

Controlling Analog Output From a Digital CPLD Using PWM

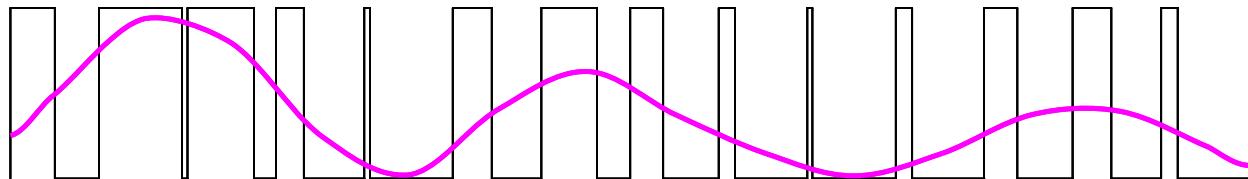
Although the Altera® MAX® IIZ CPLD is a digital programmable logic device, it is versatile enough to control analog systems. This white paper shows how the MAX IIZ CPLD, alone or with a few passive components, can replace a digital-to-analog converter, allowing it to drive an audio speaker and control LED intensity, motor speed, and servo position. This white paper also will explain pulse-width modulator (PWM) operation, as well as describe the efficient implementation and use of PWMs in a MAX IIZ CPLD.

Introduction

A pulse-width modulator (PWM) is a common way of generating analog outputs from a digital component. A PWM replaces a digital-to-analog converter (DAC), which generates analog voltage or current proportional to the digital input. As the name implies, a PWM generates a series of constant voltage or current digital pulses with pulse widths or duty cycles that are proportional to the intended analog strength. The series of modulated pulses can be converted to an analog voltage with a low-pass filter, but this is usually unnecessary.

Figure 1 shows a typical analog signal and the digital PWM representation. In general, an analog signal has a maximum amplitude, a minimum amplitude, and many levels in between. In contrast, the PWM only has two levels: maximum and minimum.

Figure 1. Analog Signal and Equivalent Pulse-Width Modulation



To convert from analog to digital, the analog signal first is sampled at a carrier frequency. For a given sample period, the area under the analog signal equals the area under the PWM pulse. The key principle behind the PWM is that a short pulse at maximum amplitude has the energy equivalent to a continuous analog signal at a lower amplitude. This simple equation determines the required sample frequency for a PWM circuit:

$$F_{SAMPLE} = 2 \times F_{RANGE},$$

where F_{SAMPLE} is the rate at which the analog signal is divided into digital packets, and F_{RANGE} is maximum frequency of the analog signal to be reproduced by the PWM. For audio, this may be 4 KHz for a phone or 20 KHz for an MP3 player. The “2” in the equation comes from the Nyquist frequency, which is the accepted oversampling rate required to reproduce an analog signal from digital samples.

The next step is to generate a clock to drive the PWM granularity. The following equation determines the PWM frequency:

$$F_{PWM} = 2 \times F_{RANGE} \times R,$$

where F_{PWM} is the clock frequency driving the PWM block, and R is the resolution. The resolution is typically a multiple of $2N$ (where N = number of bits in the digital data stream words) but with the proposed MAX IIZ PWM, any resolution is possible.

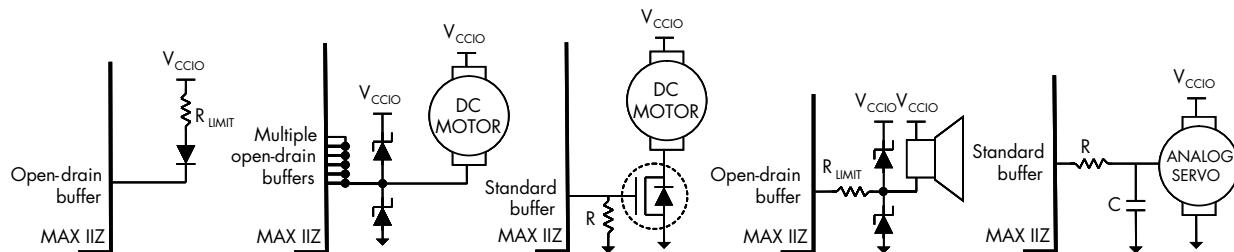
PWM Analog Output Applications

The three most useful analog applications for MAX IIZ PWMs are an LED driver, audio output, and motor control. These allow the MAX IIZ CPLD to control light, sound, and motion for the following functions:

