Application Note

Digital-to-Analog Conversion Using PWM in Z8 Encore! XP® Microcontrollers

AN015004-1207

Z8 Encore! XP® Flash Microcontrollers

Abstract

This application note describes a method to implement an 8-bit Digital-to-Analog Converter (DAC). This method utilizes the pulse-width modulation (PWM) feature of the on-chip timer in Zilog's Z8 Encore! XP® MCU. Design considerations are explained in detail with emphasis on practical implementation issues. Performance parameters of the DAC are published to aid designers when using this implementation for their applications.

The source code files, in WinZip format, associated with this Application Note, are listed below:

- AN0150-SC01—For 64 KB Z8 Encore! XP MCUs (Z8F640x)
- AN0150-SC02—For 64 KB Z8 Encore! XP MCUs (Z8F642x)
- AN0150-SC03—For 8 KB Z8 Encore! XP MCUs (Z8F082x)

Each source code file is available for download at www.zilog.com.

Z8 Encore! XP® Flash Microcontrollers

The Z8 Encore! XP products are based on Zilog's new eZ8 CPU and introduce Flash memory to Zilog's extensive line of 8-bit microcontrollers. Flash memory in-circuit programming capability allows faster development time and program changes in the field. The high-performance register-to-register based architecture of the eZ8 core maintains backward compatibility with Zilog's popular Z8® MCU. The Z8 Encore! XP microcontrollers combine a 20 MHz core with Flash memory, linear-register SRAM and an extensive array of on-chip peripherals. These peripherals make the Z8 Encore! XP products suitable for various applications including motor control, security systems, home appliances, personal electronic devices, and sensors.

Discussion

A DAC generates an analog output that is proportional to the digital input it receives. An *n*-bit DAC features a reference voltage commonly referred to as V_{REF}. The DAC output varies from 0 to V_{REF}, corresponding to the input, which varies from 0 to the maximum digital value (2^n–1) allowed at the inputs[1].

Theory of Operation

There are various techniques to achieve DAC functionality. This section discusses some of the popular techniques listed below:

- Binary-weighted Resistor Technique
- R-2R Ladder Technique
- PWM-Based Technique

Note: *This discussion is limited to the qualitative analysis of the implementation methods. For quantitative analysis of different techniques, refer to the book titled Digital Integrated Electronics by Herbert Taub and Donald Schilling (listed in References on page 7).*

Binary-weighted Resistor Technique

In the binary-weighted resistor technique (see Figure 1 on page 2) the analog output is generated by

1.This statement is true for a unipolar DAC. A bipolar DAC considers the digital input to be a signed number and varies its output from – to + as per the digital coding and/or interpretation of the input.

selecting different resistors with values corresponding to the weight (place) of the bit in the digital input.

Therefore, if the resistor value for the least-significant bit (lsb) is R, then the resistor value for the most significant bit (msb) is $R \div 2^n$, where *n* is the number of digital inputs.

The main drawback of this implementation is that it requires precision resistors in the range of 1 to 2^n. The ratio among these resistors must be maintained precisely. However, it is a ratio that is difficult to maintain.

Another drawback of the weighted resistor implementation is that it offers an output resistance that is dependent on the input. Therefore, for a fixed load on the output, the discrepancy of the actual output from the ideal output varies for different inputs.

R-2R Ladder Technique

The R-2R ladder technique overcomes the disadvantages of the binary-weighted resistor method. The R-2R ladder implementation uses a network of resistors (see Figure 2 on page 3) with just two resistor values—R and 2R. This technique is unlike the binary-weighted resistor method, wherein an increase in the number of bits results in an increase in resistors of different values.

The R-2R ladder network offers a fixed output impedance, which is independent of the digital input. Thus for a fixed load, the difference between the actual output and the ideal output is constant, regardless of the digital input.

The resistor network systems described above ideally require external switches that connect the ladder input to the voltages representing either logic 1 (V_{REF}) or logic 0 (Ground or 0 V). If V_{REF} is replaced by the supply voltage, then the resistors are directly connected to the microcontroller General-Purpose Input/Output (GPIO) pins. As a result, the number of GPIO pins utilized for DAC conversion depends on the number of DAC input bits or a network of external latches are required to hold the digital input.

