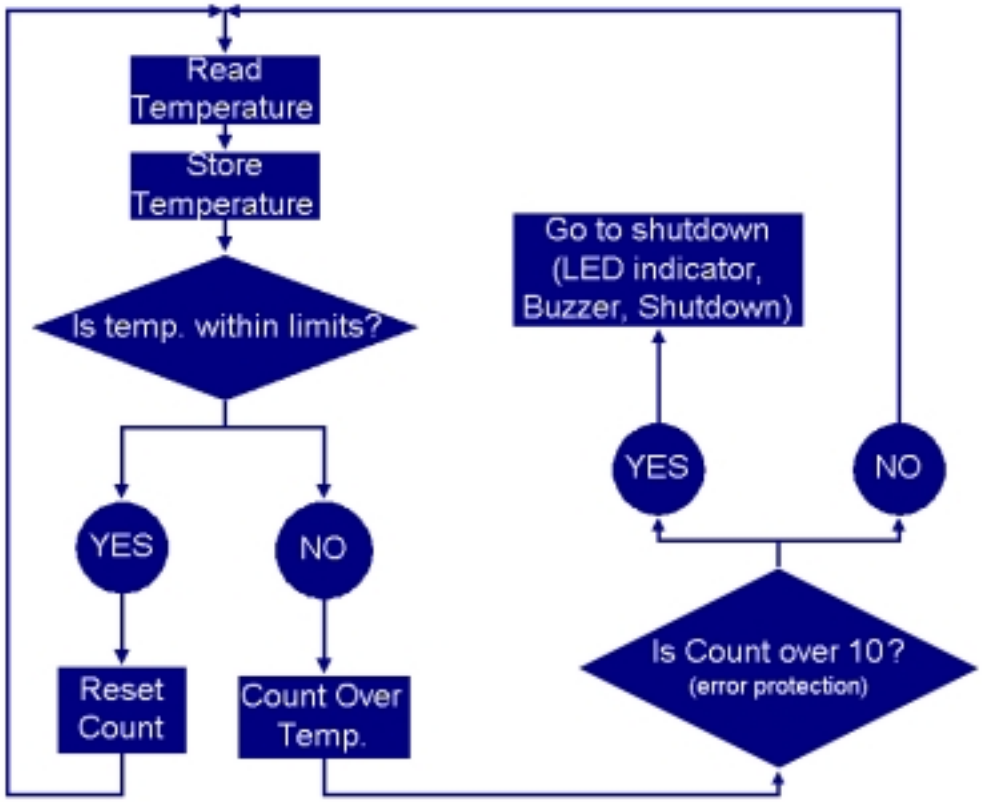
Current sensing may be accomplished in many ways. Various methods were considered including Hall Effect, other Closed Loop Magnetic methods, and various series resistance methods. For reasons of cost and simplicity, a series resistance/isolation amplifier was chosen as the method of current sensing. The voltage across a precision .005Ω resistor, representing the current in the load, is isolated by an Agilent opto-coupler and directly interfaced with the micro-controller. A 7 Amp current corresponds to a 2.8V p-p voltage signal.

SOFTWARE

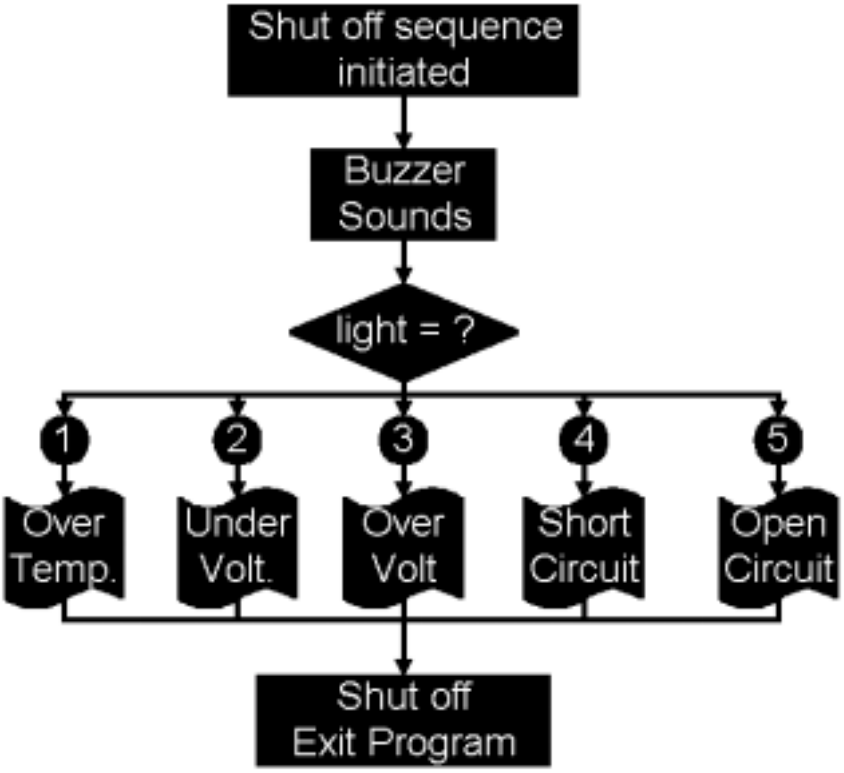
The major components that were involved in the sensing were the over/under voltage, over/under current, loss of input signal, temperature, and the data acquisition. All of these were implemented by using the BL1800 Jackrabbit micro-controller. For the temperature sensing a thermistor was used to measure the temperature of the inverter. The type that was used was a positive temperature coefficient (PTC), which is a solid-state device that exhibits several decades of resistance change over a narrow temperature range. A property of this device is that it can be able to sense the overheating of the inverter. The resistance of the thermistor changes remarkably at a certain temperature level. It is this change in resistance that changes the voltage of the thermistor, which lets the Jackrabbit know the temperature of its surroundings. The figure below illustrates typical 60, 80, 100, 120 C switching temperature curves.



Since the operating temperature of the Jackrabbit is from – 40° to 70 ° Celsius (C) the PTC of 60° C was chosen (Room temperature is about 20° C). The general sequence that was coded for the Jackrabbit to follow was that the voltage of the thermistor would initially be read. Then this value would be stored and checked to make sure that it is within limits for proper operation. If within the limits that were set, then the count (which was initially set to zero) would be reset to zero and if not within the limits the count would increase by one. The maximum that the count could reach was ten, this was done so any miss-readings by the Jackrabbit could be over looked. The following is a flow chart for the temperature sensing.



