Application Note

Universal Power and Voltage Control Using the Z8E001

AN003501-Z8X0500

ZiLOG Worldwide Headquarters • 910 E. Hamilton Avenue • Campbell, CA 95008
Telephone: 408.558.8500 • Fax: 408.558.8300 • www.ZiLOG.com

This publication is subject to replacement by a later edition. To determine whether a later edition exists, or to request copies of publications, contact:

ZiLOG Worldwide Headquarters
910 E. Hamilton Avenue
Campbell, CA 95008
Telephone: 408.558.8500
Fax: 408.558.8300
www.ZiLOG.com

Information Integrity

The information contained within this document has been verified according to the general principles of electrical and mechanical engineering. Any applicable source code illustrated in the document was either written by an authorized ZiLOG employee or licensed consultant. Permission to use these codes in any form, besides the intended application, must be approved through a license agreement between both parties. ZiLOG will not be responsible for any code(s) used beyond the intended application. Contact the local ZiLOG Sales Office to obtain necessary license agreements.

Document Disclaimer

Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZiLOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZiLOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. Except with the express written approval ZiLOG, use of information, devices, or technology as critical components of life support systems is not authorized. No licenses or other rights are conveyed, implicitly or otherwise, by this document under any intellectual property rights.

Table of Contents

Acknowledgements

Project Lead Engineer

Denny Hopp

Application and Support Engineers

Jon Veres, Mark Shaw, and Randy Rolen

System and Code Development

Denny Hopp

Universal Power and Voltage Control Using the Z8E001

Abstract

General Overview

The ZiLOG Z8Plus™ Microcontroller family includes many integrated features that greatly simplify power and voltage control system designs. This family of microcontrollers offers the on-chip peripherals that allow power control systems to be designed cost-effectively with as few external components as possible. The Z8E001 used in this design contains many on-board features that provide full functionality in a small microcontroller. These features include three timer/counters, six internal/external interrupts with selectable edge triggering, 1-µs instruction execution time using a 10-MHz crystal, analog comparator, push-pull or open-drain output bit programmability, watch-dog timer, STOP mode recovery, low-power sleep mode, and others.

This Application Note describes a design that controls the energy provided to a Xenon gas flashtube. To control the light output to the flashtube, the charge capacitor voltage must be closely controlled and monitored. The Z8E001 precisely handles all aspects of this control. This design also illustrates how the Z8E001 is used to produce output voltages in excess of 200 volts with a supply voltage as low as 14 volts.

With an overhead of only two system clock cycles required for interrupt vectoring, this design demonstrates the precise and extremely fast power control realized with the Z8E001. Charge cycles as short as 35 µs must be continually modified, and this microcontroller provides this control with ease. In addition to efficiently controlling these short charge pump cycles, the Z8E001 controls additional system timing and input/output functions simultaneously.

The potential applications for this type of precise and inexpensive voltage and power control are numerous. This type of control can be used in applications such as motor control, AC/DC and DC/DC converters, circuits requiring high-voltage generation from battery circuits, *smart* battery chargers for various battery chemistries, power supplies, and any circuit requiring a charge pump for energy storage. The Z8E001 provides the control for these designs, because the overall system design can be completed at a fraction of the cost of other systems using dedicated power and voltage control integrated circuits.

Finally, this application demonstrates the superior noise immunity of the Z8Plus microcontrollers. Because the charge pump generates over 200 volts and the trig-

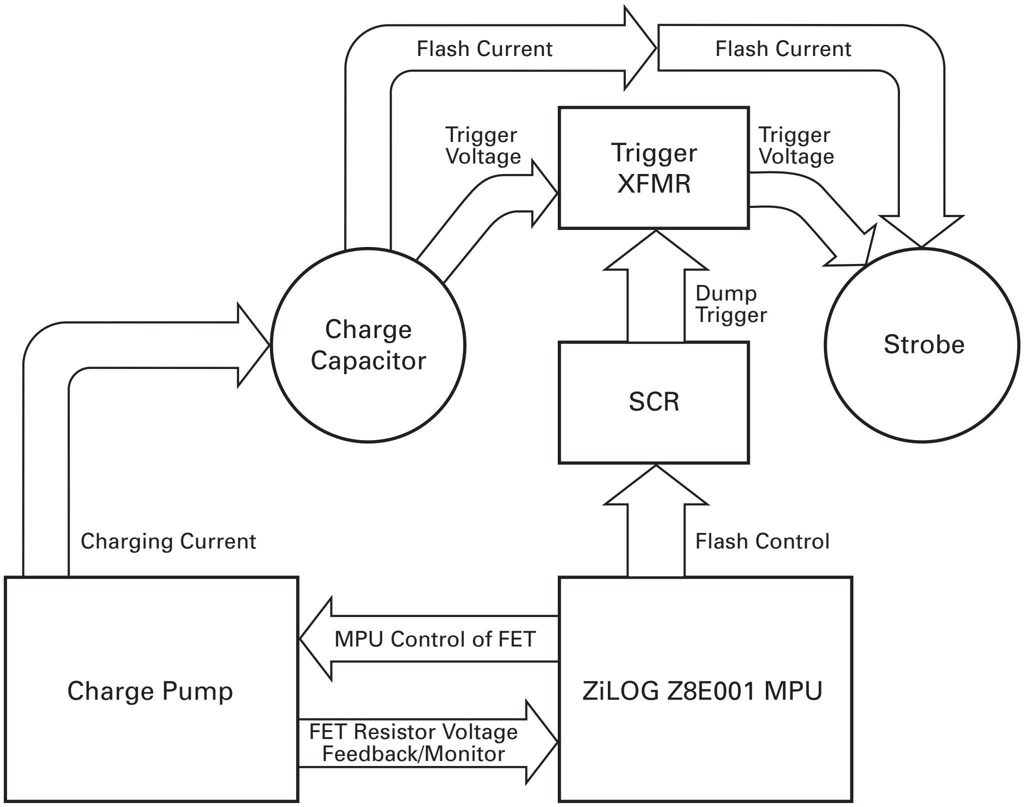
ger transformer generates thousands of volts, there is the potential for many noise-induced voltage brownout/reset conditions. Also, note that the microcontroller is switching the FET at frequencies up to 33 kHz. The ZiLOG Z8E001 runs without error in this extremely noisy environment.

Discussion

Theory of Operation

Figure 1 illustrates the overall system operation.

Figure 1. Overall System Operation



The microprocessor turns on the charge pump and monitors the charge current. When this current reaches the preset level (determined by monitoring the voltage on a feedback resistor), the microprocessor measures the time required to reach the specific charge current value, then determines how long to keep the charge pump off before starting the next charge cycle. When the charge pump is off, the charge pump current flows into the capacitor and builds up the capacitor voltage. This cycle is repeated continually for the entire operation of the circuit. The SCR is turned on once every second, and the capacitor *dumps* through the trigger transformer and strobe, causing the strobe flash. This transfer of energy from the capacitor to the strobe is caused by the *dump trigger* signal on the gate of the SCR, causing a high-voltage spike on the external strobe wire. This energy transfer is discussed in detail later.