PWM-Based Technique

In the PWM-based technique the digital input sets the pulse width of the signal, which is output to a single GPIO pin. This output signal is fed to a low-pass filter. The output of the low-pass filter is proportional to the width of the ON pulse of the PWM signal, which is proportional to the digital input. The number of GPIO pins is therefore unrelated to the number of bits that the DAC unit supports. This feature affords excellent scalability of the DAC unit.

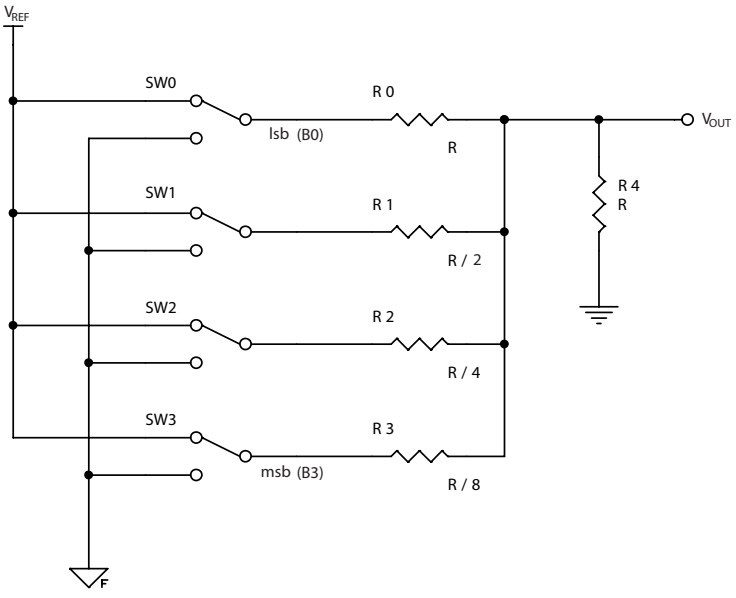

Figure 1. Binary-Weighted Resistor Circuit

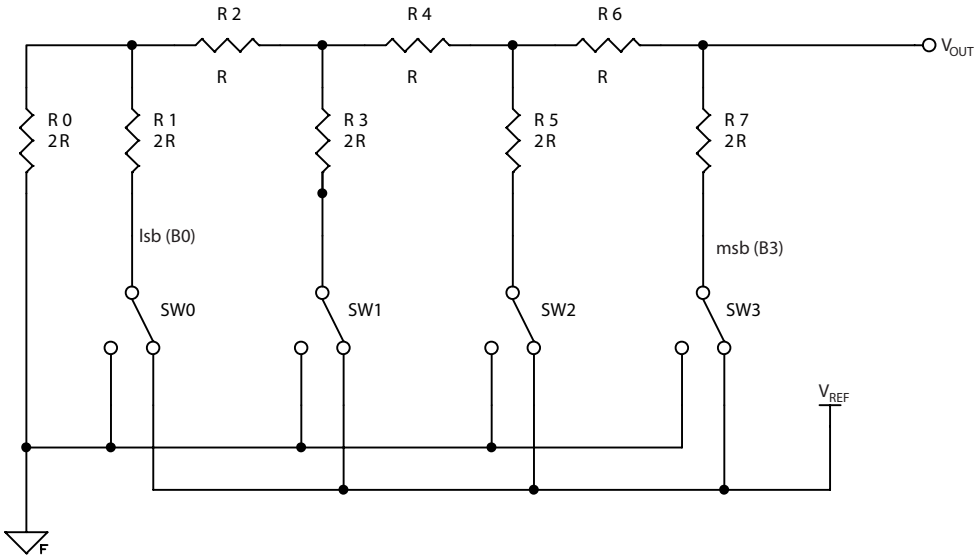

Figure 2. R-2R Circuit Diagram

The advantages of the PWM method over the resistor network-based methods are listed below:

- Single pin requirement from the microcontroller GPIO pins
- Extremely low external component requirement, without stringent precision requirements
- Output resistance that is independent of input

The PWM method requires a dedicated timer (an on-chip peripheral) operating in the PWM mode, and requires a number of PWM clock cycles for the output filters to reach a steady average output. Therefore, the PWM method requires a settling time that prevents it from being used in applications where the digital inputs are received before an averaging of the signals can occur.

Developing DAC with Z8 Encore! XP® MCU

This section describes the implementation of DAC functionality using the PWM method. The PWM output from the Z8 Encore! XP timer is passed through a low-pass filter. The low-pass filter effectively removes the PWM frequency component from the DAC outputs, and retains the DC value, which is the actual DAC output.