- Light
 - Control LED blink intensity to save power
 - Display back-light intensity
 - Tri-color LED color mixing
- Sound
 - Audio play back
 - Audible warning messages
 - Ringtones and sound effects
 - Keyboard clicks and tones
- Motion
 - Motors
 - Phone vibrators
 - Game motion feedback
 - Warning vibrations for controls
 - Cooling fan control
 - Keyboard tactile feedback
 - Servos
 - Analog control voltage
 - Digital control pulse

Figure 2 shows how easy it is to connect a MAX IIZ CPLD to an LED, speaker, or motor. Typically, a minimal number of external components is required. One misconception about PWM outputs is that they must go through some type of filter to convert the digital signal back to analog before it can be used. In the following examples, only the analog servo motor requires a filter.

Figure 2. Circuits for Converting the PWM Signal to Light, Sound, and Motion



Light

A PWM controlling the light intensity is perhaps the simplest function to understand. The human eye cannot detect a light flashing faster than 240 Hz (a period of 4.2 mS), yet it can distinguish thousands of levels of brightness. The percentage of time an LED is on during the 4.2 mS can be as small as 0.01%, or 4.2 uS, and the light still will be seen as dim but not blinking. At 50% of the duty cycle, or 2.1 mS, it is seen as half intensity, and at 100% duty cycle, it is seen as full intensity. A current-limiting resistor prevents the full intensity output from damaging the LED or the MAX IIZ I/O buffer. The MAX IIZ CPLD has an 8-mA I/O current setting that limits the output current and may eliminate a resistor from the bill of materials (BOM).

Motors

MAX IIZ CPLDs can drive some motors directly. Some micro-vibration motors have maximum stall currents and operation currents that are less than 100 mA, and it is possible to drive multiple adjacent outputs with a common PWM signal. For example, a micro-vibration motor has a maximum current of 95 mA. Each MAX IIZ I/O is rated for 16-mA drive. Therefore, six open-drain I/Os working together can drive the motor without damaging the MAX IIZ CPLD.

The motor-drive circuit has two Schottky diodes to power and ground. These diodes are normally reverse biased, but will clamp any overshoot or undershoot voltage spikes generated from the motor coil when power is removed or the motor is running after the controller is powered down. [Figure 2](#) shows how to hook up a more powerful motor using a metal-oxide-semiconductor field-effect transistor (MOSFET) device. The gate of the MOSFET has a pull-down resistor that keeps the motor off during the board's power-up and power-down cycles.

Speed control with a PWM is very simple. The amount of energy sent to the motor is

$$voltage \times current \times time = joules.$$

A motor's power and RPM will be the same if the average voltage is 50 percent of the maximum over a set sample period, or if it is the maximum voltage for 50 percent of the sample period. The recommended sample period for a motor driver is

$$F_{SAMPLE} = 1/(2 \times (RPM_{MAX}/60)).$$

Staying above this rate eliminates detectable vibration. In addition, a PWM resolution as small as 0.01% can be used for speed or power control.