Compare the block diagram in Figure 1 to the circuit schematic diagram in the Appendix. The circuit is comprised of capacitor C1, inductor L1 and FET Q2. When the FET is on, the inductor charges from the supply voltage because the FET allows the current to flow through the inductor. When the FET is turned off after the inductor is charged, the inductor charge is transferred to capacitor C1. Diode D1 allows current to flow only from the inductor and blocks the capacitor charge current from flowing back through the FET when the FET is turned on.

This *charge pump* operation is analogous to blowing up a balloon. Air is blown into a balloon in short bursts, and the balloon neck is pinched off between each burst of air. When the balloon is filled, the final air pressure in the balloon is equal to the sum of the individual air bursts put into the balloon. The charge pump circuit works in the same way. When the inductor is charged to a set point, the FET is turned off, and the inductor energy is transferred to the capacitor. After the inductor transfers its energy to the capacitor, the FET is turned back on to allow the inductor to again build up energy, starting the process over again. The diode acts as the *pinch-off* mechanism to keep the capacitor charge from flowing back into the FET when the inductor is charging. Because the SCR is held off during these repeated short-charging cycles, the current can only flow into the capacitor when the FET is off.

⚠ **Warning:** The capacitor can be charged to voltages in excess of 200 volts. Also, the strobe trigger transformer can generate thousands of volts on the external strobe wire wrapped around the flash tube. Under no circumstances should any of the components be handled during operation. Exercise extreme care to ensure that all components are fully discharged before handling any circuit component.

The following statements discuss the importance of controlling each of the charge cycles.

$$V = L\ di/dt \qquad \text{(equ 1.1)}$$

Integrate both sides of the equation and the current is:

$$i = 1/L(V\ dt \qquad \text{(equ 1.2)}$$

Because P = VI, then:

$$P = Li\ di/dt \qquad \text{(equ 1.3)}$$

To evaluate the energy stored in the inductor and transferred to the capacitor, use the fact that power is the time rate of expending energy, which can be expressed as:

$$P = dw/dt = Li\ di/dt \qquad \text{(equ 1.4)}$$

Now multiply both sides of equation 1.4 by the time differential, resulting in the equation:

$$dw = Li\ di \qquad \text{(equ 1.5)}$$

The integration of both sides of equation 1.5 results in:

$$(dx = L\ (y\ dy \qquad \text{(equ 1.6)}$$

where the left-hand side of the equation is solved for $0 \le x \le w$, and the right-hand side is solved for $0 \le y \le i$. This integration solution results in:

$$w = \Omega\ Li2 \qquad \text{(equ 1.7)}$$

Because the current is proportional to the integral of the voltage (equ 1.2), the energy is proportional to the squared integral of the voltage. The rate of energy storage is not a linear function, so the microcontroller must be able to change the charging rate in a non-linear manner for the entire range of the specified operating voltage. For this design, the specified operating voltage is 14V to 24V, so the microcontroller must be able to control the charging current so that capacitor C1 always holds the same amount of energy just prior to flashing the strobe. The software is set up so that the energy storage rate is a linear function over the charging period between strobe flashes.

The 8-bit microcontroller performs this complex math easily by using lookup tables. The ZiLOG family of microcontrollers enables lookup tables to be implemented easily and with great flexibility. The software is discussed later, but it is important to note that this easy implementation of lookup tables allows any mathematical equation, no matter how complex, to be approximated (over a defined range) to whatever precision is required for the proper operation of the power control circuit.

Look at how the lookup tables are used to control the energy storage rate of the capacitor. When the FET is first turned on, the inductor current graph is almost linear (after a few time constants, the current levels off to its steady-state value). In this design, the inductor charge is stopped before the current becomes non-linear. Even though the inductor current is linear, the energy function is not linear, and this energy storage is of primary interest. As the current increases, the voltage across resistor R5 also increases, because all of the inductor charge current is passing through this resistor. The voltage across this resistor is connected to the microcontroller port B comparator input. When this voltage exceeds the reference voltage supplied on the port B comparator reference input (via voltage divider R9, R10), an interrupt is generated in the microcontroller. The reference voltage is

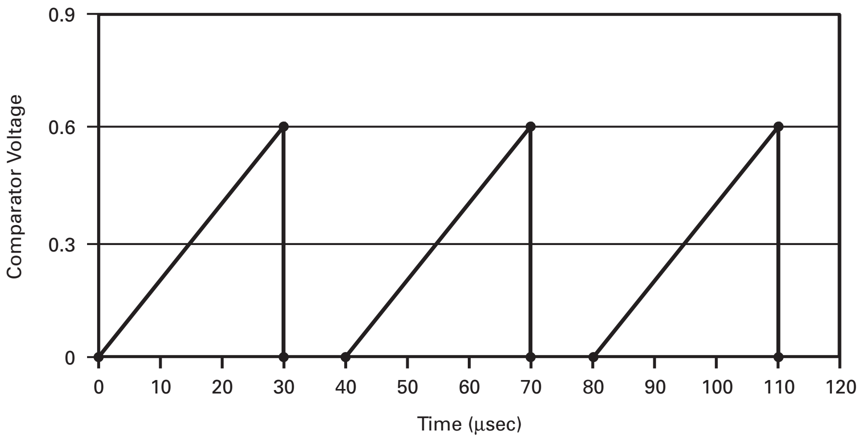
$$V_{REF} = (5V)(R9 \div (R9 + R10)) = 0.61V \qquad \text{(equ 1.8)}$$

Because the feedback resistor R5 is 3.6 Ohms, the microcontroller receives an interrupt when the inductor current reaches

$$I = 0.61V \div 3.6\ \text{Ohms} = 169.44\text{mA} \qquad \text{(equ 1.9)}$$

Figure 2 illustrates the inductor charge cycles. When the FET is turned on, the microcontroller measures the amount of time between the FET turn-on and the reception of the interrupt. When the interrupt is received, a lookup table value is retrieved based on the counter/timer value, and this table value specifies the amount of time that the microcontroller keeps the FET off before turning it back on. This process is repeated for every inductor charge cycle. As an example, look at the first charge cycle. At time 0, the FET is turned on. The FET starts the timer and waits for the interrupt. When the interrupt is received, the Z8 turns the FET off and reads the timer value. In this cycle, the value corresponds to 30 µs because the FET was on for 30 µs before the interrupt was received. The microcontroller retrieves a lookup table value that corresponds to 10 µs (the FET off time). The timer is reloaded to time 10 µs (interval between 30 µs and 40 µs), and when this hold-off period expires, the FET is turned back on and the cycle repeats itself. If the hold-off duration is increased, the FET does not exhibit as many charging cycles per time period. As a result, the energy storage in the capacitor is decreased. If the FET off-time periods are decreased, the capacitor charges to a higher voltage.

Figure 2. Inductor Charge Cycles



The lookup table values are derived either by setting up a spreadsheet to calculate the stored energy based on the area under the sawtooth charge curve or by a trial and error approach. This design uses the trial and error approach. Arbitrary values are assigned in the lookup table, the code is executed, and the capacitor maximum voltage is measured for the voltages in the 14V to 24V range. An oscilloscope measures the sawtooth on-period, and a chart is created to plot the sawtooth period vs. the maximum capacitor C1 voltage for the input voltage (in increments of 2V). If the voltage is too high for a certain charge period, then the lookup table value is increased. If the capacitor voltage is too low, then the off-time value in the lookup table is reduced. Because the efficiencies of capacitors vary greatly, the trial-and-error approach to loading the lookup table is favored. Therefore, the energy calculations must assume a perfect capacitor.