The light emitting diodes (LED) of the Jackrabbit were used to indicate the status of the inverter. The conditions that were of interest were the over/under voltage, over/under current, temperature, and loss of input signal. In case the Jackrabbit detected any of these conditions it would initiate the shutdown sequence by sounding a buzzer and then lighting the appropriate LED. After this the Jackrabbit would shut down the system. The flow chart that follows is for the sequence that is initiated by the Jackrabbit when an abnormal condition exists.



The data acquisition part of the project was done by use of a standard input/output window of the Jackrabbit. All the major components that were being sensed were displayed on this window as the Jackrabbit operated.

The implementation of the voltage and current sensing was a two-part process. The first part (which was hardware) was to lower the current and voltage so that the Jackrabbit would be able to read the values. Then in the second (which was the software) the values were used to make sure that they were within the defined limits for proper operation. A copy of the program used for the implementation of this portion may be found in the Appendix.

THERMAL MANAGEMENT AND PACKAGING

Design constraints for the inverter thermal management and packaging were given for a 10kW output dc-ac inverter. The final product was not to exceed $500 when scaled to a 10 kW design in high-volume production. Also, the system was to be less than 50 L in volume and weigh less than 32 kg, not including energy sources or batteries. As informed, the electrical engineering group was to build a prototype with a steady state output of only 1.5 kW operating at a minimum efficiency of 90%. It was assumed that a total of 175 W of heat would be produced by the power electronic devices. Downscaling of requirements for this prototype inverter power level was accomplished by gathering information on several inverters out on the current market (summarized in Table 1).

		Statpower	Statpower	Statpower	Statpower	Trace	Trace	Trace	Trace
Output	**W**	1000.00	1800.00	2500.00	3000.00	2500.00	4000.00	4000.00	5500.00
Retail Price	**$**	685.00	1000.00	1950.00	2250.00	2275.00	2975.00	2975.00	3400.00
Volume	**L**	14.42	14.42	28.02	28.02	50.11	50.11	63.05	63.05
Weight	**kg**	6.50	7.50	14.50	14.50	50.00	50.00	57.00	57.00
		Tripp-Lite	**Tripp-Lite**	**Tripp-Lite**	**Tripp-Lite**	**Whistler**	**Whistler**	**Whistler**	**Whistler**
Output	**W**	400.00	550.00	750.00	1800.00	300.00	1000.00	1500.00	2500.00
Retail Price	**$**	160.00	260.00	390.00	700.00	110.00	350.00	550.00	850.00
Volume	**L**	1.84	8.03	8.03	10.86	2.41	7.37	10.10	18.21
Weight	**kg**	4.50	8.60	8.90	21.40	1.70	3.30	4.50	8.62

Table 1: Selection of dc-ac inverters

This data were then plotted using the power output as the abscissa and weight, volume, and cost as ordinates. The plots were grouped by brand, where an average linear approximation was conducted to define the constraints for a 1.5 kW ac-dc inverter. Overall, linear fitted curves allowed for R^2 values to range from 0.5 to 0.9969, higher than polynomial, exponential, and power curves. For each category (weight, volume, and cost), only the curves with similar slopes were examined in order to have consistency with extrapolation. From the linear fitted curves, extrapolation was done in order to attain weight, volume, and cost for each brand at 10 kW and

1.5 kW respectively (Table 2).

		Trace	Statpower	Tripp-Lite	Whistler
Output	**W**	10000	10000	10000	10000
Cost	**$**	5156.30	---	3702.61	3399.80
Volume	**L**	82.32	85.42	---	71.08
Weight	**kg**	67.17	47.12	---	32.34

Table 2: Calculated parameters for the 10 kW inverters by brand

The same procedure was followed for a 1.5 kW inverter, in order to maintain a cost relation between the data (Table 3).

		Trace	Statpower	Tripp-Lite	Whistler
Output	**W**	1500	1500	1500	1500
Cost	**$**	1968.80	---	605.21	524.25
Volume	**L**	45.77	16.57	---	10.73
Weight	**kg**	47.62	8.02	---	5.15

Table 3: Calculated parameters for the 1.5 kW inverters by brand

Having the averages for the 10 kW and 1.5 kW inverters with the requirements set forth by the DOE, a simple ratio was used to calculate the downscaled requirements for the 1.5 kW prototype dc-ac inverter.

		10 kW Average	1.5 kW Average	10 kW Prototype	1.5 kW Prototype
Cost	**$**	4086.23	1032.75	500.00	126.37
Volume	**L**	79.61	24.36	50.00	15.30
Weight	**kg**	48.88	20.26	32.00	13.27

Table 4: Design requirements for the 1.5 kW prototype inverter

Design was focused on a Whistler PP 1500 AC Power Inverter that was purchased. The reason the inverter was purchased was to allow the team to work on a physical model and for

testing of the designed heat sink and enclosure. The EE teams agreed to have this inverter as a benchmarking device for the ME team. The integrated circuit board was found to weigh 1.22 kg as measured by an OHAUS I-10 digital scale. The thermal management enclosure then, must not exceed 12.4 kg. Furthermore, conversations with representatives from Whistler and Trace, dc-ac inverter manufactures, informed us that a mere 15% of costs goes into thermal management. Axxion, a local computer case manufacturer also specified that 15% of costs is considered in their operations. With the given information, the amount allowed for this project was expected to be below $20.00 in thermal management. The assumed production rate for high volume is over 100,000 units per year. The thermal management design process began once the constraints were fully defined (Table 5).

Cost	$	$20.00
Weight	kg_f	12.4
Volume	L	15.3

Table 5: Thermal management/packaging constraints for a 1.5 kW prototype inverter

Design Options. The first option the group considered was the use of water to cool down the electronics outside of a heat sink (Figure 3). The water was to stand outside the heat sink inside cooper tubing collecting the waste heat coming out of the electronics from the heat sink. The idea was that after the water had reach a desired temperature below 100 ^{o}C, the water was to be released into a water heater, making the water heater work less. With this in mind, the overall energy efficiency of a home or small business could increase.

