Technical notes on using Analog Devices DSPs, processors and development tools
Contact our technical support at dsp.support@analog.com and at dsptools.support@analog.com
Or visit our on-line resources http://www.analog.com/ee-notes and http://www.analog.com/processors

PWM and Class-D Amplifiers with ADSP-BF535 Blackfin® Processors

Contributed by Aseem Vasudev Prabhugaonkar

Rev 1 – September 29, 2004

Introduction

This application note explains the basics of class-D amplifiers and their implementation on Blackfin® processors. The discussed class-D implementation has been achieved by pulse-width modulation using the processor's on-chip PWM timers. This portion of class-D (PWM) code has been integrated with ADSP-BF535 EZ-KIT Lite™ "C" talk-through example provided with the VisualDSP++™ 3.0 examples. This application note gives an overall understanding of class-D technology, emphasizes its advantages, and demonstrates how class-D can be implemented on Blackfin derivatives.

Class-D Amplifier Fundamentals

Class-D amplifiers are sometimes said to stand for digital amplifiers, but this is not correct. In fact, class-D operation is based on analog principles. The standard classes of analog amplifiers are A, B, AB, and C. The class of an amplifier is identified on the basis of transistor's operating point, also known as quiescent point of the transistor. The transistor's operating point is the point on DC load line in output transistor characteristics. Transistor operating point denotes a specific value of collector current "Ic" for a given base current. Hence, the position of operating point on the load line depends on transistor biasing. The idea to migrate toward higher power amplifier classes like AB and C is to improve the amplifier efficiency in terms of power drawn from the DC power supply. In addition, this improved efficiency reduces the heat sink requirements for amplifiers. Thus, this is a very important advantage in portable battery-operated handheld devices. But the efficiencies achieved with class C are still around 70 percent. This is where class-D technology plays a very important role in audio amplifier designs. In class-D amplifiers, the transistors used in the output stage (power stage) operate as switches. The transistors operate either in the cutoff region or in the saturation region so that the current through the transistors is very low (ideally zero when cutoff) or the voltage across the transistors is very low (ideally zero when transistors are in saturation). This reduces the amount of power drawn from the power supply and hence increases the power efficiency of the amplifier; it also helps to design amplifiers with smaller heat sinks.

Advantage of Class-D in DSPs

The Class-D concept is very advantageous and appealing in a DSP audio system. In today's sound systems, which include low-cost portable audio systems like MP3, WMA players are designed with DSPs (digital signal processors) or ASICs. These systems are targeted for better performance and also for lower manufacturing costs. Battery-operated systems especially must be highly power efficient. The lower cost requirement and the higher power efficiency is well taken care by the class-D amplifier designs and with its advanced implementation techniques, the audio quality is improved to a larger extent.

The DSP systems usually use codecs or DACs (digital-to-analog converters) to perform D-to-A conversion of digital audio data output by the DSP. With class D, the system can be designed without a codec or a D-to-A converter, thus reducing the overall system cost. In addition, the amplifier power efficiency is also improved to a large extent, which is highly appreciated in the portable audio market.

Basic Class-D Amplifier

A basic class-D amplifier in the analog domain consists of three primary units:

- A comparator to convert the analog input signal into PWM output
- An H-Bridge, which is the class-D power amplifier
- A filter at the output of the H-Bridge to reconstruct the analog signal

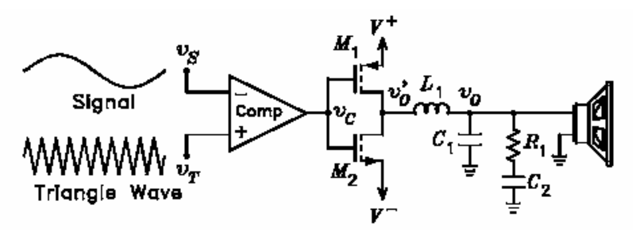

Figure 1. Class-D Amplifier – Analog Domain

In DSP audio systems, PWM generation techniques can be implemented in software to generate the PWM signal for the H-Bridge power amplifier. The major factors that govern the performance of such a system are:

- The algorithm used to generate PWM output corresponding to the digital audio signal
- The output analog filter stage used to reconstruct analog audio

Figure 2 shows an oscilloscope screen snapshot of PWM generated from comparator (blue) and filtered analog output (pink).