Discussion of the remaining circuits follows. The circuit comprised of Q1, R1, R2, and R3 provides the power-on-reset and brownout voltage reset functions. Resistors R1 and R2 provide the voltage divider network to control the transistor. When the transistor is off, the reset line is Low, and the microcontroller remains in the reset state. When the transistor is on, the reset line is connected to V_{CC} through the transistor, and the microcontroller is no longer in the reset state. At power-up, the transistor must be turned on after the microcontroller reaches the minimum operating voltage (3 volts for the Z8E001). To turn transistor Q1 on, the VBE voltage must be approximately 0.7 volts or higher. Because the VBE is exactly the same as VR1 (the two voltages are in parallel), then the supply voltage at the time of Q1 turn-on is:

$$VR1 = ((V_{CC})(R1))/(R1+R2) \qquad \text{(equ 1.10)}$$

Solving for V_{CC} (knowing that VR1 = 0.7V)

$$V_{CC} = ((0.7)(122K)) \div 22K = 3.88V \qquad \text{(equ 1.11)}$$

Therefore, the reset signal does not go High until the supply voltage reaches 3.88V. At power-up, the reset line is held Low, well past the minimum operating voltage of the Z8E001, and during voltage brownouts, the reset line goes Low (and reset the micro) long before the voltage drops below the minimum operating voltage. Different resistor values can be used if the reset voltage must be closer to the actual minimum operating voltage of the microcontroller, but for this application, these values work well. A new Z8Plus MCU family member, the Z8E003, eliminates the need for this external reset circuitry, because the voltage brownout and power-on-reset circuits are present on-chip.

The last circuit to discuss in this application is the strobe firing circuit. At one-second intervals, the Z8E001 outputs a High signal to the gate of the SCR, causing it to fire and discharge the capacitor energy through the strobe. Here is the sequence of events for the strobe firing:

1. Capacitor C1 charges to about 200 volts.
2. Capacitor C5 also charges to about the same voltage as C1 in this same interval.
3. The Z8E001 outputs a High signal to the SCR, causing it to conduct.
4. Capacitor C5 discharges quickly through the SCR, causing a high dv/dt on the primary of the trigger transformer.
5. The fast dv/dt pulse on the transformer primary causes an extremely high voltage on the transformer secondary which is connected to the external wire on the strobe tube, causing a high voltage pulse (3000+ volts) around the outside of the flash tube.
6. The pulse produces an ion path in the Xenon gas, greatly lowering the resistance between the electrodes and the flash tube.
7. When this resistance is lowered, C1 discharges its energy across the tube.
8. The electron flow caused by the capacitor discharge excites the electrons in the Xenon gas molecules.
9. The electrons return to a ground state, releasing the energy in the form of light.

Results of Operation

Using the Z8E001 program source code listed in the Technical Support section, the circuit (as shown in the schematic in the Appendix) was tested. When power is applied to the circuit, the strobe flashes once every second. The capacitor charges to approximately 190 volts just before the strobe is fired.

The source code is written so that performance modifications are made easily. The two main modifications made to the existing design are 1) increase or decrease the flash rate and 2) increase or decrease the flash intensity. The flash rate is easier to modify because only the constant value *timer_passes* must be changed. With a 10 MHz crystal, the frequency at timer23 is 10 MHz ÷ 8 = 1.25 MHz. The timer_passes register is decremented once for each timer interrupt, and when the register reaches zero, the strobe is fired. Therefore, with the current value of 25 in the timer_passes register and 50000 in the timer register, the strobe fires once per second (timer interrupts 1.25 MHz ÷ 50 KB = 25 interrupts per second, and it takes 25 passes before the timer_passes register decrements to 0, resulting in one fire every second). If the flash rate is changed, the light intensity changes also, because the capacitor charges to a higher voltage for extended flash rates and charge to a lower voltage for shorter flash rates. To maintain the same light output intensity that is set up in the current code, the lookup tables must be modified as described previously.

The second modification is to change the light output intensity. For increased light output, the lookup table values must be reduced. Conversely, for decreased light output, the lookup tables must be increased (these values relate to the off-time of the FET).

The light intensity is increased or decreased by changing the values of the comparator resistor R5 and/or the values of the voltage resistors R9 and R10. If R10 is increased to increase the reference voltage, then the comparator interrupts on higher voltages on R5, causing a higher charge rate on C1. If comparator resistor R5 is changed, the light intensity is also affected. However, a change in this resistor is not as predictable, because it does not only cause a change in the comparator interrupt voltage, but also a change in the charge rate of the inductor. For example, if R5 is raised, the comparator interrupts at a higher voltage. This change can cause the inductor to charge to a higher voltage, increasing the energy transfer to the capacitor C1, but an increase in this resistance also causes a lower charge current through the inductor, reducing the total energy per charge cycle. Therefore, changing the reference voltage divider produces the most consistent results because the inductor current is not changed. Resistor R5 should be changed only if a different inductor value is chosen to allow an efficient charge current through the inductor. In this circuit, values of R5 between 2 Ohms and 10 Ohms work well, with the lower resistance values providing better results for input voltages up to 40 volts. If higher supply voltages are used, then higher resistance values should be used (higher supply voltages result in higher di/dt values, so the resistor R5 has to be increased to lower the current change rates).

⚠ **Warning:** This circuit produces dangerously high voltages on many of the components. Exercise extreme care when operating the circuit or modifying the circuit components.

Summary

Reaffirmation of Results

Although this application of firing a strobe appears simple, the math describing the operation of the circuit (and its operating range) is complex. The supply voltage in this circuit varies from 14 to 24 volts, and the light output intensity of the strobe must remain constant throughout this entire range. The ZiLOG Z8E001 allows power control to be taken to new heights, because it controls the output power without forcing the inductor to store and transfer energy inefficiently. The Z8E001 provides for efficient inductor operation. It allows the inductor to charge naturally during the charge cycle by starting the charge when the inductor is fully discharged and discharging the inductor well before it reaches a steady-state charge (which causes wasted charge time).

This design is not possible without the presence of real-time interrupts and a two-clock-cycle interrupt overhead. If the interrupt pin is polled, the charging scheme becomes erratic and unpredictable. The charge cycles can be as low as 30 µs. Even if polled within 4 µs of every interrupt, the charge rates vary by 13.3% from cycle to cycle. If a power control scheme is only 13% accurate, it is doubtful that any engineer would choose to use it. The accuracy of this scheme is bound only by the accuracy of the comparator offset voltage (10 mV for the Z8E001). When using the 0.6 V reference in this circuit, the accuracy is better than 1.6%. If the reference voltage is doubled, the accuracy is better than 1%.

If the interrupt overhead were too high, a second interrupt might occur before the first interrupt had completed its processing. Interrupts can stack up, resulting in a total loss of output control. With the Z8E001, this situation never occurs, as the interrupt clock cycle overhead is minimal.