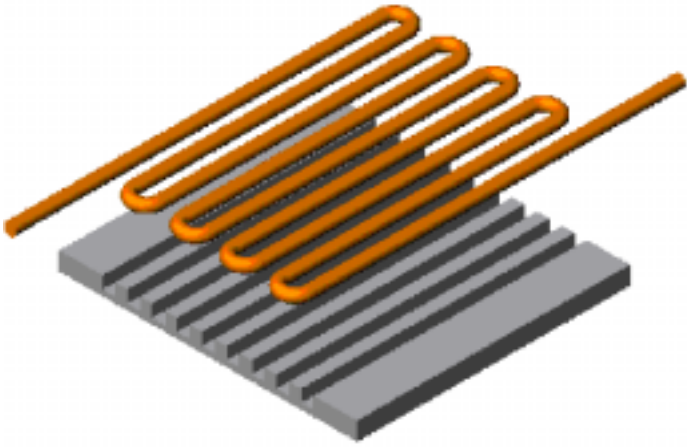


Figure 3: Sink and tubing arrangement for co-generation

However after analyzing our constraints, having a budget of no more than $20, this design was discarded since the required pressure regulator would cost more than this limit. The team then decided to keep the heat sink simple and to use only passive cooling to dissipate the amount of heat needed. Passive cooling via a heat sink was to be considered without the use of a tubing arrangement. Forced convection, or the use of a fan was put aside because a fan could be power consuming, noisy, and affect equipment reliability. Another option the team studied was the use of heat pipes (Figure 4).

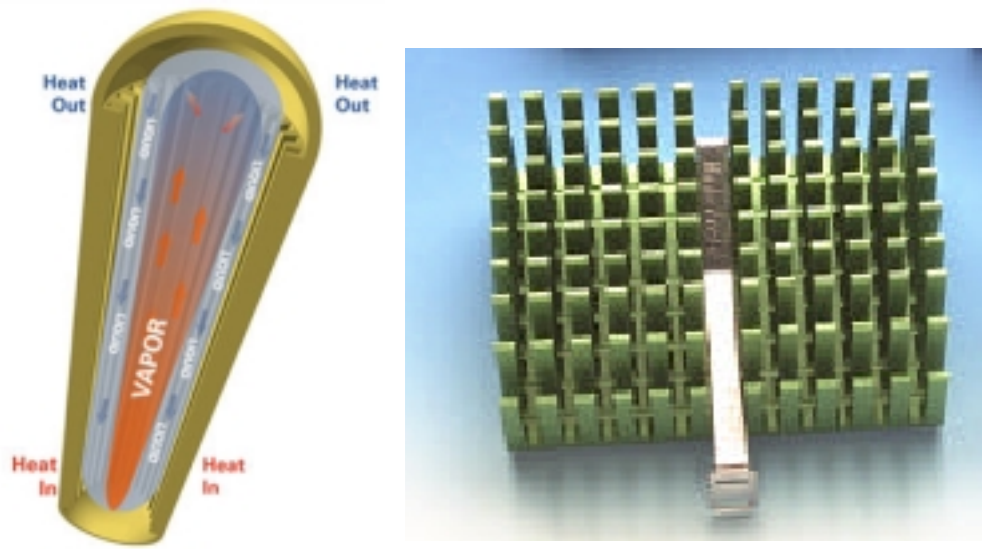


Figure 4: Heat pipe mechanical principles and heat pipe array

Our analysis indicated that a heat pipe configuration for an area of 145.2 cm^2 (expected area covered by one side of the Whistler inverter) would cost $160. In some applications like small components, or much higher power designs these devices are worth the expense, but in our case, they were not suitable.

We also considered thermoelectric coolers, which offer the potential to enhance the cooling of electronic module packages to reduce chip-operating temperatures or to allow higher module powers. Thermoelectric coolers also offer the advantages of being compact, quiet, and free of moving parts. Unfortunately, they are limited in the heat (125 W) that they can accommodate.

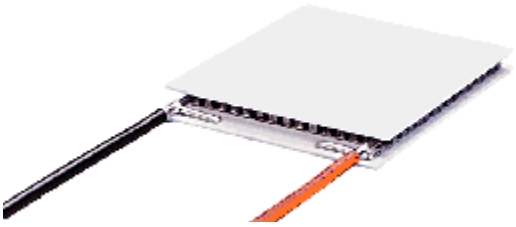


Figure 5: Thermoelectric generator

This device would not be practical for this project. To dissipate 175 W of heat, one would have to purchase two 100 W input thermoelectric generators. The cost per generator is about $40. This would exceed the $20 cost constraint for thermal management in this project.

Finally, we considered a simple metallic heat sink and enclosure. For an electronic enclosure or "box," the amount of heat that can be removed by natural convection will increase if the surface area of the "box" is increased. One method of increasing the surface area is by enlarging the conducting element. However, another method involves the addition of fins (Figure 6). This applies if the total surface area is constrained to a set volume for the overall enclosure. For this project, a volume of 15.3 L has already been established. Steinberg also states that fins will increase the size and weight of an electronic enclosure when compared to a "box" enclosure. Even so, it is possible to eliminate the use of a fan (non-passive system) to cool the electrical components in such a way. Since there are a large variety of such heat sinks available out on the market, further investigation and analysis was conducted on this passive-cooling option to satisfy the primary constraint, cost.

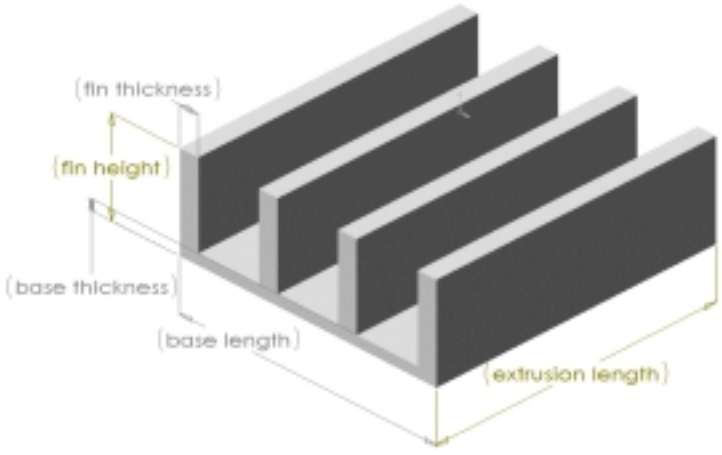


Figure 6: Typical heat sink design

Material Selection. A material selection process was conducted with regard to the heat sink material. A total of seven materials were considered. The group agreed upon the following 5 metals and 2 non-metals as prospective candidates (Table 6). Their raw cost and manufacturing cost were analyzed.

		aluminum 6061	copper	gold	silver	nickel (40% carbon)	Therma-tech	Konduit
thermal conductivity	**W/m-^{o}C**	156	390	298	417	10	8.7	11
density	**kg_m/m^3**	2700	8900	19320	10500	8169	1670	1750
raw cost/lb_f	**$/$kg_f$**	1.10	1.82	8,530.00	150.00	7.45	73.00	96.00

Table 6. Material selection candidates

The heat sink profile considered was a Wakefield 2044 profile (Figure 7). Such a profile with a length of 323.85 mm would allow the use of one heat sink to mount a row of power electronic devices.