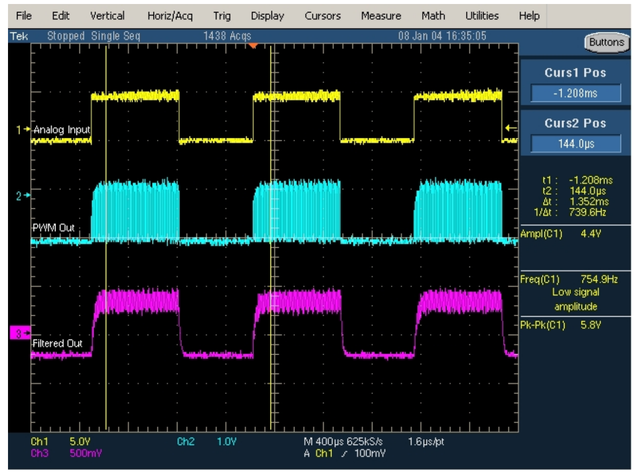

Figure 2. Input Signal, PWM and Filtered Output

Class-D Implementation on the ADSP-BF535 EZ-KIT Lite

In an effort to implement class D with ADSP-BF535 processors, timers are used in PWM mode to generate a PWM signal corresponding to the digital audio signal. Certain modifications are carried out in the ADSP-BF535 EZ-KIT Lite talk-through code for this demonstration. The audio samples are read by the processor over the serial port. The amplitude of the audio signal controls the pulse width of PWM signal. This effectively generates a PWM signal analogous to the audio input. The PWM left and right channels are driven on the `TMR0` and `TMR1` signal pins, respectively (refer the *ADSP-BF535 EZ-KIT Lite User's Manual* [1]). The generated PWM is then fed to an "LC" low-pass filter, which reconstructs the audio signal. The "LC" low-pass filter should have a cut-off frequency "fc" of about 20 KHz.

Limitations of the ADSP-BF535 PWM Implementation

The PWM implementation in this case is achieved using the PWM timers of the ADSP-BF535 processor. This imposes limitations on the maximum PWM frequency that can be used and also on the dynamic range of the audio signal. These limitations are due to the fact that the

`TWIDTH` value cannot be more than the `TPERIOD` value. Thus, the appropriate value of these two parameters is a tradeoff between audio signal resolution and the PWM frequency.

Class-D Setup with the ADSP-BF535 EZ-KIT Lite Board

Figure 3 shows the complete setup with the ADSP-BF535 EZ-KIT Lite board. An analog audio signal is fed to the codec on the board. The PWM signals for the left and right audio channels are generated on the `TMR0` and `TMR1` pins, respectively.

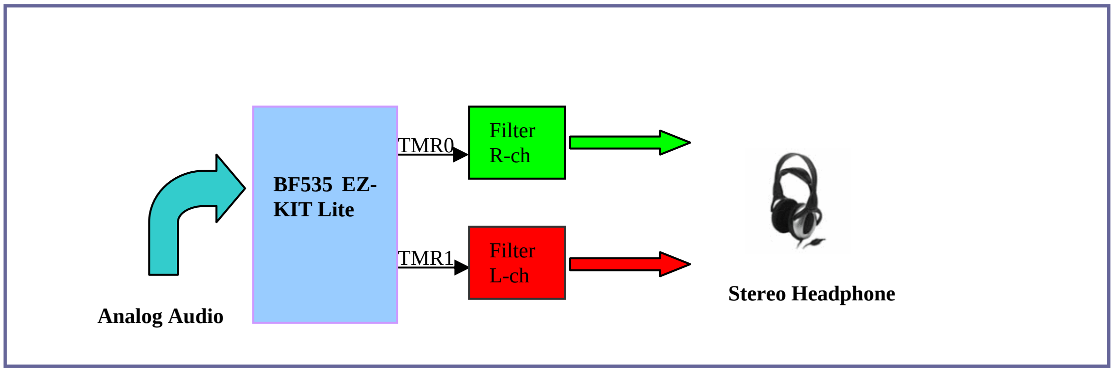

Figure 3. Class-D Setup with the ADSP-BF535 EZ-KIT Lite Board

Appendix

The PWM code is integrated with talk-through code. The entire project files are available in a zip file. Please see the portion of code, which converts digital audio samples to PWM and drives the `TMRx` pins below.

Code Snippet

```
/* ***************************************************************************
 *
 * Copyright (c) 2003 Analog Devices Inc.  All rights reserved. Please note that
 * below given is not complete code but just a snippet of code. For complete code
 * please see the zip file attached.
 *
 * ***************************************************************************/

#include "BF535 Talkthrough.h"
#include <cdefBF535.h>
#include <signal.h>

//snippet of main function that initializes PWM timers
//***************************************************************************/
void Init_PWM0(void);
void Init_PWM1(void);

void main(void)
{
    Init_PWM0();
    Init_PWM1();
    Init_Flags();
    Init_SPORT0();
    Init_Interrupts();
    Init_Codec();

    while(1)
    {
         if(sNew_Sample_Received) pSPORT0_RX_ISR_Handling();
    }
}

void Init_PWM0(void)
{
    *pTIMER0_CONFIG = 0x0019;
    *pTIMER0_PERIOD_HI = 0x0000;
    *pTIMER0_WIDTH_HI = 0x0000;
    *pTIMER0_PERIOD_LO = 0x142;
    *pTIMER0_WIDTH_LO = 0xA1;
    *pTIMER0_STATUS = 0x0100;
}

void Init_PWM1(void)
{
    *pTIMER1_CONFIG = 0x0019;
    *pTIMER1_PERIOD_HI = 0x0000;
    *pTIMER1_WIDTH_HI = 0x0000;
```

```
    *pTIMER1_PERIOD_LO = 0x142;
    *pTIMER1_WIDTH_LO = 0xA1;
    *pTIMER1_STATUS = 0x0400;
}

//*********************************************************************************/

// The following is a snippet of code which processes the audio samples. The timer
// width is modified depending on the amplitude of the input signal. The audio
// signal needs sufficient clamping prior to the modification of PWM width. This is
// done in order to ensure that the "TWIDTH" is a non-negative and non-zero value.

void Process_Audio_Data(void)
{
    unsigned int temp_sLeft_Channel_In;
    unsigned int temp_sRight_Channel_In;
    //------------------------------------------------------------------------------
    // code for data processing should be placed here
    sAC97_Tag_Out       = sAC97_Tag_In;
    sLeft_Channel_Out   = sLeft_Channel_In;
    sRight_Channel_Out  = sRight_Channel_In;

    //=====================================================
    //Process PWM
    temp_sLeft_Channel_In = sLeft_Channel_In + 0x7fff;//clamp signal
    temp_sLeft_Channel_In = temp_sLeft_Channel_In >> 8; //scale input
    *pTIMER0_WIDTH_LO = temp_sLeft_Channel_In;

    temp_sRight_Channel_In = sRight_Channel_In + 0x7fff;//clamp signal
    temp_sRight_Channel_In = temp_sRight_Channel_In >> 8; //scale input
    *pTIMER1_WIDTH_LO = temp_sRight_Channel_In;

    // confirm SPORT0 RX interrupt handling
    sNew_Sample_Received = 0;
}
```

Listing 1. Code Snippet

References

[1] *ADSP-BF535 EZ-KIT Lite Evaluation System Manual.* Revision 2.1, April 2003. Analog Devices, Inc.

[2] *ADSP-BF535 talk-through code supplied with VisualDSP++ 3.0*, Analog Devices, Inc.

[3] *ADSP-BF535 Blackfin Processor Hardware Reference*, Revision 2.0, April 2003, Analog Devices, Inc.

Document History

Revision	**Description**
Rev 1 – September 29, 2004 *by Aseem V. Prabhugaonkar*	Initial Release