This power control scheme also depends on efficient implementation of lookup tables. Lookup table values are fetched in one clock cycle with Z8Plus MCUs by use of the load constant (LDC) or load constant and auto-increment (LDCI) instructions. These instructions allow the retrieval of lookup table values without *jumping* out of normal program code execution. The one clock-cycle lookup table fetch enables the microcontroller to reload the timer quickly to count the FET off-time. The register-to-register architecture also enables data to be transferred quickly without having to first load an accumulator or *working* register, then load that value to the register for which the data was originally intended. When dealing with only 30 µs FET charge cycles, every microsecond counts.

The presence of an on-board comparator also allows the Z8E001 to control the charge scheme with few external components. The 16-bit timer allows long timing intervals for the 1-second flash rate, and one of the two remaining 8-bit timers is used to measure the charge cycle period. Even though there are many timing functions occurring in this design, there is still an additional 8-bit timer/counter available if other timing and control functions are required.

One important fact that must not be overlooked is that this microcontroller functions without error even though the design poses many potential noise and voltage spike problems. Even with a 30 kHz+ FET switching frequency, 200 V+ discharge capacitor charge, 200 V+ trigger transformer capacitor charge and pulse, and 3000 V+ trigger transformer secondary pulse, the Z8E001 continues to accurately monitor and control the charge and flash rates.

This same power control algorithm can be used for many other types of power control systems, including DC/DC converters and motor control. It can be adapted to many different types of power and voltage control situations.

Technical Support

Pin Diagram

Figure 3. Z8E001 18-Pin DIP Configuration

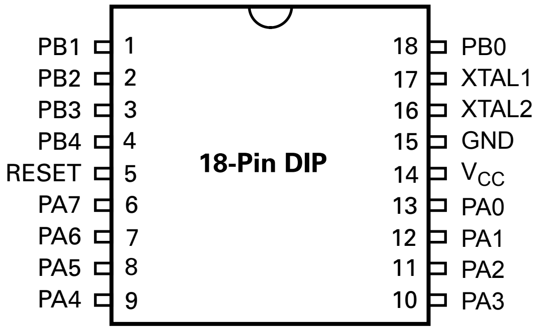


Table 1. Z8E001 18-Pin Configuration Description

Pin Number	Symbol	Description
1	PB1	
2	PB2	
3	PB3	Reference
4	PB4	Charge Monitor
5	RESET	VBO/POR Circuit
6	PA7	
7	PA6	
8	PA5	
9	PA4	
10	PA3	
11	PA2	
12	PA1	Strobe SCR
13	PA0	Charge FET
14	V_{CC}	5V Bus
15	GND	Circuit Ground
16	XTAL2	XTAL Circuit
17	XTAL1	XTAL Circuit
18	PB0	

Flowcharts

Figure 4. Main Routine

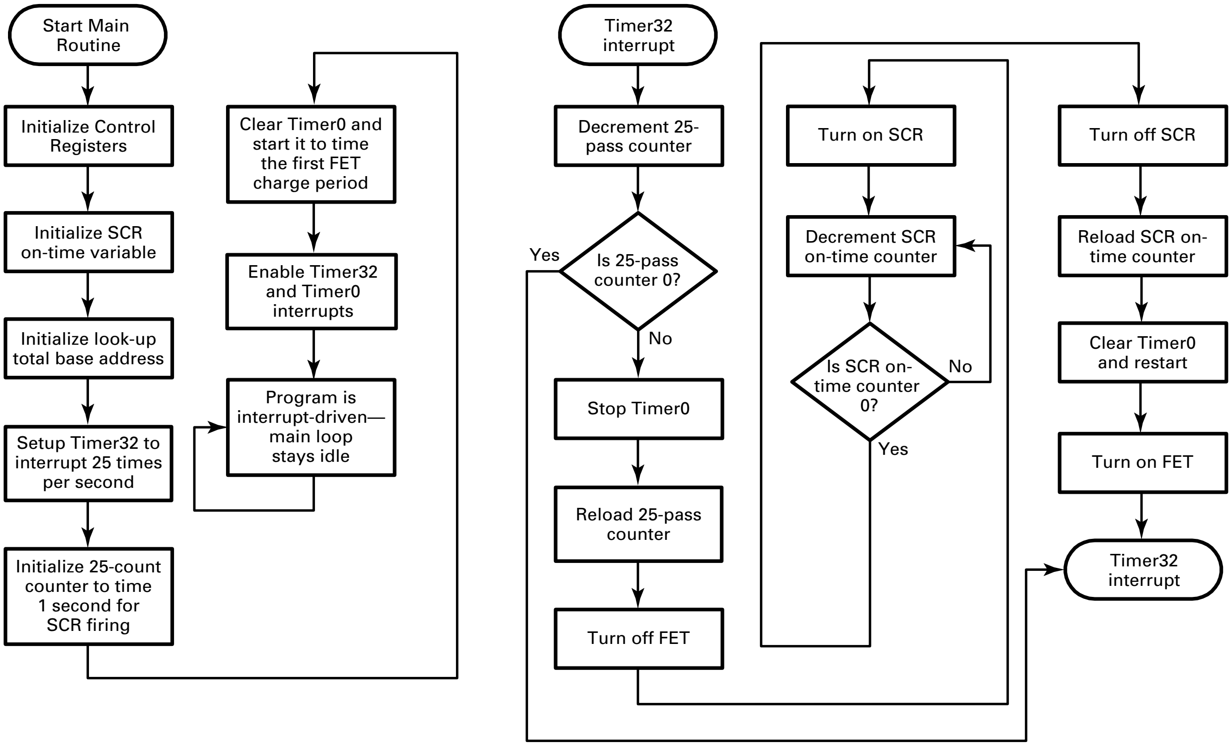
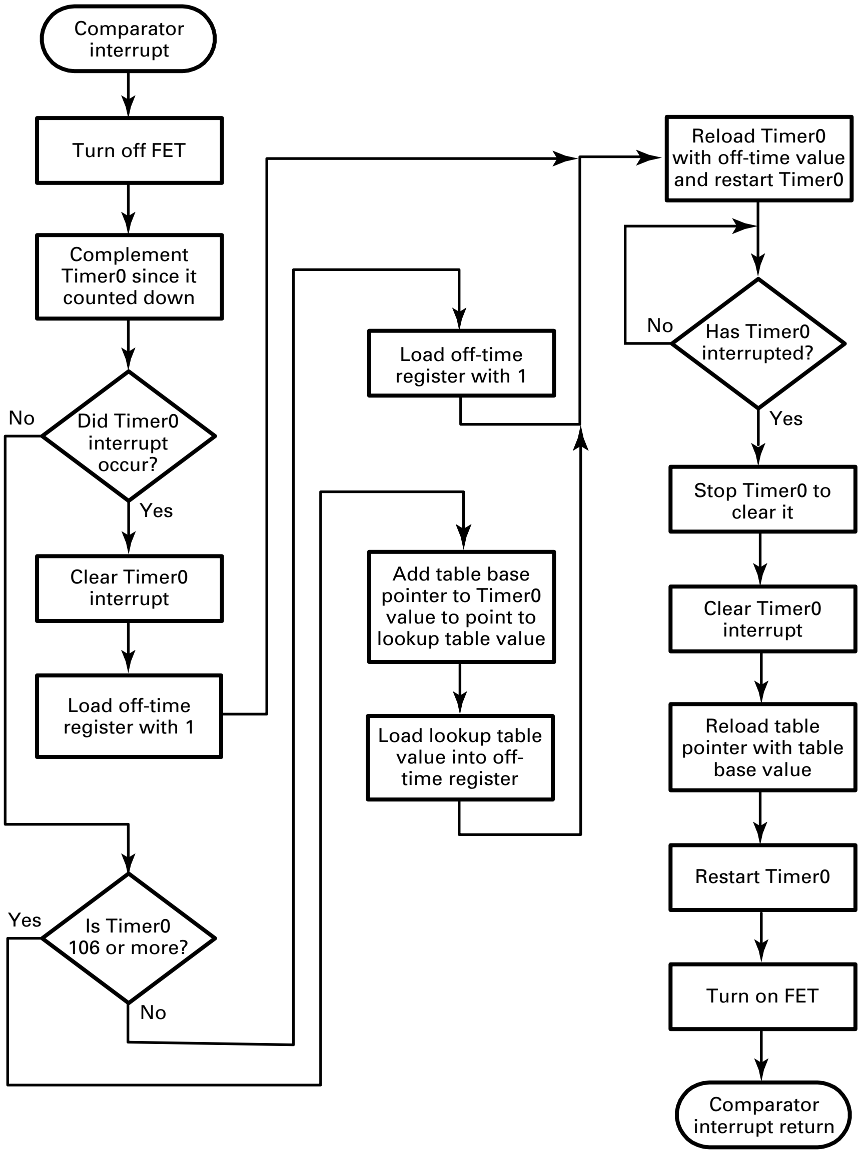

Start Main Routine
Initialize Control Registers
Initialize SCR on-time variable
Initialize look-up total base address
Setup Timer32 to interrupt 25 times per second
Initialize 25-count counter to time 1 second for SCR firing
Clear Timer0 and start it to time the first FET charge period
Enable Timer32 and Timer0 interrupts
Program is interrupt-driven—main loop stays idle
Timer32 interrupt
Decrement 25-pass counter
Is 25-pass counter 0?
Yes
No
Stop Timer0
Reload 25-pass counter
Turn off FET
Turn on SCR
Decrement SCR on-time counter
Is SCR on-time counter 0?
No
Yes
Turn off SCR
Reload SCR on-time counter
Clear Timer0 and restart
Turn on FET
Timer32 interrupt

Figure 5. Comparator Interrupt


Comparator interrupt
Turn off FET
Complement Timer0 since it counted down
Did Timer0 interrupt occur?
No
Yes
Clear Timer0 interrupt
Load off-time register with 1
Is Timer0 106 or more?
Yes
No
Load off-time register with 1
Add table base pointer to Timer0 value to point to lookup table value
Load lookup table value into off-time register
Reload Timer0 with off-time value and restart Timer0
Has Timer0 interrupted?
No
Yes
Stop Timer0 to clear it
Clear Timer0 interrupt
Reload table pointer with table base value
Restart Timer0
Turn on FET
Comparator interrupt return

Source Code

```
*******************************************************************
* power control2.asm
* 8/9/99
* ZiLOG Z8E001 Microcontroller
*******************************************************************
```

```
*
* This program controls the charge and flash rates for a strobe
* flash. The micro monitors the
* charge current of the "kick" inductor, and when the voltage is
* high enough, the inductor
* control FET is turned off. The on-time is timed by a timer/
* counter, and depending on this
* on-time, the off-time is determined by the lookup table. This
* inductor controls the charge
* rate of the capacitor that is used to fire the strobe. The micro
* also controls the timing
* between strobe flashes, and is set at 1s. An SCR is fired by the
* micro at each timeout
* period. The micro maintains the charge capacitor voltage over the
* entire 12V to 24V input
* voltage
*******************************************************************
*
  globals on
  define regdata,space=rfile space is 0 to 63
  define regcontrol,space=rfile space is 192
  to 255
  define vectordata,space=ROM       ;space is 0 to 31
  define codedata,space=ROM         ;space is 32 to
  1023

****NOTE**** Above ranges are set up in the ZDS Linker settings

  segment regdata


PASS_COUNTER        ds    1     ;counts timer interrupts for 1s
pass_counter        equ   r0

STATUS              ds    1     ;bit 0 determines if the
status              equ   r1    ;inductor routine has been
                                ;called or is an interrupt request

SCR_ON_COUNTER_HI   ds    1     ;length of time to keep the SCR on
scr_on_counter_hi   equ   r2

scr_on_counter      equ   rr2

SCR_ON_COUNTER_LO   ds    1
scr_on_counter_lo   equ   r3

POINTER_HIGH        ds    1     ;location of lookup table value
                                ;needed to control
```

```
pointer_high        equ   r4    ;the FET off time

pointer             equ   rr4

POINTER_LOW         ds    1
pointer_low         equ   r5

POINTER_HOLD_HIGH   ds    1     ;lookup table base address
pointer_hold_high   equ   r6

POINTER_HOLD_LOW    ds    1
pointer_hold_low    equ   r7

OFF_HOLD            ds    1     ;holds the required FET off-time
off_hold            equ   r8


; Program masks

SCR             equ   00000010b   ;PA1 is scr output
FET             equ   00000001b   ;PA0 is FET output
inductor_int    equ   00010000b   ;irq4 is inductor interrupt
timer32         equ   00100000b   ;timer2 interrupt is irq5, and
T32 start
                                  ;bit for the tctllo is also bit 5
                                  ;Program constants

scr_on_period   equ   1000        ;every 1000 counts equals 2ms
that
                                  ;SCR will be held on
timer_load      equ   50000       ;equals 1 sec and is loaded into
                                  ;timer32
                                  ;(used in conjunction with
                                  ;"timer_passes"
                                  ;to count 1 sec)
timer_passes    equ   25          ;number of timer timeouts before
                                  ;SCR is fired
                      segment regcontrol

tctllo                ds    1       ;C0
tctlhi                ds    1       ;C1

                      ds    %C4-%C2
t0arlo                ds    1       ;C4
t1arlo                ds    1       ;C5
t0arhi                ds    1       ;C6
t1arhi                ds    1       ;C7
t2ar                  ds    1       ;C8
```

```
t3ar                  ds    1       ;C9
t2val                 ds    1       ;CA
t3val                 ds    1       ;CB
t0val                 ds    1       ;CC
t1val                 ds    1       ;CD

                      ds    %D0-%CE
ptain                 ds    1       ;D0
ptaout                ds    1       ;D1
ptadir                ds    1       ;D2
                      ds    1
ptbin                 ds    1       ;D4
ptbout                ds    1       ;D5
ptbdir                ds    1       ;D6
ptbsfr                ds    1       ;D7

                      segment vectordata

                      jr start      ;rollover vector
                      dw    start   ;irq0 not used
                      dw    start    ;irq1 not used
                      dw    start   ;irq2 not used
                      dw    start   ;irq3 not used
                      dw    inductor ;FET interrupt (irq4)
                      dw    timer2  ;timer32 interrupt (irq5)

                segment codedata

start:  ld tctlhi,#0            ;disable WDT
start1: di
        clr tctllo              ;disable timers (required because
                                ;the micro program jumps back to
                                ;start if a dropout signal is
                                ;received, and timers will be
        ld regptr,#0            ;running) working register group 0
        ld spl,#%40             ;stack is at top of RAM
        ld ptbout,#0            ;clear outputs
        ld ptaout,#0
        ld ptbdir,#00000000b    ;port b all inputs
        ld ptadir,#00000011b    ;port a bits 0,1,4,5,6,7 except
                                ;PB1 are outputs
        ld ptasfr,#00000000b    ;all outputs are push/pull
        ld ptbsfr,#00011000b    ;ptb3 and ptb4 are comparator
                                ;inputs
                                ;(bit 3 is reference, bit 4 is
                                ;input)
                                ;ptb2 enabled as input
        ld imask,#00110000b     ;enable IRQ 4, 5
```

```
                                    ;initialize program values

        ld pass_counter,#timer_passes timer counter (25 int's/1s)

                                    ;initialize the SCR hold on
                                    ;counter

        ld scr_on_counter_hi,#HIGH(scr_on_period)
        ld scr_on_counter_lo,#LOW(scr_on_period)

        clr status
        clr ireq

        ld pointer_high,#HIGH(table15)  ;load lookup table base
                                        ;pointer
        ld pointer_Low,#LOW(table15)


check_done:and ireq,#(~inductor_int)    ;clear inductor interrupt


;save the pointer base address in the hold registers for reloading
;into the table pointer
;registers
        ld pointer_hold_high,pointer_high
        ld pointer_hold_Low,pointer_Low

*******************************************************************
* Timer setup
*
* The oscillator used in this design is 10MHz, so the frequency at
* the timer is 10MHz/8=1.25MHz.
* The total countdown timeout is 1s, so the timer will countdown
* 40ms, and repeat this
* count 25 times before the SCR is fired. Therefore, 40ms/
* 0.8us=50000. 25 counts of 50000
* will equal 1s. A separate 25-occurrence countdown counter is
* included in the interrupt
* routine.
*******************************************************************

timer_setup:ld t3ar,#HIGH(timer_load)
                               ;load counter for t32
                               ;countdown
            ld t2ar,#LOW(timer_load)
            ld t3val,#HIGH(timer_load)
            ld t2val,#LOW(timer_load)
            clr ireq           ;clear interrupt request register

            ld tctllo,#timer32 ;enable t32

fet_timer_setup:clr t0arlo     ;0.8uS/count for FET period
            clr t0val
            or tctllo,#00000001b;enable t0 to time FET on-time

            or ptaout,#FET     ;start inductor charge
            ei                 ;enable interrupts

main:       jr main

******************************************************************
* Timer 32 Interrupt
*
* This interrupt (IRQ5)is entered every 40ms. To count a full 1s,
* this routine is entered
* 25 times, so a countdown counter is included and reset in this
* routine. After 25 interrupts,
* the SCR is fired. The scr-on-counter is decremented to set the
* scr on-time, then the SCR is
* again turned off. The FET on-time t0 counter is restarted, and
* the FET output is turned back
* on to restart the cycle (before the SCR is fired, the FET is
* turned off, and timer0 is stopped).
*
******************************************************************

timer2: djnz pass_counter,exit_now      ;each decrement is 100s

        and ptaout,#(~FET)              ;turn off FET
        and ireq,#(~inductor_int)       ;cancel any pending
                                        ;inductor ints

        and tctllo,#11111110b           ;stop timer0
        and ireq,#11111110b             ;clear timer0 ireq0

        ld pass_counter,#timer_passes   ;reload ten occurrence
                                        ;counter
        or ptaout,#SCR                  ;turn on SCR output

scr_wait:decw scr_on_counter            ;decrement scr_on_counter
        jr nz,scr_wait

        and ptaout,#(~SCR)              ;turn off the SCR
```

```
;reload the SCR on-time counter

        ld scr_on_counter_hi,#HIGH(scr_on_period)
        ld scr_on_counter_lo,#LOW(scr_on_period)


fet_timer_setup2:clr t0arlo             ;clear FET interrupt timer
        clr t0val
        or tctllo,#00000001b            ;enable t0

        or ptaout,#FET                  ;turn the FET back on

exit_now:iret

*******************************************************************
* Comparator Interrupt
*
* This routine is entered each time that the comparator input
voltage rises above the reference
* voltage. This routine turns the FET output off, stops and
complements timer0 (started when the
* FET was turned on to measure the on-time), then retrieves the
corresponding off-time value from
* the lookup table. Timer0 is then reloaded with this value and
started again. This routine then
* waits for the timer to time-out, then the FET is turned back on.
T0 is then reset and restarted
* to again time the on-period of the FET signal.
*
****************************************************************************************************

inductor:   and ptaout,#(~FET)      ;turn off FET output to stop
                                    ;inductor charging
            and tctllo,#11111110b   ;stop timer0
            com t0val               ;complement t0 since it has
                                    ;counted down

;if timer0 interrupt is set, then the sawtooth period was greater
than 256 timer0 counts, or
;204us (frequency at timer is 10MHz/8 = 1.25MHz)

            tm ireq,#00000001b      ;timer0 interrupted?
            jr z,highcheck          ;if not, proceed with the high
                                    ;limit check

            and ireq,#11111110b     ;clear the timer0 interrupt
            ld off_hold,#1          ;set the off-time to 1 since
                                    ;the t0 int occurred
                                    ;signifying a long on-period
            jr proceed3             ;skip over lookup table
                                    ;retrieval

highcheck:  cp t0val,#106  ;was the high limit value reached?
            jr ult,normal  ;if not, proceed as normal
            ld off_hold,#1 ;if so, make off-time as short as
            possible
            jr proceed3             ;skip over lookup table
                                    ;retrieval

normal:     add pointer_Low,t0val ;add location offset to table
location
            jr nc,proceed ;if carry flag, add 1 to High byte
            inc pointer_high

proceed:    ldc off_hold,@pointer ;get the lookup table value
proceed3:   ld t0val,off_hold ;load the counter with the off-time
            value
            ld t0arlo,off_hold
            or tctllo,#00000001b    ;t0 single pass

keep_checking: tm ireq,#00000001b         ;timer0 timed out?
               jr z,keep_checking         ;if so, off-time wait is
                                          ;complete
               and tctllo,#11111110b      ;stop timer0
               clr t0val
               clr t0arlo
               and ireq,#11111110b        ;clear ireq0
               and ireq,#(~inductor_int)  ;clear comparator int

;load original lookup table values

            ld pointer_high,pointer_hold_high
            ld pointer_Low,pointer_hold_Low
            or tctllo,#00000001b    ;restart timer0 to measure FET
                                    ;on-time
            or ptaout,#FET          ;turn on FET
            iret

******************************************************************
* Lookup Table
*
* The lookup table is loaded with FET off_time values that are used
* to control the FET off time.
* The larger the number, the longer the off-time.
```

```
*
*******************************************************************

table15:    db 255    ;0.8us 1
            db 255    ;1.6us 2
            db 255    ;3.2us 3
            db 255    ;4.0us 4
            db 255    ;4.8us 5
            db 255    ;5.6us 6
            db 255    ;6.4us 7
                      db 255   ;7.2us 8
            db 255    ;8.0us 9
            db 255    ;8.8us 10
            db 255    ;9.6us 11
            db 255    ;10.4us 12
            db 255    ;11.2us 13
            db 255    ;12.0us 14
            db 255    ;12.8us 15
            db 255    ;13.6us 16
            db 255    ;14.4us 17
            db 255    ;15.2us 18
            db 250    ;16.0us 19
            db 245    ;16.8us 20
            db 240    ;17.6us 21
            db 235    ;18.4us 22
            db 230    ;19.2us 23
            db 225    ;20.0us 24
            db 220    ;20.8us 25
            db 215    ;21.6us 26
            db 210    ;22.4us 27
            db 206    ;23.2us 28
            db 202    ;24.0us 29
            db 199    ;24.8us 30
            db 196    ;25.6us 31
            db 193    ;26.4us 32
            db 190    ;27.2us 33
            db 187    ;28.0us 34
            db 184    ;28.8us 35
            db 181    ;29.6us 36
            db 178    ;30.4us 37
            db 175    ;31.2us 38
            db 172    ;32.0us 39
            db 169    ;32.8us 40
            db 166    ;33.6us 41
            db 163    ;34.4us 42
            db 160    ;35.2us 43
            db 157    ;36.0us 44
            db 154    ;36.8us 45
            db 151    ;37.6us 46
            db 148    ;38.4us 47
            db 145    ;39.2us 48
            db 141    ;40.0us 49
            db 138    ;40.8us 50
            db 135    ;41.6us 51
            db 132    ;42.4us 52
            db 129    ;43.2us 53
            db 126    ;44.0us 54
            db 123    ;44.8us 55
            db 120    ;45.6us 56
            db 117    ;46.4us 57
            db 114    ;47.2us 58
            db 112    ;48.0us 59
            db 110    ;48.8us 60
            db 108    ;49.6us 61
            db 106    ;50.4us 62
            db 104    ;51.2us 63
            db 102    ;52.0us 64
            db 100    ;52.8us 65
            db 98     ;53.6us 66
            db 96     ;54.4us 67
            db 94     ;55.2us 68
            db 92     ;56.0us 69
            db 90     ;56.8us 70
            db 88     ;57.6us 71
            db 86     ;58.4us 72
            db 84     ;59.2us 73
            db 82     ;60.0us 74
            db 80     ;60.8us 75
            db 78     ;61.6us 76
            db 76     ;62.4us 77
            db 74     ;63.2us 78
            db 72     ;64.0us 79
            db 70     ;64.8us 80
            db 68     ;65.6us 81
            db 66     ;66.4us 82
            db 64     ;67.2us 83
            db 62     ;68.0us 84
            db 60     ;68.8us 85
            db 58     ;69.6us 86
            db 56     ;70.4us 87
            db 52     ;71.2us 88
            db 40     ;72.0us 89
            db 40     ;72.8us 90
            db 40     ;73.6us 91
            db 40     ;74.4us 92
            db 40     ;75.2us 93
```

```
            db 40     ;76.0us 94
            db 40     ;76.8us 95
            db 40     ;77.6us 96
            db 40     ;78.4us 97
            db 40     ;79.2us 98
            db 40     ;80.0us 99
            db 40     ;80.8us 100
            db 40     ;81.6us 101
            db 40     ;82.4us 102
            db 40     ;83.2us 103
            db 40     ;84.0us 104
            db 40     ;84.8us 105
            db 40     ;85.6us 106

*****************************************************************
```