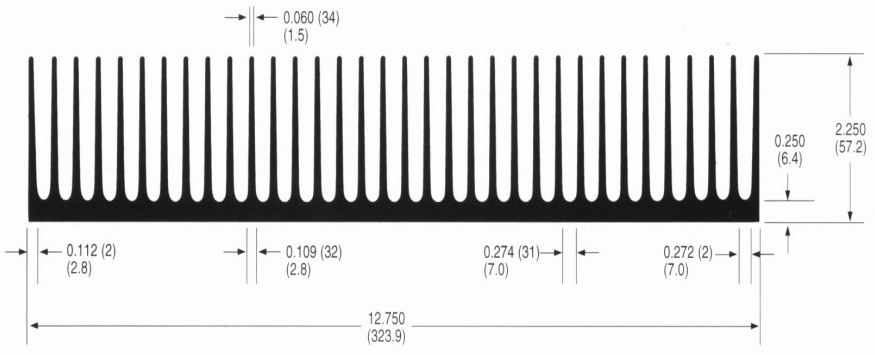


Figure 7: Fin design profile

Plots of dissipated power versus raw material cost and weight were generated to determine if the constraints were exceeded. The following plot shows the cost based on raw material needed to dissipate a certain amount of heat. For a Wakefield 2044 flat back heat sink, it is shown that all materials, except aluminum exceeded the $20.00 expenditure allowed for removing 175 W of heat. Gold and Silver were omitted from graph because of their high cost.

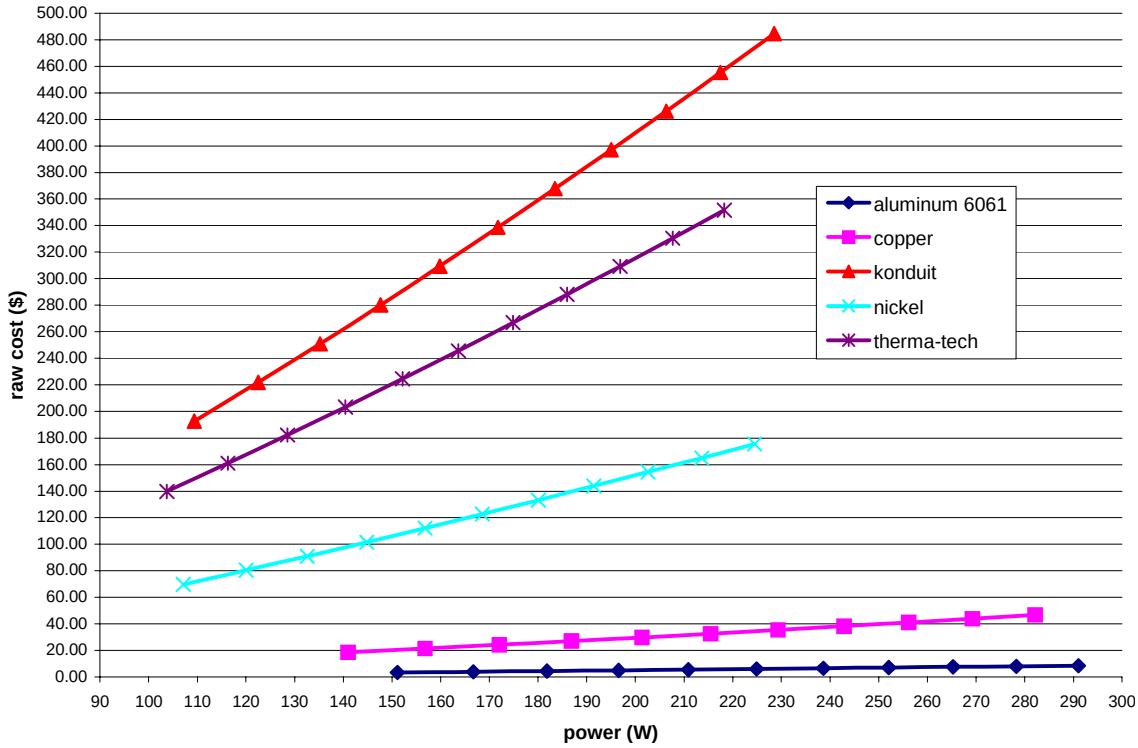


Figure 8: Raw material cost as a function of power dissipation

Furthermore, the weight limit of 12.4 kg_f for the thermal management enclosure was exceeded by silver, nickel and gold (Figure 9). As a result, only aluminum was found to be below the constraints of weight and cost for a power dissipation of 175 W as displayed in Figures 8 and 9.

Manufacturing Process. As given, the production rate for the overall renewable energy interface was set at 100,000+ components per year. Based on this information, and aluminum as the definite material, a material manufacturing process was chosen. The selection of a material places certain restrictions on the available manufacturing processes. Extrusion was the process chosen. For the Wakefield profile 2044 (Figure 7), the extrusion die cost is $1,000. This cost is acceptable when considering a production rate of 100,000+ heat sinks. At this production rate, Wakefield's quote on price was set at $24.95 per 305 mm of extruded aluminum for profile 2044. This is equivalent to $2.08 per 25 mm of extruded aluminum.