Hardware Architecture

The external component requirement for the Z8 Encore! XP PWM DAC is minimal (see Figure 3 on page 4). The PWM output of the microcontroller is passed through a simple resistor-capacitor (RC) low-pass filter. The PWM spectra vary significantly depending on the input values. The low-pass filter should accommodate for the worst case.

For superior performance, the low-pass filter is two-pole, providing an attenuation of 40 db/decade past its cut-off frequency. The higher PWM frequencies are filtered out by the low-pass filter, thereby reducing the noise in the DAC output.

There are several design considerations involved in the selection of the cut-off frequency of the low-pass filter. Primarily, the filter cut-off frequency must be much lower than the PWM frequency to reduce the noise generated by PWM switching.

However, a very low cut-off frequency imposes a corresponding limitation on the DAC data input rate, because the time constant associated with the filter increases.

As the rate of change of input is specific to an application, the only parameter under the designer's control is the PWM frequency. A microcontroller generates a PWM waveform by counting the number of clock pulses at a fixed rate and toggling the PWM output when it reaches the reload value. For higher counts, the microcontroller takes more time to reach the reload value, when it generates a low PWM frequency. It follows that for lower counts, the generated PWM frequency is high.

From a hardware standpoint, a higher PWM frequency is easier to filter, but it corresponds to a smaller reload value and therefore results in lower resolution. By contrast, a smaller PWM frequency is difficult to filter; moreover, the lower limit is dictated by the data input rate.

Considering these constraints, the Z8 Encore! XP PWM DAC is designed with the following specifications:

- The maximum number of data conversions per second is 100 (sampling frequency of 100 Hz).
- A cut-off frequency is approximately 800 Hz (8 times the input rate) to ensure that the fastest DAC output signal is not attenuated.

For optimum performance, an operational amplifier can be used as a buffer amplifier (see Figure 3).

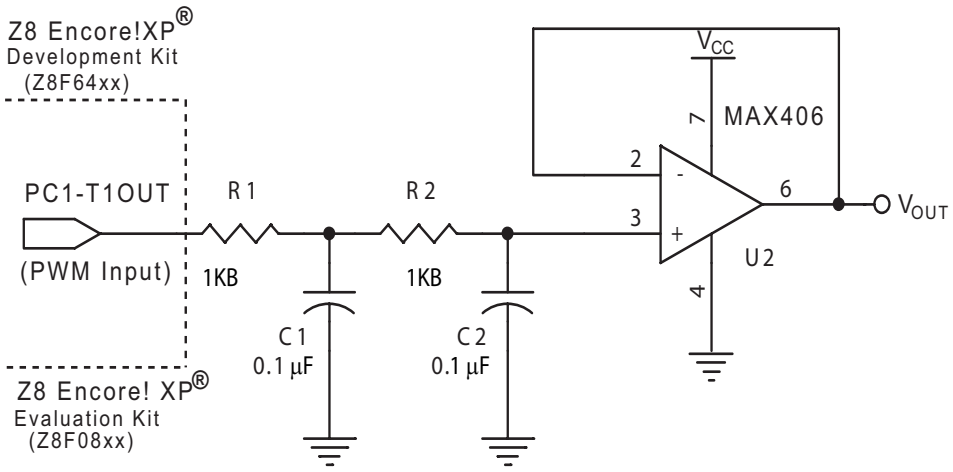

Figure 3. Circuit Diagram for Z8 Encore! XP PWM DAC

- The PWM signal is driven at 36 kHz so that the PWM frequency is within the filter stop band.

Single-supply operational amplifiers with low offset voltage and input bias current specifications are preferred. For cost-effective solutions, however, the operational amplifier can be eliminated, with the precaution that the load is at least 10 times the combined value of the resistors (R1+R2) used.

Software Implementation

The software implementation to achieve a digital-to-analog conversion using the PWM method is based on the following design issues:

- Scaling of the DAC digital input
- Accommodating the `0x00` digital input
- Response time before the next digital input can be processed

Scaling of the DAC Digital Input

The digital input is scaled to the appropriate levels according to the PWM reload value.

A Z8 Encore! XP timer in PWM mode, with a prescalar of 1 and a clock frequency of 18.432 MHz results in a PWM reload value of 0x398, which corresponds to a frequency of 20 kHz. With these specifications, for an 8-bit DAC input that runs from 0x00 to 0xFF, the digital input must be scaled from 0x000 to 0x398.