Assembling/Compiling Code

To assemble and compile the code, the ZiLOG Developer Studio (ZDS) Version 2.11 is used, and the target micro is the Z8E001. The code above is stored as an .asm file and must be included in the project when the project is first created. This source file (used with ZDS) is the only file that required for proper assembling/compiling of the program.

The *define* assembler directives allow the linker to set the ROM and RAM space boundaries for proper execution of the code. Because the actual code does not specify the ROM and RAM boundaries, these values must be specified within the ZDS linker settings. This setup is accomplished by selecting *project–settings–linker–ranges* from the ZDS menu. The define statements containing *space* = *ROM* denote code segment space, while the statements containing *space* = *rfile* denote register or RAM space. The proper linker range settings are as follows (decimal values):

```
vectordata = 0 to 31 (ROM space)
codedata = 32 to 1023 (ROM space)
regdata = 0 to 63 (RAM space)
regcontrol = 192 to 255 (RAM space)
```

Timing Diagrams and Tables

The full lookup table is shown in the Technical Support—Source Code section. As mentioned during the Theory of Operation discussion, this table is used by the program to determine the length of the *holdoff* or off-time between inductor charging cycles. The *db* is a ZiLOG assembler directive to assign a byte of ROM to the value that follows the statement. For example, the statement *db 111* assigns the value 111 (decimal) to a byte of ROM. It does not matter where in ROM the value is assigned—that is the beauty of the lookup table! To set the start of the look-up table in the source code, two instructions are required, as follows:

```
ld pointer_high,#HIGH(table15)
ld pointer_low,#LOW(table15)
```

These instructions take the actual 2-byte ROM address of the start of the lookup table and put the High byte into pointer_high and the Low byte into pointer_low. Now the table is used and the correct off-value is retrieved. The following instructions perform this data retrieval function:

```
normal:               add pointer_low,t0val ;add location offset
to table location
jr nc,proceed ;if carry flag, add 1 to High byte
inc pointer_high
proceed:              ldc off_hold,@pointer ;get the lookup table
value
proceed3:             ld t0val,off_hold ;load the counter with the
off-time value
ld t0arlo,off_hold
or tctllo,#00000001b  ;t0 single pass
keep_checking:        tm ireq,#00000001b;timer0 timed out?
jr z,keep_checking    ;if so, off-time wait is complete
```

Just before the execution of these instructions, the comparator interrupt occurred, and the length of the FET on-time is held in the t0val register (timer 0 actual value register).The first three instructions load the 2-byte pointer register with the actual look-up table value that is retrieved (the Low byte of the pointer table address is added to the t0val to get the table offset, and if there is a carry, then the High byte is incremented because the table locations requires a full 16-bit address). The *ldc* instruction then loads the value at the table location pointed to by the 16-bit *pointer* register pair into the *off-hold* register (this value is the required off-time that correlates to the FET on-time table offset that was used to obtain the table location). This value is loaded into timer 0, and the timer is restarted. The timer 0 interrupt is then polled, and when it is set, the FET is turned back on, and this process repeats itself when the comparator interrupt is received.

The other values in the table are for reference only. The first value after the semicolon shows the on_time FET period that corresponds to the off-time that is loaded into the lookup table. For example, the code:

```
db 14                 ;40.8µs 50
```

indicates that the value of 14 is used as the FET off time when the on-time is measured to be 40.8µs. The 50 value gives the numerical sequence value of the table location.

Test Procedure

Equipment

The equipment used to test and emulate this Application Note is as follows:

- ZiLOG Z8ICE000ZEM or Z8ICE001ZEM emulator
- Variable DC power supply (14V to 24V range)
- True RMS voltmeter with maximum value hold
- Oscilloscope

The jumper settings for the emulator are as follows:

1. JP1 is open to isolate emulator power from the target board (target board is powered from the variable DC supply).
2. JP2 is connected. It connects the RESET pin of the ICE chip to pin 5 of the emulated Z8Plus.
3. JP3 is open to isolate the ICE chip oscillator from the target board.
4. JP4 is open to isolate the ICE chip oscillator from the target board.