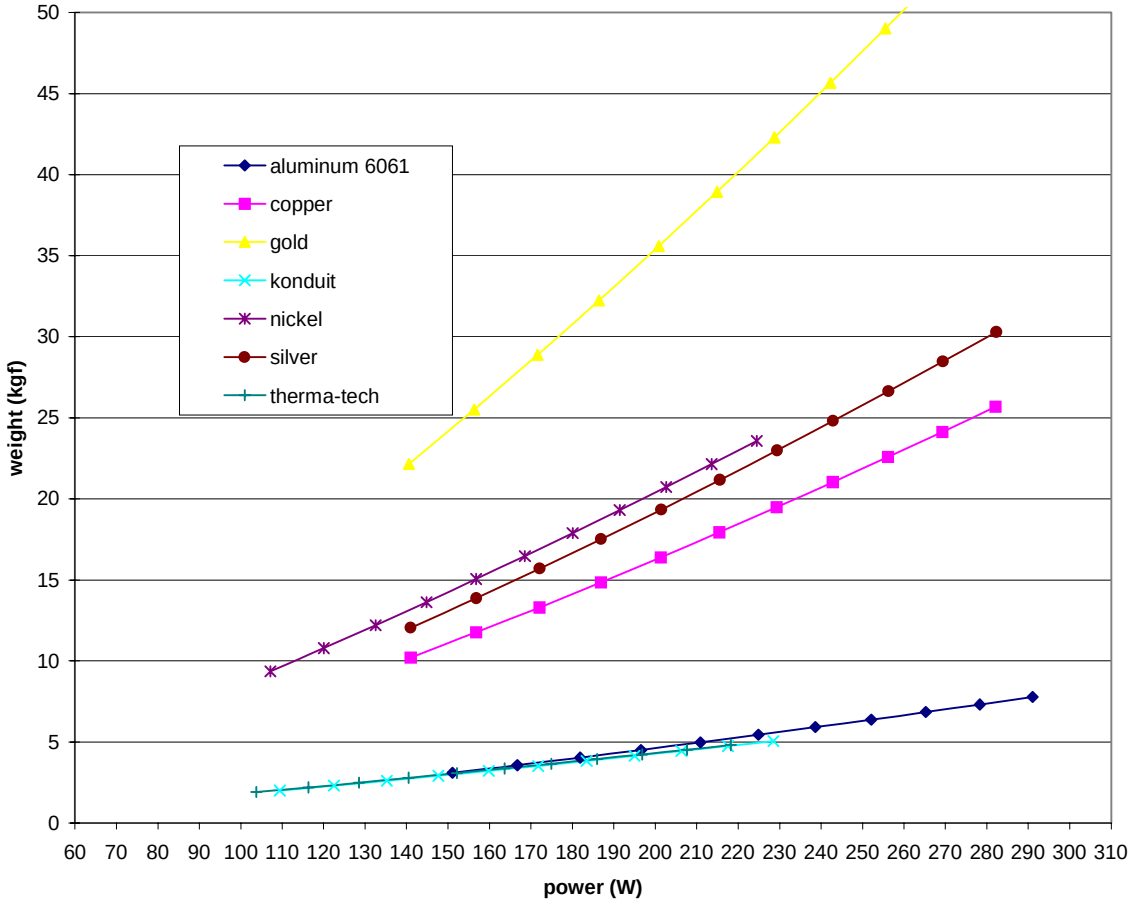


Figure 9: Weight as a function of power dissipation

Description of Final Design. The final design of aluminum-finned heat sinks, shown in Figure 10, were purchased and machined to an extrusion length of 102 mm. Side panels, top and bottom plates were machined to serve as the enclosure for the inverter circuit board. An all aluminum enclosure was the material the team decided on, since the fins were aluminum the incasing would performer better. Having same materials, the box would have the same properties, and the most important one would be in the case of thermal conductivity.

As one can see from the drawing, the box consists of six major parts. The two finned sides are 324mm long by 101.6 high by 57.15mm depth. These heat sinks are composed of 34, 1.524 thick fins and are mounted flush against the power electronic devices. The top is an aluminum slab 413 long, 190.5 wide and 6.35mm thick. The top aluminum slab was chosen 6.35mm or .25in thick to accommodate some electronics in it, this way the top of the transformers on the inverter were flush against the slab of aluminum, and some taller capacitors had clearance, helping to cool down the circuitry further more. The bottom plate maintained the same

dimensions, but instead of 6.35 mm thick, 1.60 mm thickness was used to be able to cut down expense and weight. The front and back pieces are made of strips of aluminum 276.5 long and 109.53mm high. These pieces are bent 90 degrees into an L shape at the 85.7mm mark lengthwise. On the front panel a display giving readings of the current and voltage in as well as the plugs for the outlets and on/off switch are located. On the back panel the inlet probes are marked as red for positive and black as negative.

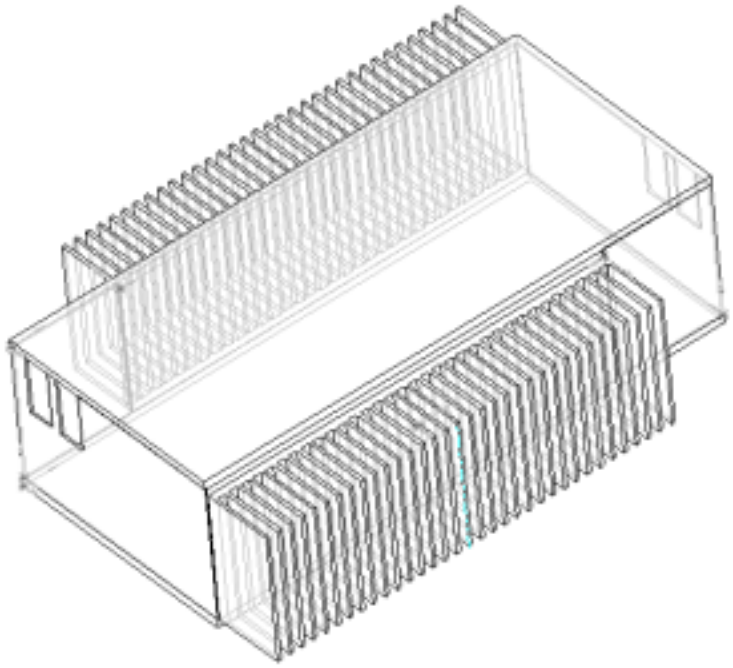


Figure 10: Final enclosure design

Meeting Constraints. The overall physical constraints for a 1.5 kW prototype inverter were met. The final weight of the enclosure was found to be 7.7 kg_f as measured by an Ohaus scale model I-10. The volume constraint was also met: the final design occupies a total volume of approximately 14.4 L. This is the volume for a box enclosure, meaning the volume the design would occupy if placed inside a box. This was in reference to the Future Energy Challenge design requirements when downscaled to a 1.5 kW inverter. For the case of weight, the final design was 42% lighter than that set by the constraints. The volume is almost a 1 L smaller and the amount spent for thermal management was 18.5% less. Final physical parameters are shown below, with a comparison of the constraints to that of the final product specifications.