Sound

It may be necessary to add a current-limiting series resistor to the audio output (speaker or headphone) circuit. Because a speaker is a coil, it is important to have Schottky clamp diodes to prevent coil discharge overshoot and undershoot that could damage the MAX IIZ I/Os. One common misconception is that an audio output requires a filter to convert the digital PWM signal to an analog voltage. A speaker is a machine just like the motor, so the amount of energy imparted on air molecules by the cone is also

$$voltage \times current \times time = joules.$$

The speaker's ability to generate audio compression waves is the same if the average voltage is 50 percent of the maximum over a set sample period or if it is the maximum voltage for 50 percent of the sample period. A PWM carrier wave of 44.1 KHz generates a CD-quality frequency range, 11 KHz is used for MPEG audio, and 8 KHz is used for digital phone applications. Multiple open-drain I/Os can be used together to increase the power output of the PWM driving a speaker. Controlling the number of I/Os enabled is a simple volume control for an audio PWM system.

Servos

The traditional analog servo arm position is proportional to the control voltage value between the servo power and ground. The PWM output is converted to a stable DC voltage by a low-pass RC filter, and the RC filter value should be designed to have a low-pass cutoff frequency equal to the PWM sample frequency:

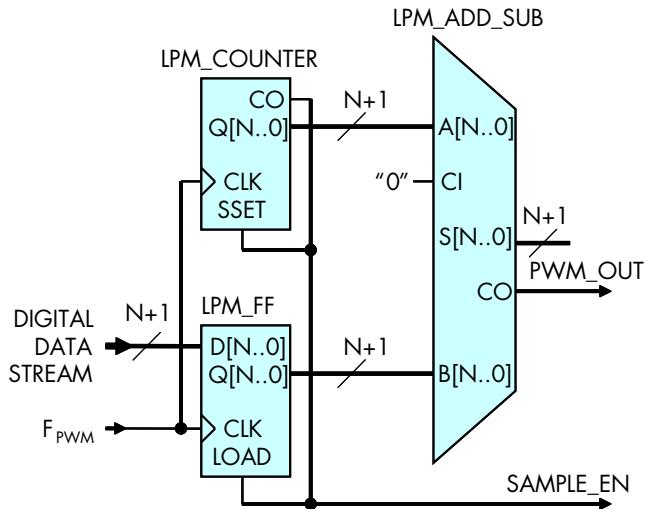
$$F_{SAMPLE} = 1/(2 \times \Pi \times R \times C)$$

A cutoff frequency below F_{SAMPLE} increases stability but reduces response time, while a cutoff frequency above F_{SAMPLE} increases jitter but reduces settling time. A new generation of servos uses digital pulse width to specify the servo arm angle, so MAX IIZ PWMs can drive a digital servo directly.

Implementing a PWM

The MAX IIZ architecture supports arithmetic functions much more efficiently than other CPLD architectures. It can implement counters and adders using only one logic element (LE) per bit of word width, and the programmable nature of MAX IIZ CPLDs makes it very easy to make counters of any width and count length. [Figure 3](#) shows the three parts of a MAX IIZ PWM, built using the Quartus® II software's Library of Parameterized Macros (LPM). These LPM blocks-LPM_FF, LPM_COUNTER, and LPM_ADD_SUB-are created using a GUI tool in Quartus II software, which generates Verilog, VHDL, and AHDL file formats. The PWM is clocked by an F_{PWM} -Hz clock source.

Figure 3. MAX IIZ PWM



The LPM_FF holds the digital value that is to be converted to analog by the PWM. The LPM_FF is loaded when the SAMPLE_EN signal is “1,” and the width is set by the bit width of the digital data stream.

The LPM_COUNTER is a binary counter that can be set synchronously by the SSET signal to any start value, but is typically “0.” The LPM_COUNTER carry output (CO) is “1” when the counter reaches its maximum count. The LPM_COUNTER CO generates the sample enable signal, SAMPLE_EN. Typically the sample period is

$$2^{(N+1)}/F_{PWM},$$

but a custom value can be specified for the synchronous set value in the LPM_COUNTER MegaWizard® Plug-In. A PWM with M levels of resolution is made by assigning the synchronous set value to

$$2^{(N+1)} - M - 1.$$

Examples where a non- 2^N resolution would be useful include controlling LED brightness in one percent increments from 0 to 100%, which would require a seven-bit counter with an M of 27. Controlling a servo arm movement from 0° to 245° in 1° increments would require an 8-bit counter with an M of 10.

The LPM_ADD_SUB is a binary adder, using the LPM_COUNTER and LPM_FF outputs as the “A” and “B” inputs. The LPM_ADD_SUB carry input (CI) is a constant “0,” the LPM_ADD_SUB SUM output is not used, and the PWM output (PWM_OUT) is the LPM_ADD_SUB CO. The following three examples explain how this simple structure works for a 4-bit PWM.

- When LPM_FF sample is “1,” there are 16 additions made as the LPM_COUNTER sequences from 0 to 15. The CO of the adder is “0” for 15 of 16 counts, 0 to 14, and “1” when the count is 15, thus generating a 1/16% duty cycle.
- When LPM_FF sample is “8,” the CO of the adder is “0” for counts 0 to 7, and “1” for the eight remaining counts, 8 to 15, thus generating an 8/16% duty cycle.
- When LPM_FF sample is “15,” the CO of the adder is “0” for only the 0 count, and “1” for the 15 remaining counts, 1 to 15, thus generating a 15/16% duty cycle.

The PWM is very resource efficient. PWMs with a common sample period and resolution can share a single LPM_COUNTER, and a non- 2^N resolution does not affect efficiency. Table 1 shows the LEs required as a function of bit width.

Table 1. PWM LE Requirements

Block	First PWM*	Additional PWM (1)
LPM_FF	N	N
LPM_COUNTER	N	0
LPM_ADD_SUB	N	N
TOTAL	$3N$	$2N$

Note:

(1) $N = \log_2(\text{Resolution})$

Even on the smallest MAX IIZ CPLD, the EPM240Z, with 240 LEs, one 8-bit PWM only requires one-tenth of the MAX IIZ resources. Ten 8-bit PWMs require only 168/240 of the MAX IIZ resources.

Conclusion

The PWM is a very powerful tool for converting digital data to analog actions. A MAX IIZ CPLD with PWMs can control all of the analog elements of a system, including light, sound, and motion. The MAX IIZ architecture is well suited to making simple, logic-efficient PWMs, so using the PWM outputs to control analog components like LEDs, speakers, motors, and servos requires only a few passive components.

Further Information

- “How do digital servos work?”:
www.servocity.com/html/how_do_servos_work_.html
- Nyquist frequency:
http://en.wikipedia.org/wiki/Nyquist_frequency
- Sampling Frequency:
http://en.wikipedia.org/wiki/Sampling_frequency
- Literature: MAX II Devices:
 - www.altera.com/literature/lit-max2.jsp
 - *Reduce Total System Cost in Portable Applications Using Zero-Power CPLDs:*
www.altera.com/literature/wp/wp-01001-reduce-total-system-cost-in-portable-apps-using-max.pdf
 - *Six Ways to Replace a Microcontroller With a CPLD:*
www.altera.com/literature/wp/wp-01041-six-ways-to-replace-microcontroller-with-cpld.pdf
 - *Using Zero-Power CPLDs to Substantially Lower Power Consumption in Portable Applications:*
www.altera.com/literature/wp/wp-01042-using-zero-power-cplds-to-lower-power-in-portable.pdf
 - *Using LEDs as Light-Level Sensors and Emitters:*
www.altera.com/literature/wp/wp-01076-led-driver-reduces-power-adjusting-intensity-ambient-light.pdf
 - *Developing Multipoint Touch-Screens and Panels With CPLDs:*
www.altera.com/literature/wp/wp-01086-multipoint-touchscreens-panels-cplds.pdf

Acknowledgements

- Rafael Camarota, Non-Volatile Product Line Manager, Low-Cost Products Group, Altera Corporation
- Judd Heape, Sr. Technical Marketing Manager, Consumer and Automotive Business Unit, Altera Corporation



101 Innovation Drive
San Jose, CA 95134
www.altera.com

Copyright © 2008 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.