General Test Setup and Execution

The test procedures are as follows:

1. Connect the DC adapter to the emulator to power the board.
2. Connect the 18-pin ribbon cable to the target board and emulator.
3. Connect the variable DC power supply to the +/– wires on the target board.
4. Connect the oscilloscope input lead to the Q2 FET source, and the ground lead to ground (the oscilloscope connection is across resistor R5).
5. Connect the voltmeter across capacitor C1.

After the power supply is adjusted for 14 volts, the emulator is started and the strobe begins flashing once every second. The oscilloscope shows the sawtooth FET current that is approximately 68µs in duration. The C1 capacitor voltage is approximately 188 volts (maximum voltage hold setting on the voltmeter). The same tests are taken for successive increments in the DC power supply. Table 2 shows the actual measured values for this circuit in 1-volt supply voltage increments.

⚠ **Warning:** This circuit produces dangerously high voltages on many of the components. Exercise extreme care when operating the circuit or modifying the circuit components.

Test Results

Table 2. Measured Values for Circuit

Input Voltage (V_{DC})	Capacitor Voltage (V_{DC})	FET Charge Cycle Period (µs)
14	188	68.0
15	187	62.4
16	186	59.2
17	186	55.2
18	186	48.8
19	187	46.4
20	188	44.0
21	187	41.6
22	190	40.0
23	190	37.6
24	190	36.8

This circuit is set up to maintain a voltage of 187 volts (flash voltage) across the flash capacitor, so the largest voltage regulation error is only 1.6%. The software can maintain an excellent voltage regulation, even though the FET charge resistor tolerance is 1%, and the capacitor has a 5% tolerance. To achieve better regulation, the look-up table values between the data points listed above can be tested and adjusted as required (these points were interpolated in Table 1).

The non-linearity of the look-up table is evident from the values shown in the table. At the lower voltages, the FET charge cycle period decreases rapidly with increases in voltage. At the higher voltages, the decrease in the charge cycle is much less for each successive increase in voltage. The look-up table enables the software to simulate a non-linear equation without using complicated mathematics.

To obtain higher voltages on the flash capacitor, one of three changes can be made:

1. Change the reference resistors R9 and R10.
2. Change the FET feedback resistor R5.

3. Change the values in the look-up table.

The most precise results are achieved when a combination of these three methods is used to either raise or lower the output energy. The Theory of Operation section contains a more detailed explanation of these changes.

The easiest way to *rough-in* the look-up table is to determine the table values for a few points in the voltage range. A low, mid, and high voltage value (in the appropriate range) is selected, and the resulting oscilloscope trace is observed. Estimate the hold-off time and entered the estimate into the table. In addition, the intermediate values are interpolated from the actual test values. The voltage is adjusted across its voltage range, and the table values are adjusted again.This process continues until the desired voltage or power control is obtained.

References

Everything You Always Wanted To Know About Flashtubes, EG&G Optoelectronics (publication date unknown).

IES Guide for Calculating the Effective Intensity of Flashing Signal Lights, Aviation Committee of the Illuminating Engineering Society (1964)

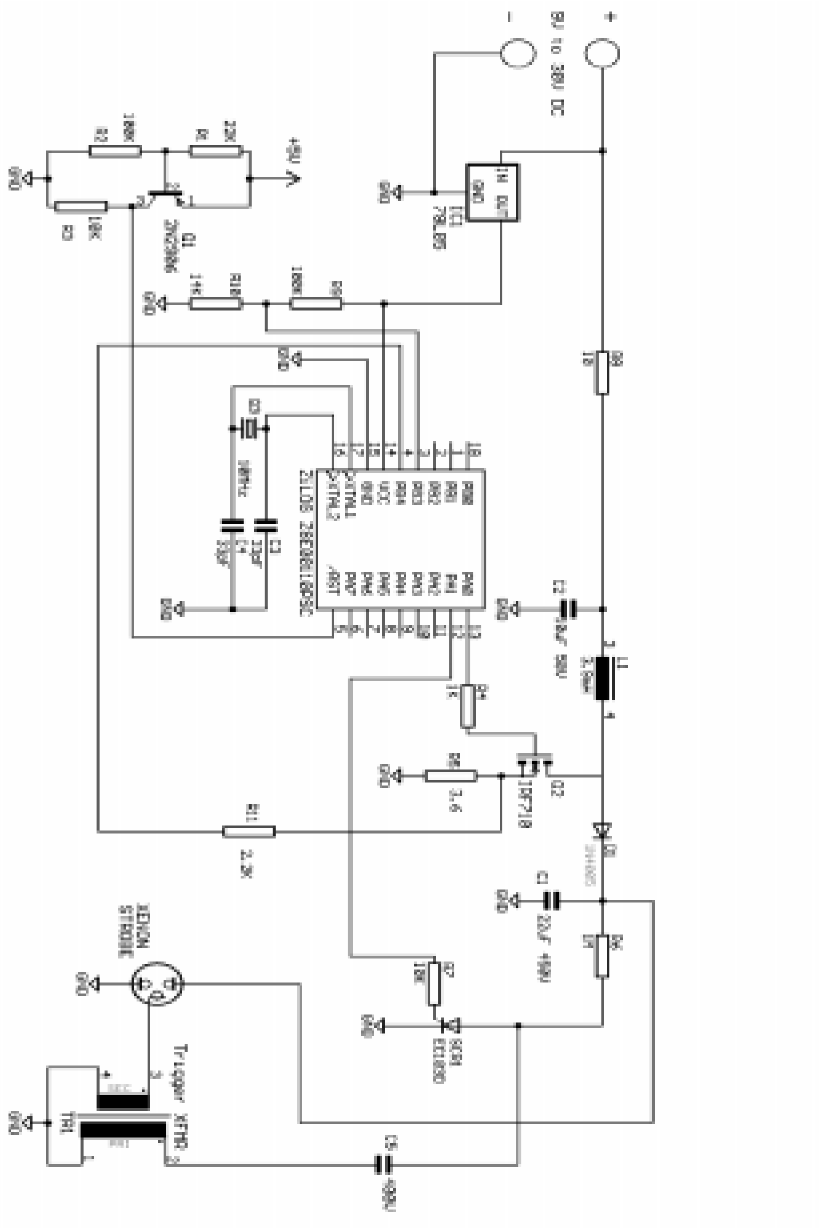
Glossary

Flashtube. A gas discharge tube designed to produce short pulses of light. Xenon is the normal gas used in flashtubes because it most efficiently converts electrical energy into visible photon energy.

Flashtube Construction. The flashtube is constructed of a glass or quartz tube, and can be in many different shapes and sizes. The tube is filled with Xenon gas, and an electrode is sealed in at each end

Appendix

Figure 6. Application Schematic


+
-
8V to 30V DC
IC1
78L05
IN
OUT
GND
R1
22K
R2
100K
R3
10K
Q1
2N3904
+5V
R9
100K
R10
14K
R8
10
ZILOG Z8E001PSC
XTAL1
XTAL2
VCC
GND
/RST
X1
10MHz
C3
33pF
C4
33pF
C2
.1uF 50V
L1
3.9mH
R4
1K
Q2
IRF710
R5
3.6
R11
2.2K
D1
C1
22uF 400V
R6
1M
R7
10K
SCR1
XENON
STROBE
Trigger XFMR
TR1
C5
GND