	Constraints	Final Design
Weight (kg_f)	13.27	7.70
Volume (L)	15.30	14.40
Cost ($)	20.00	16.32

Table 8. Constraints versus final design

Testing was performed using two car batteries of approximately 13 V connected in parallel, the purchased Whistler inverter and two 500 W light bulbs. Our results indicated that the heat sink temperature did not exceed 50°C for this particular test

CONCLUSION/RESULTS

In conclusion, the project was completed with all modules being integrated and completed to approximately 90%. One hundred percent completion of the project was not accomplished because of several minor problems. Due to lack of time these problems were not able to be corrected before the semester was over.

For the Power Electronics portion of the project, the results we were able to obtain were very encouraging, including a measured efficiency of 91%. The remaining issue of major concern was the need for feedback to the common leg to regulate unbalanced loads. This issue was not addressed due to lack of time.

For the Thermal management and Packaging portion of the project, the weight, volume, cost and thermal management requirements were all satisfied.

APPENDIX

Constant Duty Cycle for Cuk Converter Program

```
/*
Electrical Ingenuity
Constant Duty Cycle for Cuk Converter
  This program uses Timer B match register 1 to create a PWM signal on PE7.
   The duty cycle can be adjusted by modifying the value at the dutycycle variable
*/

float maxcount = 1023;          // Timer B is a 10 bit counter
float dutycycle;                // 10% of maxcount < dutycycle < 90% of maxcount
shared unsigned int count;      // dutycycle converted to int
shared unsigned int flag;       // bit 0 is used to determine next state of output
int value,j;                            //counter
int values[1]={94.4};  //array values used to test program
void timerb_isr();

main(){


        dutycycle = 50;                 //modify duty cycle here
        count = 0;
        value=512;


        WrPortI(SPCR, &SPCRShadow, 0x84);       // initialize port A as output
        WrPortI(PADR, &PADRShadow, 0xFF);               // initialize LEDs as off

        flag = 0;

        WrPortI(PEFR, &PEFRShadow, 0x00); // set port E in standard mode
        WrPortI(PEDDR, &PEDDRShadow, 0x80);     // set port E bit 7 as output
        WrPortI(PECR, &PECRShadow, 0x20);   // set upper mibble of port E to clock
                                                out on B1 match

        SetVectIntern (0xB,timerb_isr);
        WrPortI(TBCR, &TBCRShadow, 0x01);       // clock timer B with (perclk/2) and
                                                set interrupt level to 1

        count = (int)((dutycycle * maxcount)/100);
        WrPortI(TBL1R, NULL, count&0xFF );// set initial match
        WrPortI(TBM1R, NULL, count>>2);
        WrPortI(PEB7R, NULL, 0x80);             // set initial output value

        WrPortI(TBCSR, &TBCSRShadow, 0x03);     // enable timer B and B1 match
                                                interrupts

        while (1){
```

```
                costate
                {
                dutycycle = (float)value;
                count = (int)((dutycycle * maxcount)/100);
                runwatch();                             // update watch expression
                }
        }
}


#asm
timerb_isr::
; For this ISR to function properly you MUST update (TBL1R) after (TBM1R)!!!

                        push    af                              ; save registers
                        push    hl

        ioi     ld              a, (TBCSR)      ; clear the interrupt

                        ld              hl, (flag)          ; get flag value
                        inc     hl                              ; increment
                        ld              (flag), hl          ; save for next time

                        ld              a, 01h              ; mask off
                        and     l                               ;                       all but bit 0
                        jr              z, match_xxxx ; jump if need to turn output on next time

match_0000:
                        ld              a, 00h              ; set up to
        ioi     ld              (PEDR), a       ;           turn output off at next interrupt

                        ld              a, (count+1) ; put high byte value into bits 6 & 7
                        add     a, a                    ; these instructions shift the value
                        add     a, a                    ;                       in register a by 6 bits
                        add     a, a
                        add     a, a
                        add     a, a
                        add     a, a
  ioi   ld              (TBM1R), a

                        ld              a, (count)          ; get low byte value
        ioi     ld              (TBL1R), a      ; writing to low byte enables next interrupt

                        jr              done

match_xxxx:
                        ld              a, 80h              ; set up to
        ioi     ld      (PEDR), a       ;           turn output on at next interrupt

                        ld              a, 00h
```

```
  ioi     ld      (TBM1R), a
  ioi     ld              (TBL1R), a      ; writing to low byte enables next interrupt

done:
                        pop     hl                      ; restore registers
                        pop     af

                        ipres                           ; restore interrupts
                        ret                             ; return
#endasm
```