As the maximum digital input corresponds to the PWM reload value, the scaled values for the digital inputs are calculated using the following equation:

$$\text{Scaled value} = \frac{\text{reload value x digital input}}{\text{digital input (max)}}$$

$$= \left[\frac{\text{reload value}}{\text{digital input (max)}}\right] \text{x digital input}$$

There are two efficient ways to handle this computation:

- The multiplication in the derived equation can be simplified to a left-shift operation if the scaling factor (reload/digital max) is a power-of-two (1, 2, 4, 8, …).
- As the scaling factor is a constant for a given DAC implementation, it can be precomputed and stored for use.

The nearest value to 0x398 (also a power-of-two multiple of 0xFF) is 0x3FC (0xFF x 4 = 0x3FC). Selecting 0x3FC for the reload value with a prescalar of 1 results in a PWM frequency of 18.0705 kHz (approximately 18 kHz) at a system clock frequency of 18.432 MHz.

Note: *This PWM frequency falls within the audio frequency range and may not be suitable for all applications. Selecting 0x1FE as the reload value results in a PWM frequency of 36.141 kHz (36 kHz approximately), which is outside the audio frequency range.*

The benefit of the PWM technique is apparent, as it is tolerant to a larger variation in the system parameters than any of the resistor-based methods.

Accommodating 0x00 Input

For a PWM generator, the pulse-width value of 0x00 does not have any meaning. For Z8 Encore! XP MCU, setting the PWM registers with the value 0x00 results in a full-scale output. Therefore, for a digital input of 0x00, the timer PWM is disabled, resulting in an output of 0 V.

Response Time

The DAC output (past the low-pass filter) does not change instantaneously according to the DAC input variations. The time constant of the RC filter, which is the response time, determines the time during which the PWM reload value and the digital input must not be changed. This time duration is a constraint that directly affects the maximum DAC output signal frequency. Response time measurement can be converted from *seconds* to the *number of PWM cycles* for efficient use of resources.

In the present implementation, the 18 kHz PWM signal has a period of 55.55 µs. The RC time constant (τ) of the filter is 1.8 ms. The output stabilizes after roughly 5 x τ duration, which is 9 ms. 9 ms can accommodate 162 PWM cycles. As a result, 162 PWM interrupts can be counted before setting the dac_ready flag to TRUE.

For the DAC implementation, the number of PWM cycles (DELAY_COUNT) is fixed at 150. An update rate of 9 ms results in a frequency of 111 Hz, which is comfortably above the 100 Hz update rate specified for the DAC.

DAC APIs

The entire PWM DAC implementation consists of three application programming interfaces (APIs) and one interrupt service routine (ISR).

dac_initialize()

The dac_initialize() API performs the following operations:

- Initializes timer 1 for PWM functionality

- Sets the PWM frequency through the T1CPH and T1CPL registers
- Sets the timer interrupt with high priority
- Sets the pointer to the `dac_isr` for servicing the timer interrupt
- Sets the `dac_ready` flag

dac_start()

The `dac_start()` API performs the following operations:

- Gets the DAC input value from the calling program
- Resets the `dac_ready` flag and scales the input value to the appropriate PWM count
- Disables the timer 1 operation to load the scaled value in T1PWMH and T1PWML registers
- Enables the timer 1 after the loading is complete

dac_off()

The `dac_off()` API performs the following operations:

- Disables the timer 1 operation
- Sets the `dac_ready` flag

After the `dac_off()` API operations are performed, the output is not sustained.

In the software implementation, the `dac_ready` flag allows user programs to detect if the previous conversion is complete, and if the DAC output is stable. The user program then initiates a new conversion by calling the `dac_start()` API, with the digital input to be converted as its sole argument.

An ISR, `dac_isr()`, is provided for the same PWM timer's interrupt to count the number of PWM cycles that occur after the most recent input value. The filter response time is derived from the response time calculations discussed in section Response Time on page 5. This response time value is expressed in terms of the number of PWM cycles.

When the ISR count reaches this response time value, the `dac_ready` flag is set to TRUE, indicating to the user program that the output is stabilized until the final input word, and the `dac_start()` API can be called again with fresh input. The `dac_ready` flag frees the user program from response time considerations.

Testing Z8 Encore! XP® PWM Digital-to-Analog Converter

The Z8 Encore! XP PWM DAC implementation is tested on two Z8 Encore! XP MCUs—the Z8F640x and Z8F082x devices using the setup illustrated in Figure 3 on page 4.