Manually Adjusted Duty Cycle for Cuk Converter Program

```
/* Electrical Ingenuity
PWM.c

The program uses Timer B match register 1 to create a PWM signal on PE7.  The
duty cycle can be changed dynamically by pressing S1 to increase it or by
pressing S2 to decrease it.  The output frequency is fixed at a rate determined
by the processor speed - a Jackrabbit with a CPU clock of 14.8 MHz will create
a frequency of approximately 7.246 KHz.  As the duty cycle approaches the limits
of 10% and 90% the output has some jitter due to the ISR processing time - once
in a while an interrupt gets skipped. This program was written by the jackrabbit
customer support and was slightly modified to fit our purposes
*/

float maxcount = 1023;                  // Timer B is a 10 bit counter
float dutycycle;                                // 10% of maxcount < dutycycle < 90% of maxcount
shared unsigned int count;      // dutycycle converted to int
shared unsigned int flag;               // bit 0 is used to determine next state of output

void timerb_isr();

main(){

        dutycycle = 50;
        count = 0;

        WrPortI(SPCR, &SPCRShadow, 0x84);           // initialize port A as output
        WrPortI(PADR, &PADRShadow, 0xFF);               // initialize LEDs as off

        flag = 0;

        WrPortI(PEFR, &PEFRShadow, 0x00); // set port E in standard mode
        WrPortI(PEDDR, &PEDDRShadow, 0x80);     // set port E bit 7 as output
        WrPortI(PECR, &PECRShadow, 0x20);// set upper mibble of port E to clock out on B1 match

        SetVectIntern (0xB,timerb_isr);
```

```
        WrPortI(TBCR, &TBCRShadow, 0x01);          // clock timer B with (perclk/2) and set interrupt
level to 1

        count = (int)((dutycycle * maxcount)/100);
        WrPortI(TBL1R, NULL, count&0xFF );// set initial match
        WrPortI(TBM1R, NULL, count>>2);
        WrPortI(PEB7R, NULL, 0x80);                // set initial output value

        WrPortI(TBCSR, &TBCSRShadow, 0x03);        // enable timer B and B1 match interrupts

        while (1){

                costate             // debounce switch S1
                {
                        if (BitRdPortI(PBDR, 2))
                        {
                                BitWrPortI(PADR, &PADRShadow, 1, 0); // turn off LED DS1
                                abort;                          // if button not down, skip out of
costatement
                        }
                        waitfor(DelayMs(50));                                       // debounce
switch S1
                        if (BitRdPortI(PBDR, 2))
                        {
                                BitWrPortI(PADR, &PADRShadow, 1, 0);        // turn off LED DS1
                                abort;                          // if button not still down, skip out of
costatement
                        }
                        BitWrPortI(PADR, &PADRShadow, 0, 0);        // turn on LED DS1
                        if (dutycycle < 90) dutycycle = dutycycle + 1;    // increase dutycycle
                        waitfor(DelayMs(200));
                }

                costate             // debounce switch S2
                {
                        if (BitRdPortI(PBDR, 3))
                        {
                                BitWrPortI(PADR, &PADRShadow, 1, 1);  // turn off LED DS2
                                abort;
                        }
                        waitfor(DelayMs(50));                                       // debounce
switch S2
                        if (BitRdPortI(PBDR, 3))
                        {
                                (BitWrPortI(PADR, &PADRShadow, 1, 1)); // turn off LED DS2
                                abort;              // if button is not still down, skip out of costatement
                        }
                        BitWrPortI(PADR, &PADRShadow, 0, 1);        // turn on LED DS2
                        if (dutycycle > 10) dutycycle--;                    // decrease dutycycle
                        waitfor(DelayMs(200));
```

```
                }
                count = (int)((dutycycle * maxcount)/100);
                runwatch();                             // update watch expression
        }
}


#asm
timerb_isr::
; For this ISR to function properly you MUST update (TBL1R) after (TBM1R)!!!

                        push    af                              ; save registers
                        push    hl

        ioi     ld              a, (TBCSR)      ; clear the interrupt

                        ld              hl, (flag)          ; get flag value
                        inc     hl                              ; increment
                        ld              (flag), hl          ; save for next time

                        ld              a, 01h              ; mask off
                        and     l                               ;                       all but bit 0
                        jr              z, match_xxxx ; jump if need to turn output on next time

match_0000:
                        ld              a, 00h              ; set up to
        ioi     ld              (PEDR), a       ;           turn output off at next interrupt

                        ld              a, (count+1) ; put high byte value into bits 6 & 7
                        add     a, a                    ; these instructions shift the value
                        add     a, a                    ;                       in register a by 6 bits
                        add     a, a
                        add     a, a
                        add     a, a
                        add     a, a
  ioi   ld              (TBM1R), a

                        ld              a, (count)          ; get low byte value
        ioi     ld              (TBL1R), a      ; writing to low byte enables next interrupt

                        jr              done

match_xxxx:
                        ld              a, 80h              ; set up to
        ioi     ld      (PEDR), a       ;           turn output on at next interrupt

                        ld              a, 00h
  ioi   ld      (TBM1R), a
  ioi   ld              (TBL1R), a      ; writing to low byte enables next interrupt

done:
```

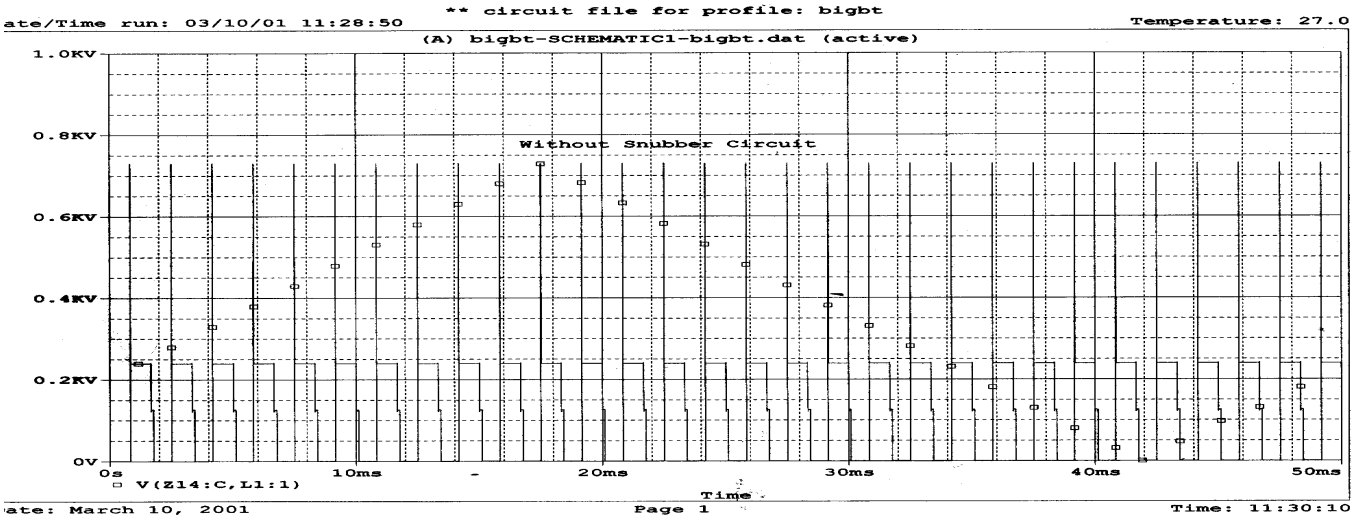
```
                pop     hl                      ; restore registers
                pop     af

                ipres                           ; restore interrupts
                ret                             ; return
#endasm
```