Equipment Used

The equipments used for testing consist of the following:

- Z8 Encore! XP Development kit (Z8ENCORE000ZC0C) with Z8F640x MCU
- Z8 Encore! XP Development kit (Z8F64200100KIT) featuring Z8F642x MCU
- Z8 Encore! XP Development kit (Z8F08200100KIT) with Z8 Encore! XP 8K/4K MCU
- ZDS II IDE — Z8 Encore! v4.10.1 for the Z8F640x MCU
- ZDS II IDE — Z8 Encore! v4.10.1 for the Z8F642x MCU
- ZDS II IDE — Z8 Encore! v4.10.1 for the Z8F082x MCU
- Tektronix TDS724D oscilloscope

Procedure

Follow the steps below to test the Z8 Encore! XP PWM DAC:

1. The DAC code was downloaded to Z8 Encore! XP MCU's Flash memory; the project file was built using the appropriate ZDS II IDE, and the DAC program was executed.

2. The DAC input was varied from 0x00 to 0xFF.
3. The output signal was measured with a multimeter.
4. The measurements were tabulated and the performance parameters were determined (see Table 1).

Table 1. Performance Parameters

Parameter	Value	Units
Resolution	8	bits
Zero scale error (max)	0	mV
	0	% full scale
Full scale output	3.299	V
Gain error	1	mV
	0.03	% full scale
Differential nonlinearity error (max)	1.8	mV
	0.05	% full scale
Absolute accuracy error (max)	2.9	mV
Response time	6	ms

Results

The DAC implementation functions in the expected manner. The DAC filter attenuation is –19.04 dbV per decade. The DC parameters are tabulated in Table 1.

Summary

This application note describes an implementation of an 8-bit Digital-to-Analog Converter using the Z8 Encore! XP timer in PWM mode and a passive low-pass filter. This implementation addresses the design issues associated with scaling the digital input, accommodating the 0 input, and determining the response time. The Z8 Encore! XP PWM DAC's performance parameters are published.

This software-implemented DAC supports 8-bit resolution and is fully monotonic over the entire range. The DC performance parameters are comparable to commercially-available hardware digital-to-analog converters, making the Z8 Encore! XP PWM DAC ideal for low-frequency applications.

References

The documents associated with DAC and the Z8 Encore! XP MCU are listed below:

- DAC Theory:
  - *Digital Integrated Electronics* by Herbert Taub and Donald Schilling, McGraw Hill Publishers; ISBN 0-07-062921-8
- Z8 Encore! XP MCU documents available on www.zilog.com:
  - Z8 Encore!® 8K/4K Series Development Kit User Manual (UM0150)
  - Z8F08xx/Z8F04xx Flash Microcontrollers Product Specification (PS0243)
  - Z8 Encore! XP® 64K Series Flash Microcontrollers High-Performance 8-Bit MCU Product Specification (PS0199)
  - Z8 Encore!® Flash Microcontroller Development Kit User Manual (UM0146)
- ZDS II document available at www.zilog.com:
  - Zilog Developer Studio II–Z8 Encore!® User Manual (UM0130)

Appendix A—Glossary

Definitions for terms and abbreviations used in this application note are listed in Table 2.

Table 2. Definition of Terms/Abbreviations

Term/Abbreviation	Definition
DAC or D/A Converter	Digital-to-Analog Converter.
PWM	Pulse-Width Modulation.
Offset or zero-scale error	For a DAC, the offset error is the analog output value when the digital input is zero. Such an error affects all the codes by the same amount, and is usually compensated for by a trimming process. If trimming is not possible, then the error is referred to as the zero-scale error.
Gain error	For a DAC, the gain error is defined as the difference between the ideal analog output and the actual analog output when the digital input is full-scale.
Resolution	The resolution of the DAC is defined as the smallest change that can occur in the analog output as a result of a unit change in the digital input.
Differential nonlinearity error	For a DAC, the differential nonlinearity (DNL) error is the difference between the change in actual output and 1 lsb output for a unit change in the input value. If the DNL exceeds 1 lsb, then there is a possibility of the converter becoming non-monotonic. This statement means that the magnitude of the output grows smaller with each increase in the magnitude of the input.
Integral nonlinearity error	The integral nonlinearity error is the deviation of the actual analog output values from the ideal output values when the offset and gain error are adjusted to zero. For DAC, the deviations are measured at each step.
Absolute accuracy error	An absolute accuracy error is the difference between the actual output and the ideal output. This difference includes the offset, the gain, and the integral nonlinearity error.
Response Time	Response time is the time required for the output to reach the final value. This time is usually specified for a full-scale change in voltage.