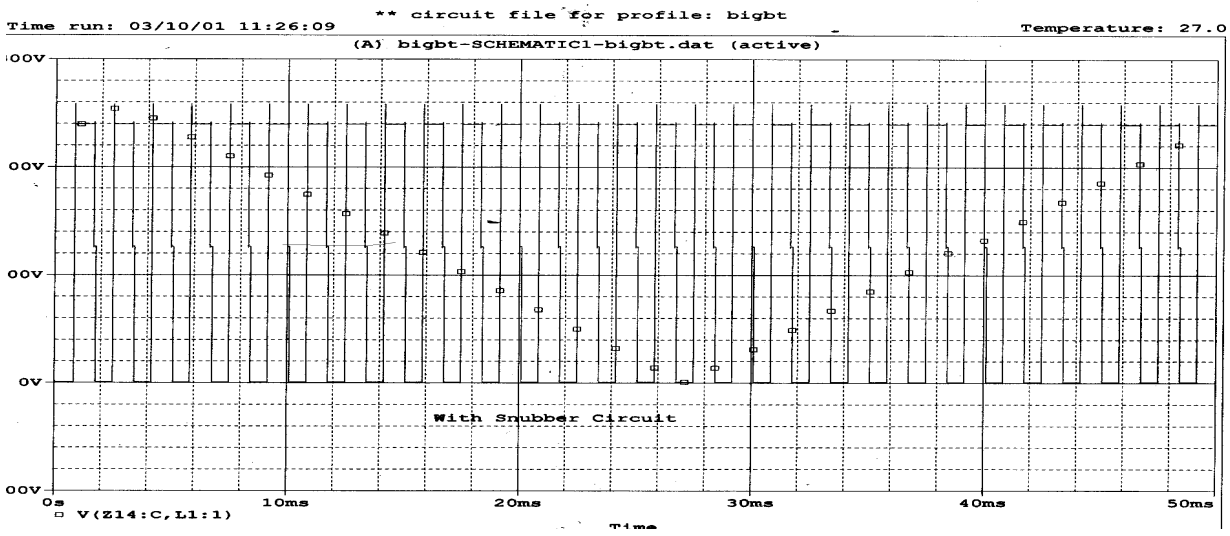
Vcontrol Dependant Duty Cycle for Cuk Converter Program

```
/*
Electrical Ingenuity
The program uses Timer B match register 1 to create a PWM signal on PE7.
The duty cycle depends on the value of Vcontrol which is being read from the analog input 0
The output frequency is fixed at a rate determined
by the processor speed - a Jackrabbit with a CPU clock of 14.8 MHz will
create a frequency of approximately 7.246 KHz.

*/

float maxcount = 1023;          // Timer B is a 10 bit counter
float dutycycle;                // 10% of maxcount < dutycycle < 90% of maxcount
shared unsigned int count;      // dutycycle converted to int
shared unsigned int flag;       // bit 0 is used to determine next state of output


int values1[10];
void timerb_isr();
int i, value1;
int count1, count2;

main(){

        dutycycle = 50;
        count = 0;
        count1 = count2 = 0;
        jrioInit();

        WrPortI(SPCR, &SPCRShadow, 0x84);           // initialize port A as output
        WrPortI(PADR, &PADRShadow, 0xFF);                   // initialize LEDs as off

        flag = 0;

        WrPortI(PEFR, &PEFRShadow, 0x00); // set port E in standard mode
        WrPortI(PEDDR, &PEDDRShadow, 0x80);     // set port E bit 7 as output
        WrPortI(PECR, &PECRShadow, 0x20);// set upper mibble of port E to clock out on B1 match
```

```
	SetVectIntern (0xB,timerb_isr);

	WrPortI(TBCR, &TBCRShadow, 0x01);		// clock timer B with (perclk/2) and set interrupt
level to 1

	count = (int)((dutycycle * maxcount)/100);
	WrPortI(TBL1R, NULL, count&0xFF );// set initial match
	WrPortI(TBM1R, NULL, count>>2);
	WrPortI(PEB7R, NULL, 0x80);			// set initial output value

	WrPortI(TBCSR, &TBCSRShadow, 0x03);	// enable timer B and B1 match interrupts

	while (1){
	while (count1<10) {
		costate
		{
			for (i=0; i<10; i++){
			waitfordone {
				cof_anaIn(0, &value1);		// execute one step
			}
			values1[i%2]=value1;
			DelaySec(3000);
		//values to be read are from 0 to 1024 (0 to 2.5 Volts DC)

			//convert value to corresponding duty clycle

		dutycycle =(float)((value1)*(0.16666));
		count = (int)((dutycycle * maxcount)/100);
		runwatch();				// update watch expression

}
	}
	}
}
}

#asm
timerb_isr::
; For this ISR to function properly you MUST update (TBL1R) after (TBM1R)!!!

			push	af				; save registers
			push	hl

	ioi	ld		a, (TBCSR)	; clear the interrupt

			ld		hl, (flag)	; get flag value
			inc	hl				; increment
			ld		(flag), hl	; save for next time

			ld		a, 01h		; mask off
```

```
                and     l                               ;               all but bit 0
                jr              z, match_xxxx ; jump if need to turn output on next time

match_0000:
                ld              a, 00h          ; set up to
        ioi     ld              (PEDR), a       ;       turn output off at next interrupt

                ld              a, (count+1) ; put high byte value into bits 6 & 7
                add     a, a                    ; these instructions shift the value
                add     a, a                    ;               in register a by 6 bits
                add     a, a
                add     a, a
                add     a, a
                add     a, a
  ioi   ld              (TBM1R), a

                ld              a, (count)      ; get low byte value
        ioi     ld              (TBL1R), a      ; writing to low byte enables next interrupt

                jr              done

match_xxxx:
                ld              a, 80h          ; set up to
        ioi     ld      (PEDR), a       ;       turn output on at next interrupt

                ld              a, 00h
  ioi   ld      (TBM1R), a
  ioi   ld              (TBL1R), a      ; writing to low byte enables next interrupt

done:
                pop     hl                              ; restore registers
                pop     af

                ipres                                   ; restore interrupts
                ret                                     ; return
#endasm
```


** circuit file for profile: bigbt
ate/Time run: 03/10/01 11:28:50
Temperature: 27.0
(A) bigbt-SCHEMATIC1-bigbt.dat (active)
1.0KV
0.8KV
0.6KV
0.4KV
0.2KV
0V
Without Snubber Circuit
0s
10ms
20ms
30ms
40ms
50ms
V(Z14:C,L1:1)
Time
ate: March 10, 2001
Page 1
Time: 11:30:10


** circuit file for profile: bigbt
Time run: 03/10/01 11:26:09
Temperature: 27.0
(A) bigbt-SCHEMATIC1-bigbt.dat (active)
00V
00V
00V
0V
00V
With Snubber Circuit
0s
10ms
20ms
30ms
40ms
50ms
V(Z14:C,L1:1)
Time

You have reached the end of this final report.

Use the button below to return to the Main Document

Return to Main Document