Appendix B—Flowcharts

Appendix B displays the flowcharts of main DAC routine and DAC interrupt service routine (Figure 4 and Figure 5).

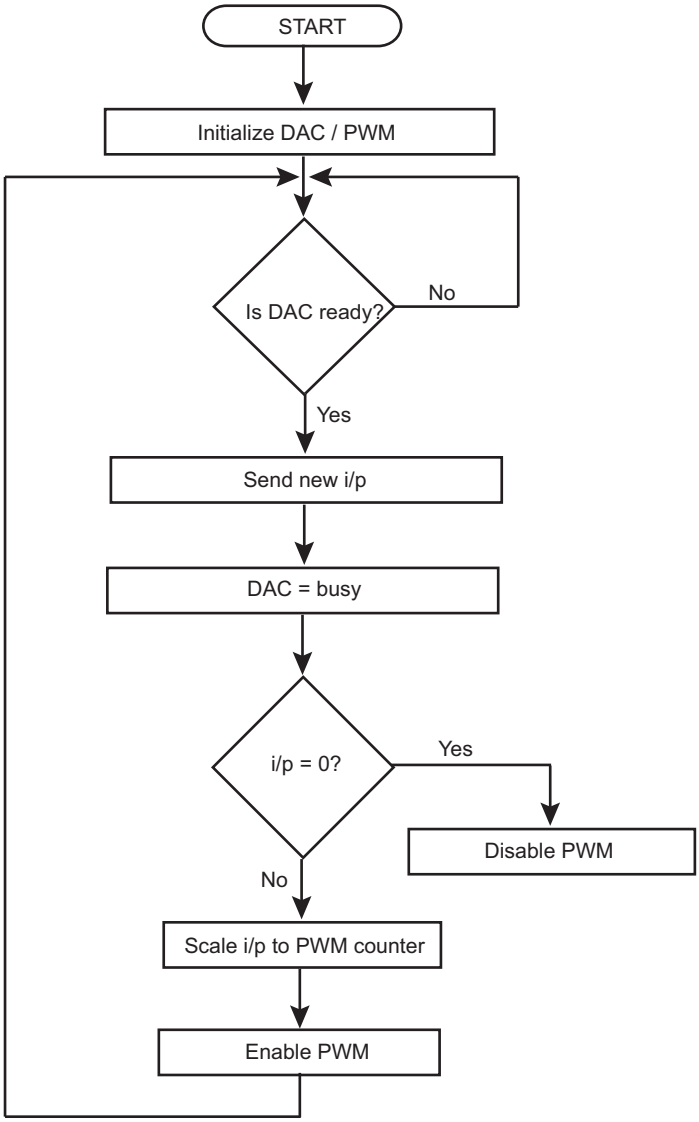

Figure 4. Flowchart for Main Routine in DAC Implementation

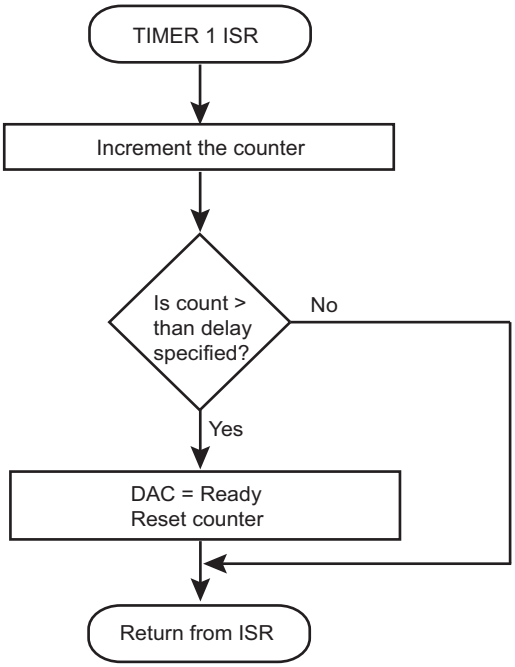

Figure 5. Flowchart for the DAC ISR

Warning: DO NOT USE IN LIFE SUPPORT

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer