Hybrid Linearized Class-BD Double Sided (LBDD) Digital Pulse Width Modulator (DPWM) for Class-BD Audio Amplifiers

S. W. Kuta[a],*, W. Kołodziejski, J. Jasielski

[a] *State Higher Vocational School in Tarnów, Mickiewicza 8, 33-100 Tarnów, Poland*

Article history:
Received 1 December 2017
Received in revised form
22 December 2017
Accepted 23 December 2017
Available online 27 December 2017

Abstract

The paper presents an original architecture and implementation of 9-bit LBDD hybrid DPWM circuit for Class-BD digital audio amplifier. The input PCM signals are directly transformed into 24-bit LBDD DPWM signals and then are requanized to the 9-bit digital outputs using noise-shaping process to support high fidelity with practical values of time resolution, and finally are converted by the DTCs into the two physical trains of 1-bit PWM signals. The architecture of the proposed Class-BD hybrid DPWM circuit is composed of two Class-AD ones. The hybrid quantizer converts 6 MSB bits using counter method, based on the STM32F407xx microcontroller, while the remaining 3 LSB bits – using a method based on the Programmable Tapped Delay Line (PTDL). All necessary time waveforms are generated on the base of the internal microcontroller oscillator 168 MHz. The proposed 9-bit Class-DB DPWM circuit allows to attain SNR of 110 dB and THD about 0,2% within the audio baseband, at switching frequency of 328.1 kHz, clock frequency of 42 MHz and modulation index M = 0.95. Basic verification of algorithm and circuit operation as well as simulation and preliminary experimental results have been performed.

Key words: Class-D Digital Audio Amplifier, Digital Pulse Width Modulator (DPWM), Linearized Pulse Width Modulation (LPWM), Linearized Class-AD Double sided (LADD), Linearized Class-BD Double sided (LBDD), Programmable Tapped Delay Line (PTDL), Analog Delay Locked Loop (ADLL), Digital to Time Converter (DTC)

Introduction

Implementation of the Class-D concept in digital audio systems provides an opportunity to remain the audio data in the digital domain through the amplification process, and convert these data to the analog domain only at the output stage.

Fig.1 shows a comparison of the analog Class D amplifier to the digital one [1, 2]. Both amplifiers are controlled by the digital input source. An analog Class-D amplifier used for digital system has to include high precision DAC to convert the digital input audio data to the analog signal, which is then transformed into NPWM signal to control the output power stage.

The H-bridge output stage of the Class-BD amplifier is switching by the two Natural sampling Class-BD Double-sided (NBDD) signals, generated by the NBDD modulator as a comparison results of the original and inverted audio signals to the triangle waveform. With reference to Class-AD amplifier, effective sampling frequency for Class-BD one is doubled, without increasing the transition frequency on the output. In terms of differential output, NBDD is superior, and has by far the most attractive spectral characteristics for all the other NPWM methods, and is also the most attractive PWM scheme for Class-D audio amplifiers implementation. Unfortunately, the NBDD has a high level of CM signal on the output terminals, therefore to eliminate this signal, the output filter of the Class-BD amplifier has to be more complicated [3]. Digital Pulse Width Modulator (DPWM) used in digital Class-D amplifier converts directly N-bit digital pulse-code modulation (PCM) input signal into a stream of 1-bit modulated pulses.

Two basic modulation techniques may be used, either DPWM methods based on the best possible emulation of the fundamental NPWM schemes in the digital domain, or multi-bit Sigma-Delta (Σ-Δ) Modulation (Fig. 1b).

The PCM signal could be transformed into a standard DPWM signal using directly uniform PWM (UPWM) [3-6]. Unfortunately, the fundamental problem of the UPWM process is its inherent nonlinearity. This nonlinearity may be reduced by increasing the sampling rate of the digital input signal (interpolation), and application of precompensation linearization algorithms.

The conversion of PCM-UPWM by precompensation is well established and various methods exist [3–7]. Using enhanced sampling methods as Pseudo Natural Pulse Width Modulation (PNPWM) or Linearized Pulse Width Modulation (LPWM), the precompensation methods attempt to emulate NPWM in the best possible way, such that modulator linearity is no longer an obstacle in DPWM systems.

*Corresponding author: st.kuta@pwsztar.edu.pl

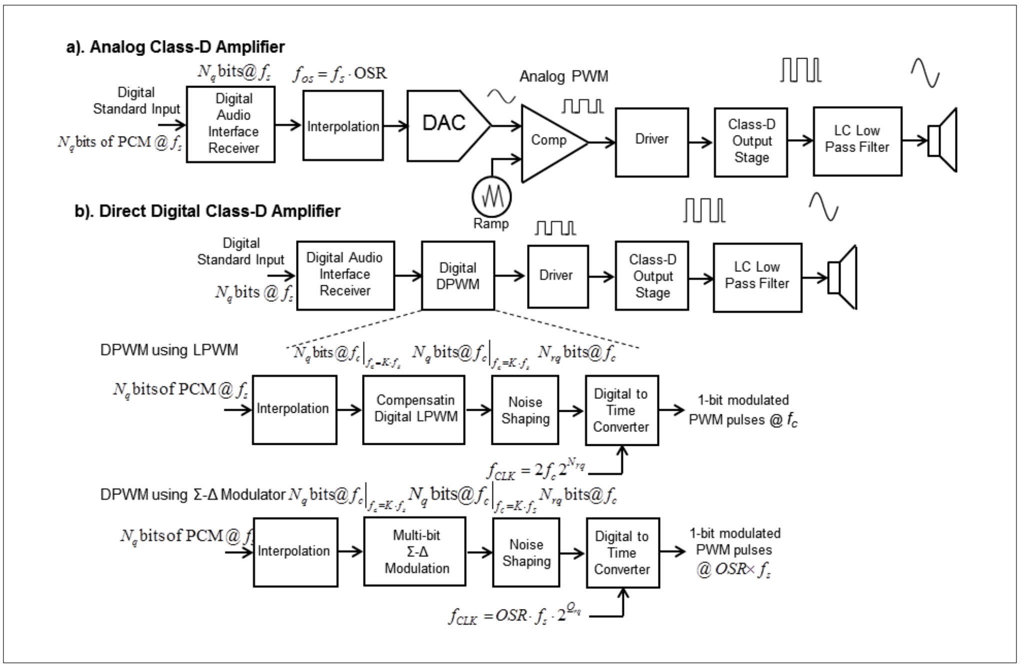

a). Analog Class-D Amplifier
Digital Standard Input
N_q bits of PCM @ f_s
N_q bits@ f_s
Digital Audio Interface Receiver
$f_{OS} = f_s \cdot$ OSR
Interpolation
DAC
Analog PWM
Comp
Ramp
Driver
Class-D Output Stage
LC Low Pass Filter
b). Direct Digital Class-D Amplifier
Digital Standard Input
N_q bits @ f_s
Digital Audio Interface Receiver
Digital DPWM
Driver
Class-D Output Stage
LC Low Pass Filter
DPWM using LPWM
N_q bits@ $f_c|_{f_c=K\cdot f_s}$
N_q bits@ $f_c|_{f_c=K\cdot f_s}$
N_{rq} bits@ f_c
N_q bits of PCM @ f_s
Interpolation
Compensatin Digital LPWM
Noise Shaping
Digital to Time Converter
1-bit modulated PWM pulses @ f_c
$f_{CLK} = 2 f_c 2^{N_{rq}}$
DPWM using Σ-Δ Modulator
N_q bits@ $f_c|_{f_c=K\cdot f_s}$
N_q bits@ $f_c|_{f_c=K\cdot f_s}$
N_{rq} bits@ f_c
N_q bits of PCM @ f_s
Interpolation
Multi-bit Σ-Δ Modulation
Noise Shaping
Digital to Time Converter
1-bit modulated PWM pulses @ $OSR \times f_s$
$f_{CLK} = OSR \cdot f_s \cdot 2^{Q_{rq}}$

Figure 1. Comparison of analog (a) and direct digital (b) Class-D amplifiers

Similarly as NBDD among four fundamental NPWM schemes, as well Linearized Class-BD Double-sided (LBDD) modulation which emulates NBDD scheme is superior among all the other digital LPWM modulations (in terms of differential digital output).

The calculated N_q -bit positions of the leading and trailing edges of the PWM pulse within each *n*th period ($T_c = 1/f_c$) of the output stage switching frequency has to be requantized to a lower resolution N_{rq}-bit digital output using noise-shaping process to support high fidelity with practical values of time resolution. The truncation of the digital N_q-bit input data to N_{rq}-bits results in higher quantization noise floor in the bandwidth (BW), i.e. lower Signal to Quantization Noise Ratio (SQNR). The requantizing process increases the SQNR within the audio baseband and moves the quantization noise power to an unused part of the bandwidth created by oversampling [4–10].

The obtained from the noise shaping process N_{rq}-bit positions of the leading and trailing edges of the PWM pulse within each *n*-th switching period T_c are finally converted by digital-to-time converter (DTC) into a physical stream of 1-bit modulated PWM pulses to control the output power transistors.

Practically, even for a requantized resolution N_{rq}-bit of the digital output, a required clock frequency f_{CLK} is too high to implement the DCT, therefore to circumvent the high-frequency clock problem a hybrid DCT is used [11–17], which integrates the counter method and a method based on the tapped delay line.

The topology of the DPWM modulator based on a Σ-Δ modulation [1, 2, 8–10] (Fig.1) is quite similar to that one employing standard DPWM. The algorithmic transformation of the digital PCM signal to PWM signal is put into the closed Σ-Δ modulator loop thus the noise and distortion generated during the transformation are suppressed by the high gain of the forward loop of the Σ-Δ modulator. Similarly to standard DPWM, the Σ-Δ noise-sharper quantizes the output of the algorithmic PWM from N_q - to N_{rq}-bit.

The interpolation factor allows a trade-off between modulator linearity, dynamic range and factors relating to the power conversion as efficiency and power stage linearity. This trade-off can be easier achieved for amplifiers based on Σ-Δ modulator than for ones employing standard DPWM [1, 2, 4–10]. In order to obtain satisfactory dynamic range, switching frequency should be increased to at least 384 kHz, to obtain a dynamic range of around 106 dB.

In the paper an original architecture and implementation of 9-bit hybrid DPWM circuit using LBDD modulator for Class-BD digital audio amplifier have been proposed.

LBDD modulator is composed of two linearized Class-AD double sided (LADD) ones, one of them converts the direct audio signal, while the second one converts the inverted audio signal. The signals and subsystems converting at the same time direct and inverted audio signals will be indexed in the paper by the letters L, R (for example: LADD_L, LADD_R).

Paper Organization

A linearized LBDD algorithm is presented in Section 2, system clock selection and architecture of the proposed 9-bit LBDD modulator are presented in Section 3, whereas implementation details of the used blocks are presented in Section 4. The simulated parameters and characteristics, as well basic verification of circuit manufacturability are shown in Section 4. The overall conclusions are given in Section 5.

Materials and Methods

Measurements have been performed using two-channel accurate Agilent Infinium Oscilloscope with WindowsXP operating system with maximum frequency of 500MHz and 1GS/s sampling. The LBDD algorithm has been implemented using STM32F4 Discovery circuit containing a CortexM4 microcontroller. The following microcontroller peripherals have been used: 12-bit ADC converter, TIM1 and TIM3 timers. To achieve the system converting 3-LSB (2:0) bits of the data, the quantizer system based on 3-Bit Programmable Delay Modules PLDM7 Series FAST/TTL Logic with 7 Delay Steps and 7 ns Inherent Delay has been implemented.

Linearized LBDD Algorithm

The LBDD algorithm, which emulates the NBDD modulation scheme, has to be enough simply to provide real-time calculation of the DPWM signals, because the positions of the leading and the trailing edges of the modulated pulses within each *n*-th switching period T_c have to be calculated as well for original audio signal as for inverted one. The used in our design LBDD algorithm is computationally simple, does not lead to excessive increase in total computational complexity, and is suitable for real-time calculations.

The locations of the leading edge $t_p(n)$ and the trailing edge $t_k(n)$ are calculated digitally at resolution of 24-bits through LPWM algorithm [2, 4–6]. Interpolating of the PCM audio data stream, sampled at f_s frequency, increases the original sampling rate of a sequence to a higher rate (*OVR*- times, where *OVR* is the interpolation factor) with a target switching frequency $f_c = OVR \cdot f_s = 1/T_c$. Further interpolation, followed by the first one, generates Q intermediate samples between two adjacent ones, with the distance: $T_Q = T_c/(Q+1)$, as it is shown in Fig. 2 for odd $Q = 5$. Using these two adjacent and new samples we receive piecewise linear approximation of the digital audio input signal.

Solving the crossing points between the triangular carrier signal and the approximated signal we obtain the PWM pulse-edge location of the leading edge $t_p(n)$ and the trailing edge $t_k(n)$ within each *n*-th switching period T_c .

The leading edge locations $t_p(n)$ of the PWM pulses, for arbitrary *n*, determines equation:

$$\bar{t}_p(n) = \frac{T_c\left[1 + i\, y_{i+1}(n) - (i+1) y_i(n)\right]}{4 + \left[y_{i+1}(n) - y_i(n)\right](Q+1)} \quad \text{and} \quad t_p(n) = nT_c + \bar{t}_p(n) \tag{1}$$

where *i*-th range between $y_i(n)$ and $y_{i+1}(n)$ samples should be find using a successive comparison method:

$$y_i(n) < S_{f,i} \ \text{ and } \ y_{i+1}(n) > S_{f,i+1} \Rightarrow \text{ for odd } Q:\ 0 \le i < \frac{Q-1}{2} + 1,$$
$$\text{for even } Q:\ 0 \le i \le \frac{Q}{2} + 1 \tag{2}$$

The trailing edge locations $t_k(n)$ of the PWM pulses, for arbitrary *n*, determines equation:

$$t_k(n) = nT_c + \frac{T_c\left\lfloor j\, y_{j+1}(n) - (j+1) y_j(n) - 3\right\rfloor}{\left\lfloor y_{j+1}(n) - y_j(n)\right\rfloor(Q+1) - 4} \tag{3}$$

where *j*-th range between $y_j(n)$ and $y_{j+1}(n)$ samples should be find using a successive comparison method:

$$y_j(n) < S_{f,j} \ \text{ and } \ y_{j+1}(n) > S_{f,j+1} \Rightarrow \tag{4}$$
$$\text{for odd } Q:\ \frac{Q-1}{2} + 1 \le j < Q+1, \ \text{ for even } Q:\ \frac{Q}{2} + 1 \le j \le Q+1$$

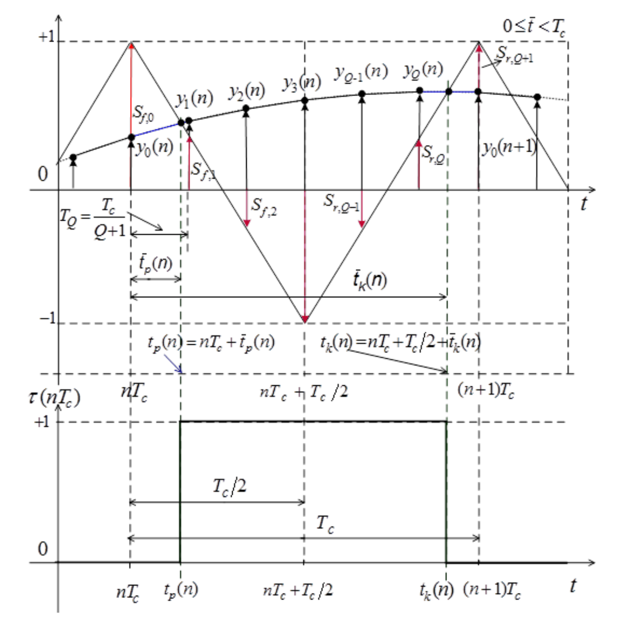


Figure 2. Piecewise linear approximation of the digital audio input for odd Q = 5.

On the base relationships (1–4) we can calculate LBDD time domain waveforms, and then the frequency spectrum of the LBDD modulator, using Matlab® tools. Fig. 3 shows the THD characteristics of the LBDD digital output versus frequency ratio $f_r = f_m/f_c$, for different value Q = *0, 1, 3, 5,* at: modulation index M = 0.95, switching frequency $f_c = 352.8$ kHz, where f_m is frequency of the modulating audio signal.

Practically, approximation beyond Q = 3 is not rational, because THD is almost the same for $Q > 3$. For Q in the range: 1–3 we receive a very simple computationally algorithm providing easy-going real-time calculation of the DPWM signals, ensuring at the same time low THD level: less than 0.09% for $Q = 1$, and less than 0,037% for $Q = 3$ within the audio baseband and modulation index M = 0,95.

The real-time calculations of the pulse-edge locations should be carried out within each *n*-th period of the switching frequency, using polyphase filtering for interpolation, both for the original audio signal: $t_p^L(n)$, $t_k^L(n)$ and for the inverted one: $t_p^R(n)$, $t_k^R(n)$ (for the left and right part of the LBDD modulator, respectively).

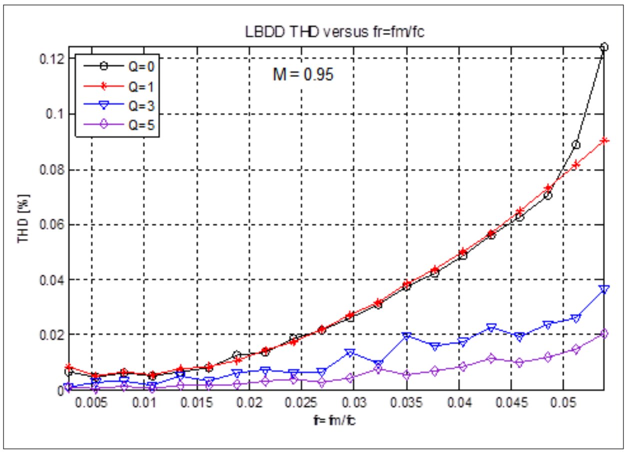


Figure 3. THD of the LBDD digital output versus $f_r = f_m/f_c$ for different Q values

Noise shaping

The finely calculated N_q-bit pulse-edge locations have to be requantized to a lower resolution N_{rq}-bit digital output using noise-shaping process [4,6,7]. To simplify the analysis of the noise shaping system, the quantizer can be modeled as an linear network with an independent additive noise source $e_q(n)$, as it is shown in Fig. 4 . It is assumed, that the error $e_q(n)$ is a stationary white noise sequence uncorrelated with the signal sequence $x(n)$ and is uniformly distributed over the range: $-\Delta/2 < e_q(n) < \Delta/2$, where $\Delta = R/2^{N_q+1}$ is the step of the quantizer, R represents the amplitude of the quantizer ($R = T_c/2$ for double-sided PWM) and N_q+1 represents the word length.

This noise shaping model is one of the simplest, but assures the necessary dynamic range D of the requantized resolution N_{rq} for high quality Class-D amplifier, at the same time, the real-time requantizing processes (for original and inverted signals) not require a significant computing power.

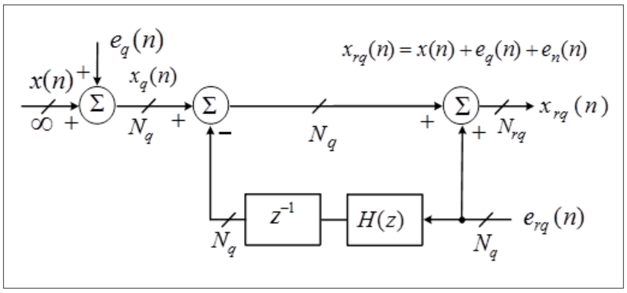


Figure 4. Model of noise shaping architecture

The digital input $x_q(n)$ (calculated N_q-bit pulse-edge locations: $t_p^L(n)$, $t_p^L(n)$, $t_p^R(n)$, $t_k^R(n)$) and quantization error on input $e_q(n)$ are transmitted to the output in unchanged forms. Requantizing noise source e_{rq} *(n)* (generating after the truncation of the calculated N_q-bit digital data to N_{rq} bit resolution) is transferred to the output as the output requantized error noise $e_n(n)$, which depending of the filter transmittance $z^{-1}H(z)$ is suppressed within the audio BW and gained above audio BW.

On the basic model in Fig. 4, we can obtain the following z-domain relationship between the output requantized error noise and the requantizing noise source, defined as noise transfer function, $H_{NTF}(z)$, that may be interpreted as a filter:

$$H_{NTF}(z) = \frac{E_n(z)}{E_q(z)} = 1 - H(z)z^{-1}, \qquad (5)$$

Very effective suppression of the requantizing noise source e_{rq} *(n)* within the audio BW ensures m-th order noise transfer function $H_{NTF}(z)$ with integer coefficients [4-5,7]:

$$H_{NTF}(z) = 1 - H(z)z^{-1} = \left(\frac{z-1}{z}\right)^m, \qquad (6)$$

Both, (5) and (6), are integer valued Finite Impulse Response (FIR) filters, and the order of (5) is smaller than order of (6) by one.

Simulation of the noise shaping process using fifth order filter and 9-bit quantizer, applied to the 24-bit digital input at switching frequency $f_c = 352{,}8$ kHz shows, that this process suppress noise source e_{rq} *(n)* up to - 60dB in the worst case scenario, for the corner frequency 20 kHz of the audio BW.

System Clock Selection and Architecture of the Proposed 9-bit LBDD Modulator

The implementation of LBDD PWM modulator based on STM32F407xx microcontroller has to solve three fundamental problems:

The applied linearization algorithm LBDD PWM should emulate NBDD PWM in the best possible way to receive comparable performance.

All algorithms should be computationally simple and suitable to real-time calculations.

The clock selection and timing of all signal processing blocks have to be based on the maximum frequency of the three microcontroller AHB buses of 168 MHz [18–20].

The essential limitation of the digital-to-time converter (DTC), which converts obtained from the noise shaping process N_{rq}-bit positions of the leading and trailing edges of the PWM pulse within each n-th switching period into a physical sequence of 1-bit modulated PWM pulses is excessive clock speed. The DTC with N_{rq}-bit resolution, based on counter method in down-counting mode, requires clock frequency [15–17]:

$$f_{CLK} = f_c \times 2^{N_{rq}+1} \qquad (7)$$

giving time resolution of generated pulse:

$$\Delta t_{\min} = T_{CLK}/2 \qquad (8)$$

Even for moderate resolution (exceeding 8 bits), simple counter method needs excessive clock speeds.

Most often, to moderate an excessive requirement of high clock frequency, instead of simple counter method, the hybrid method is used [14–16]. The hybrid quantizer converts the data of most significant bits (MSB) $[(N_{rq}-1):m]$ using counter method, while the remaining part of least significant bits (LSB) $[(m-1):0]$ – using a quantizer system based on the programmable tapped delay line (PTDL).

Hybrid N_{rq}-bit LADD modulator, processing the data of $(N_{rq}-m)$ MSB bits on the base counter method, requires clock frequency f_{CLKh} [5-6, 14-16]:

$$f_{CLKh}=\frac{f_{CLK}}{2^m}=f_c\times 2^{(N_{rq}-m)+1}=OVR\times f_s\times 2^{(N_{rq}-m)+1} \qquad (9)$$

The quantizer processing the data of LSB (2:0) bits comprises a cascade of 2^m identical delay segments, at the same time the delay introduced by single segment equals [12–17]:

$$t_{d1}=T_{CLKh}/2^m=T_{CLK} \qquad (10)$$

Consequently to assure N_{rq}-bit resolution, DTC processing error must be smaller than $(t_{d1}/2)$.

Quantizer based on the tapped delay line circumvents the high-frequency clock problem, however, the practical realization may be cumbersome if feasible at all. Asymmetrical layout of large number $N=(2^{N_{rq}}-1)$ delay cells, additional and variable delay introduced by each path connecting delay line tap with appropriate multiplexer's input as well some other factors like Process, Voltage and Temperature (PVT) variations and different mismatches result in nonlinearity of the quantizer. However linear characteristic with evenly distributed steps can be achieved for low resolution of the converter, when delay line is formed of small number of the delay cells, closed in feedback loop of the DLL [12–16].

Integration of both afore-described methods into a hybrid DTC allows a trade-off between the high-frequency clock requirement and quantization linearity of the digital to time conversion [13–16].

Each of the two digital-to-time converters (DTC^L an DCT^R respectively) converts $N_{rq}-m=6$ MSB (8:3) bits of the data using advanced-control timer TIM1 or TIM8 of the microcontroller, while the remaining $m=3$ LSB (2:0) bits of the data – using a quantizer system based on the 3-bit PTDL.

The clock selection and timing of all signal processing blocks are based on the maximum frequency of the three microcontroller AHB buses: f_{osc} = 168 MHz [18–20].

According to the equations (9, 10), the clock frequency of the high-speed APB2 domains (connecting utilized peripheral devices such as TIM1, TIM8, ADC and also the clock frequency of the external quantizer system based on the 3-LSB (2:0) PTDL, has been set at:

$$f_{PCLK2}=f_{CLKh}=\frac{f_{osc}}{P_1}=\frac{168\text{ MHz}}{4}=42\text{ MHz} \qquad (11)$$

$$T_{CLKh}=2^m T_{CLK}=\frac{1}{f_{CLKh}}=\frac{1}{42\text{ MHz}}\approx 23{,}81\text{ ns}$$

The clock for the embedded 12-bit analog-to-digital converter (DAC) is the same as the APB2 clock.

$$f_{ADCCLK}=f_{PCLK2}=42\text{ MHz} \qquad (12)$$

The total DAC conversion time T_{CONV} [18] has been set at the same value as the switching period T_c, in accordance with equations (7, 9) with OVR=1, to avoid interpolation of the PCM audio data stream sampled at the frequency f_{CONV} and to save significantly computation time.

The peripheral DAC the STM32F407xx microcontroller system has 16 multiplexed channels and it is possible to organize the conversions in two groups: regular and injected [18]. According to equation (9), to receive $T_{CONV}=128T_{CLKh}=1024T_{CLK}$, a regular group of fife conversions has been composed. The regular channels and their order in the conversion sequence have been selected in the ADC_SQRx registers and the total number of conversions in the regular group of the L[3:0] bits has been written in the ADC_SQR1 register.

$$T_{CONV}=\sum_{1}^{G}\text{Sampling time}+12\text{ cycles}=4\times 15T_{CLKh}+68T_{CLKh}=128T_{CLKh} \qquad (13)$$

So, the employed DAC generates the PCM audio data stream with frequency:

$$f_{CONV}=f_c=\frac{f_{CLK}}{2^{N_{rq}+1}}=\frac{f_{CLKh}}{2^{(N_{rq}-m)+1}}=\frac{336\text{ MHz}}{2^{10}}=\frac{42\text{ MHz}}{2^7}=328{,}125\text{ kHz}$$

and:

$$T_{CONV}=T_c=2^{N_{rq}+1}T_{CLK}=1024\frac{1}{f_{CLK}}=\frac{1024}{336\text{ MHz}}\approx 3{,}04762\ \mu\text{s}$$

$$T_{CLK}=\frac{1}{f_{CLK}}=\frac{T_c}{2^{N_{rq}+1}}=\frac{1}{2^m f_{CLKh}}=\frac{1}{2^3\times 42\text{MHz}}=2{,}9762\text{ ns}$$

Fig. 5 presents the architecture of the circuit converting 3-LSB (2:0) bits of the data - using a quantizer system based on the 3-bit PTDL (a) and time domain waveforms of the control and output signals (b).

Peripheral 12-bit successive approximation analog-to-digital converter (ADC) [18–20] samples an input audio signal with the sampling frequency $f_s=f_{CONV}\cong 328{,}125\text{ kHz}$ and convert it into the two PCM sample series for positive and inverted audio signal. These PCM sequences are used to calculate the data of the leading- and trailing- edge locations $t_p(n)$ and $t_k(n)$ respectively of the PWM signals.

The interrupt handler of the Converter ADC1 is called with sampling time $T_{CONV} = 3.0476$ (ADC1->CR2 = ADC_CR2_EXTEN_0) by timer TIM3, generating square wave at frequency f_{TIM3} = 328.125kHz and duty factor 0.5.

To generate this square wave, the configuration of the timer TIM3 - Init_TIM3() is folowing:

```
   void Init _ TIM3()
   {
   TIM3->CR1 = TIM _ CR1 _ URS;
        TIM3->CR2 = TIM _ CR2 _ MMS _ 1;
        TIM3->SMCR = TIM _ SMCR _ SMS &0x6;
        TIM3->DIER      = 0x0000;
        TIM3->CCMR1 |= TIM _ CCMR1 _ OC1M _ 2| TIM _
 CCMR1 _ OC1M _ 1;
        TIM3->CCMR1 |= TIM _ CCMR1 _ OC1PE | TIM _
 CCMR1 _ OC2PE;
        TIM3->CCMR2     = 0x0000;

        TIM3->CCER = 0x0001;

        TIM3->PSC = 0x0001;

        TIM3->ARR = 0x007F;
        TIM3->CNT = 0x0000;

        TIM3->CCR1 = 0x003F;

        TIM3->CCR2 = 0x0000;

        TIM3->CCR3 = 0x0000;

        TIM3->CCR4 = 0x0000;

        TIM3->DCR = 0x0000;
                TIM3->DMAR = 0x0000;
        TIM3->EGR = TIM _ EGR _ UG;
                TIM3->CR1  |=  TIM _ CR1 _ ARPE  |
 TIM _ CR1 _ CEN;
   }
```

Table 1.

Setting timer registers TIM3: ARR=0x007F and PSC=0x0001, the counter of the timer TIM3 counts up from 0 to 127 clock cycles (1/42MHz=23,81ns).

The BUS APB1 frequency (84 MHz) is divided by 2 (PSC=0x0001) to get o the timing clock frequency of TIM3 (42MHz). Generating by timer TIM3 signal at frequency of f_{TIM3} = 328.125kHz is then output by the first channel CH1 on pin 4 of port B, PB4.

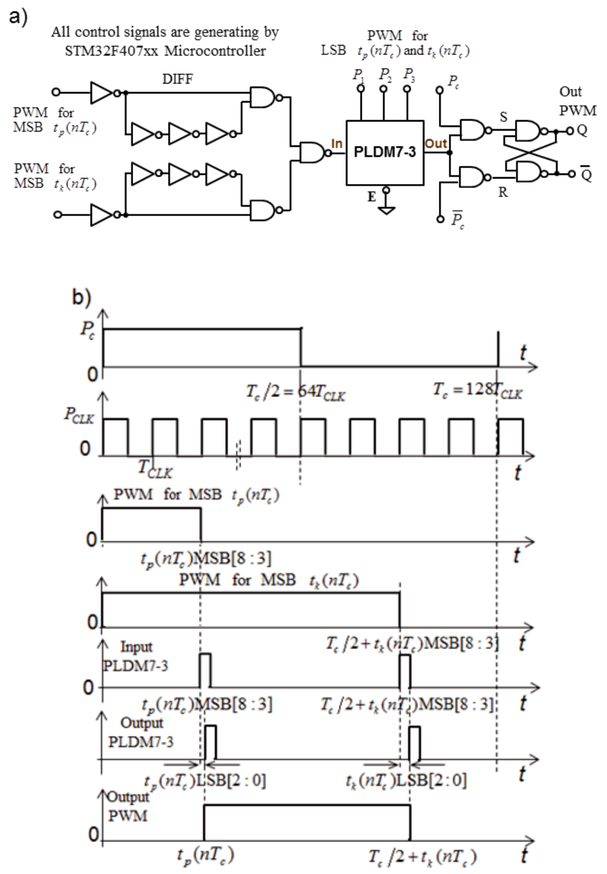


Figure 5. Architecture of the circuit converting 3-LSB (2:0) bits of the data - using a quantizer system based on the 3-bit PTDL. (a) and time domain waveforms of the control and output signals (b)

The interrupt handler of the ADC1 is following:

```
   void ADC _ IRQHandler(void)
   {
   if(ADC1->SR & ADC _ SR _ EOC)
   {
   valADC = (int)ADC1->DR;
   val _ U _ ADC _ sim = (2.94 * valADC/4096.0-VT);
   tab _ l _ sim[1] = tab _ l _ sim[0];
   tab _ l _ sim[0] = val _ U _ ADC _ sim;
   if(tab _ l _ sim[1]!=0)
   {
   valTp _ sim = ( (STALA) *( VT - tab _ l _ sim[0])) /
 (4*VT + (tab _ l _ sim[1] - tab _ l _ sim[0]));valTk _
 sim = ( (STALA) *(3*VT + tab _ l _ sim[0])) / (4*VT
 - (tab _ l _ sim[1] - tab _ l _ sim[0]));
   }
   if(nl==2){nl = 0;}
   }
   ADC1->SR = 0x0000;
   }
```

Table 2.

Fig. 6 shows a piecewise linear approximation of the PCM input signal and the leading edge $t_p(n)$ and trailing edge $t_k(n)$ locations of the PWM pulses over one sampling period $T_{CONV} = T_c$. There are also marked locations of the update events UEV$_1$ – UEV$_5$ corresponding to the leading – and trailing locations of the PWM pulses.

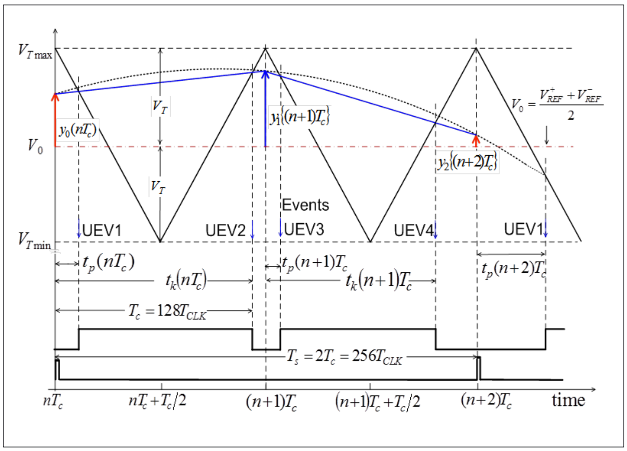


Figure 6. Piecewise linear approximation of the digital audio input together with marked locations of update events UEV1−UEV.

According to equations (1-4), for $Q = 1$, we calculate the pulse-edge locations $t_p(n)$ and $t_k(n)$ over one switching periods

$$\bar{t}_p(nT_c) = \left.\frac{[1 - y_0(n)]}{4 + [y_1(n+1) - y_0(n)]} \times 1024 \times T_{CLK}\right|_{V_T = 1\,\mathrm{V}} \tag{14}$$

$$\bar{t}_k(nT_c) = \left.\frac{[3 + y_0(n)]}{4 + [y_0(n) - y_1(n+1)]} \times 1024 \times T_{CLK}\right|_{V_T = 1\,\mathrm{V}} \tag{15}$$

$$\bar{t}_p(\{n+1\}T_c) = \left.\frac{[1 - y_1(n+1)]}{4 + [y_2(n+2) - y_1(n+1)]} \times 1024 \times T_{CLK}\right|_{V_T = 1\,\mathrm{V}} \tag{16}$$

$$\bar{t}_k(\{n+1\}T_c) = \left.\frac{[3 + y_1(n+1)]}{4 + [y_1(n+1) - y_2(n+2)]} \times 1024 \times T_{CLK}\right|_{V_T = 1\,\mathrm{V}} \tag{17}$$

The calculated data are requanized to the lower 9-bit resolution, and then are divided into the two parts of 6 MSB(8:3) bits and 3 LSB(2:0) bits of data and sequentially are written into LADD$_L$ LADD$_R$ registers.

Using 6 MSB(8:3) bits of the calculated values of the leading- and trailing- edge locations $t_p(n)$ and $t_k(n)$ of the PWM signals, firstly two digital PWM signals with a linearized Class-AD single sided (LASS) DPWM impulses are generated in each *n*-period of the switching frequency T_c – according to described algorithm (Fig. 5b). The beginnings of both pulses start at the beginnings of the switching frequency periods and end after the time of the leading edge locations $t_p(n)$ for the first signal, and trailing edge locations $t_k(n)$ for the second one. To generate these waveforms, timer TIM1 has been used.

Configuration of the timer TIM3 – Init_TIM3() is folowing:

```
  void Init _ TIM1(){
  TIM1->CR1 = TIM _ CR1 _ ARPE | TIM _ CR1 _ URS;
TIM1->CR2 = TIM _ CR2 _ MMS _ 1;
  TIM1->SMCR = 0x0000;
  TIM1->CCMR1=TIM _ CCMR1 _ OC1M _ 2|TIM _ CCMR1 _
OC1M _ 1|TIM _ CCMR1 _ OC2M _ 2|
       TIM _ CCMR1 _ OC2M _ 1;
  TIM1->CCMR1 |= TIM _ CCMR1 _ OC1PE | TIM _ CCMR1 _
OC2PE;
  TIM1->CCER  =  TIM _ CCER _ CC1E  |  TIM _ CCER _
CC2E;
  TIM1->PSC = 0x0003;
  TIM1->ARR = 0x007F;
   TIM1->CNT = 0x0000;
   TIM1->RCR = 0x0000;
   TIM1->CCR1 = 0x003F;
  TIM1->CCR2 = 0x003F;
  TIM1->BDTR = TIM _ BDTR _ MOE | TIM _ BDTR _ OSSR |
TIM _ BDTR _ OSSI;
   TIM1->DIER = TIM _ DIER _ UIE;
   TIM1->EGR |= TIM _ EGR _ TG;
   TIM1->CR1 |= TIM _ CR1 _ CEN;
  TIM _ ITConfig(TIM1,TIM _ IT _ CC1|TIM _ IT _ CC2,
ENABLE);
   NVIC _ EnableIRQ(TIM1 _ CC _ IRQn);
   }
```

Table 3.

Differentiating the trailing edge locations at the time $t_p(n)$ of the first signal and the trailing edge locations at the time $t_k(n)$ of the second one, we can generate digital PWM signals with a linearized Class-AD double sided (LADS) DPWM impulses, as shown in Fig. 5b.

Passing narrow pulses from the outputs of both differentiators through a programmable tapped delay line (PTDL), we receive the pulses delayed in time. The phase delay times between the input and the output pulses of the differentiators shall be determined by the 3 LSB(2:0) bits of the calculated values of the leading- and trailing- edge locations $t_p(n)$ and $t_k(n)$ of the PWM signals. This requires to program twice the PTDL in each switching period nT_c. At the beginning of the period nT_c , we have to enter 3 LSB(2:0) bits of the leading- edge locations $t_p(n)$ at the programming input of the PTDL and hold these data by the first half of the period $T_c/2$, while at the beginning of the second half of the period $T_c/2$, we have to enter 3 LSB(2:0) bits of the trailing- edge locations $t_k(n)$ and hold these data by the second half of the period $T_c/2$.

As it is shown in Fig. 5a, output flip-flop is setting by delated pulses associated with 6-bit leading- edge locations $t_p(n)$, whereas it is resetting by delated pulses associated with 6-bit trailing- edge locations $t_k(n)$ of the PWM signals, therefore on the flip-flop output final 9-bit LBDD DPWM signal is generating.

Results and Discussion

Fig. 7 shows time waveforms and frequency spectrum of the LBDD DPWM signals that have been generated using Matlab® tools, according to the described in the paper algorithm. Peripheral 12-bit successive approximation analog-to-digital converter (ADC) [18–20] samples an input audio signal with the sampling frequency equal to the switching frequency $f_s = f_{CONV} \cong 328{,}125$ kHz (in applied algorithm interpolation factor $Q = 0$). Analyzing the presented zoomed view of noise-shaped 9-bit LBDD DPWM output, we see, that applied LBDD algorithm (for $Q = 0$) to transform the PCM audio signal into the noise shaped 9-bit DPWM data suppresses all unwanted spectral components and allows to attain the noise floor below −100 dB within the baseband.

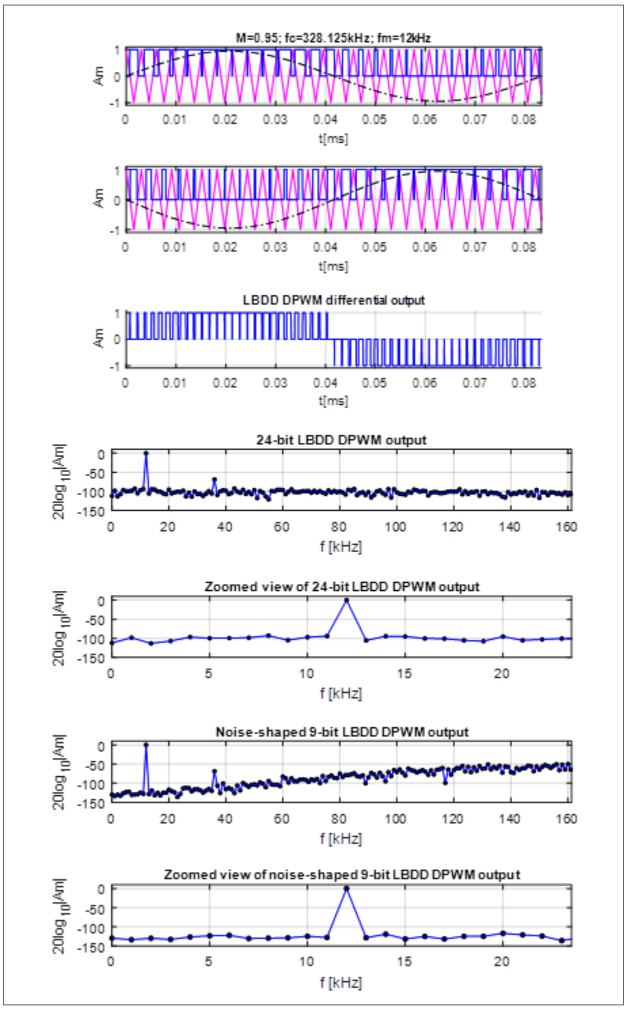


Figure 7. Comparison between spectral content of the 24-bit DPWM output and spectral content of the noise shaped 9-bit DPWM output at: $M = 0.95$, $f_c =$ 328,1 kHz, $f_m = 12$ kHz for: a) $Q = 0$, b) $Q = 1$.

This algorithm is not complicated and does not require significant computing power of the processor. 9-bit noise shaped requantizing process lowers noise floor within the baseband about (25–30)dB. Further reduction of noise floor, below −130 dB within the baseband, we can receive for $Q = 1$.

Fig. 8 presents oscillograms of the time domain waveforms of the selected control and output signals, generating by the proposed 9-bit LBDD modulator.

The oscillograms in the Fig. 8 illustrate sequentially:

Two digital PWM impulses, generated by the peripheral timer TIM1of the microcontroller on the basis of 6 MSB(8:3) bits of the calculated values of the leading- and trailing- edge locations $t_k(n)$ and $t_p(n)$. The beginnings of both pulses start at the beginnings of the switching frequency periods and end after the time of the leading edge locations $t_p(n)$ for the first signal, and trailing edge locations $t_k(n)$ for the second one.

Spikes generated by the ADDL subsystem presented in Fig. 5a, obtained from the differentiation falling edges of the generated by TIM1pulses on the basis of 6 MSB(8:3) bits. The lower part of the waveform presents these spikes on the output of the programmable tapped delay line (PTDL). The phase delay times between the input and the output pulses of the PTDL are determined by the 3 LSB(2:0) bits of the calculated values of the leading- and trailing- edge locations $t_p(n)$ and $t_k(n)$ of the PWM signals.

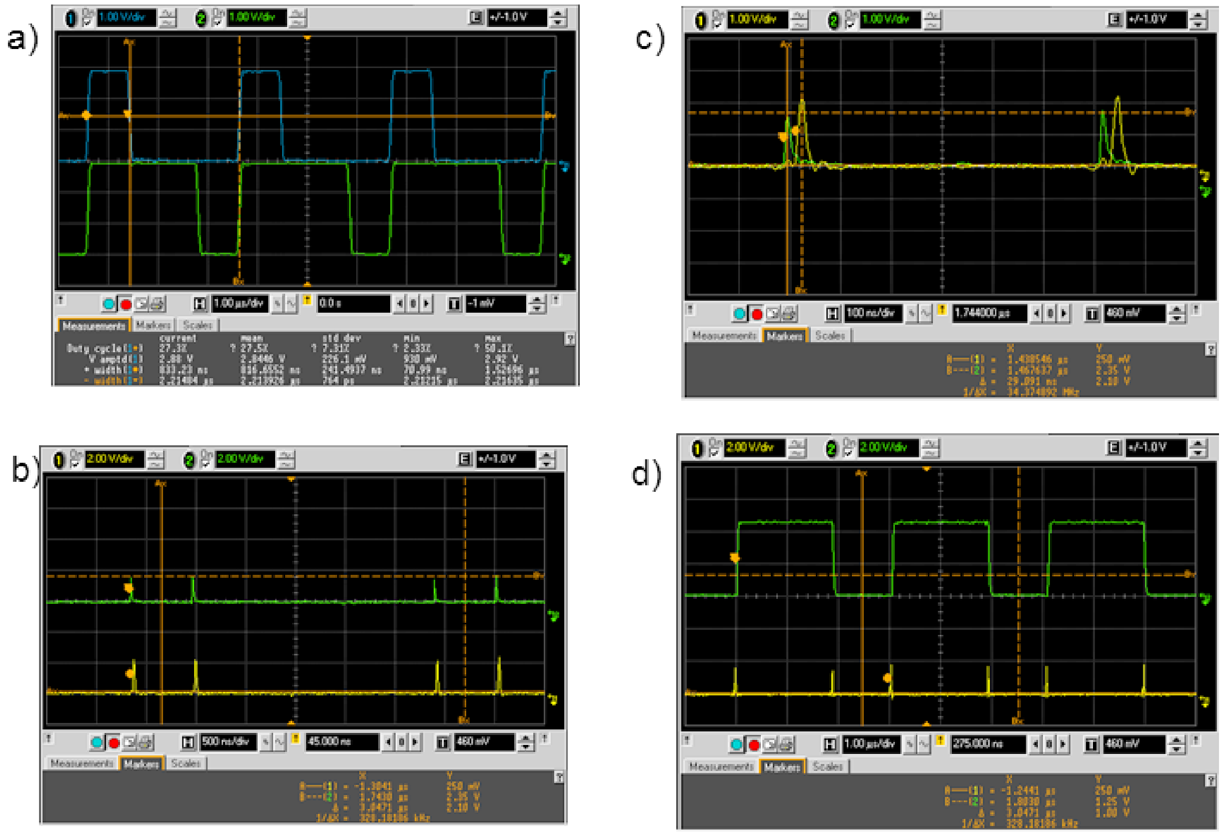
Zoomed view of the phase delay time between the input and the output pulse of the PTDL (the delay time in oscillogram is about 29 ns).

Final 9-bit LBDD DPWM signal generating on the output of the RS flip-flop.

The LSB bits of the data, transferred to the PORT E of the STM32F4 Discovery system and addressing multiplexer lines of the PTDL, are perfectly synchronized, moreover, the delay times caused by signal path transferring the pulses from the outputs of both differentiators, through the PTDL until the output of RS flip-flop are approximately balanced, thus the processing errors are very small, and high resolution of the PWM modulator is possible to achieve.

Conclusions

A novel architecture of the proposed Class-BD hybrid DPWM modulator is composed of two Class-AD ones. The hybrid quantizer converts 6 MSB bits using counter method, based on the STM32F407xx microcontroller, while the remaining 3 LSB bits – using a method based on the Programmable Tapped Delay Line (PTDL). Computationally very simple and suitable for real-time calculations LBDD algorithm and noise-shaping process have been used to calculate 9-bit DPWM outputs for original and inverted audio input signals. Noise-shaping process using fifth order filter and 9-bit quantizer, applied to the calculated



	current	mean	std dev	min	max
Duty cycle(1)	27.3%	? 27.5%	? 7.31%	? 2.33%	? 50.1%
V ampt d(1)	2.88 V	2.8446 V	226.1 mV	930 mV	2.92 V
+ width(1)	833.23 ns	816.6552 ns	241.4937 ns	70.99 ns	1.52696 µs
- width(1)	2.21484 µs	2.213926 µs	764 ps	2.21215 µs	2.21635 µs

```
              X                 Y
A—(1) : -1.3041 µs        250 mV
B---(2) : 1.7430 µs        2.35 V
    Δ : 3.0471 µs          2.10 V
 1/ΔX : 328.18186 kHz
```

```
              X                 Y
A—(1) : 1.438546 µs       250 mV
B---(2) : 1.467637 µs      2.35 V
    Δ : 29.091 ns          2.10 V
 1/ΔX : 34.374892 MHz
```

```
              X                 Y
A—(1) : -1.2441 µs        250 mV
B---(2) : 1.8030 µs        1.25 V
    Δ : 3.0471 µs          1.00 V
 1/ΔX : 328.18186 kHz
```

Figure 8. Oscillograms of the time domain waveforms of the selected control and output signals, generating by the proposed 9-bit LBDD modulator.

24-bit LPWM output at switching frequency f_c = 328.1 kHz and relatively low clock frequency of f_{CLK} = 42 MHz, allows to attain SNR of 110 dB and THD of the output signals less than about 0,2 % within the audio baseband at modulation index M = 0.98.

The detailed algorithm converting 6 MSB bits of the data on the base counter method using advanced-control timers TIM1 and TIM3 of the STM32F407VGT6 microcontroller, and quite original quantizer to convert the remaining 3 LSB bits of the data have been described. Original architecture and relatively simply implementation of the hybrid digital to time converters are characterized by excellent synchronization between all buildings blocks and negligible signal paths delays, thus the processing errors are very small assuring high resolution of the converters.

Basic verification of algorithm and circuit operation as well as simulation and preliminary experimental results have been done. After complete successful hardware realization this solution seems to be very interesting for practical power amplifier applications.

References

1. M. Berkhout, L. Dooper, *IEEE Transactions On Circuits And Systems-I: Regular Papers*, 2010, **57**, 991–1002.
2. R. Cellier, G. Pillonnet, A. Nagari, N. Abouchi, "An Review of Fully Digital Audio Class D Amplifiers Topologies", Circuits and Systems and TAISA Conference, 2009. NEWCAS-TAISA '09. Joint IEEE North-East Workshop, June 28-July 1, 2009.
3. Ch. Morrison, S. Weiss, M. Macleod and R. Stewart, *Comparison of Single- and Double-Sided Pulse Width Modulated Signalswith Non-Linear Predistortion*, Proceedings of the 5th European DSP Education and Research Conference, 2012, Amsterdam.
4. S. M. Munk, K. S. Andersen, State of the Art Digital Pulse Modulated Amplifier System, AES 23rd Int. Conf, Copenhagen, Denmark, 2003 May 23–25.
5. C. Pascual, Z. Song, P. T. Krein, et. al., *IEEE Transactions on Power Electronics*, 2003, **18**, 474–485.
6. M. Johansen, K. Nielsen, *A Review and Comparison of Digital PWM Methods for Digital Pulse Modulation Systems*, 107th AES Convention , 1999 September 24–27, New York.
7. F. Guanziroli, R. Bassoli, C. Crippa, D. Devecchi, and G. Nicollini., *IEEE Journal Of Solid-State Circuits*, 2012, **47**, 686–698.
8. Luo Shumeng and Li Dongmei, *Journal of Semiconductors*, 2013, **34**, 115001-1–115001-6.
9. L. Dooper, M. Berkhout, *IEEE Journal Of Solid-State Circuits*, 2012, **47**, 1524–1534.
10. Jia-Ming Liu, Shih-Hsiung Chien, Tai-Haur Kuo, *IEEE Journal Of Solid-State Circuits*, 2012, **47**, 1344–1354.

11. B. H. Gwee, J. S. Chang, H. Li, *IEEE Transactions On Circuits And Systems -Ii: Analog and Digital Signal Processing*, 2002, **49**, 245–2256.

12. A. Syed, E. Ahmed, D. Maksimovic, E. Alarcon, *Proc. IEEE Power Electron. Spec. Conf.*, 2004, **6**, 4689–4695.

13. B. J. Patella, A. Prodic´, A. Zirger, D. Maksimovic´, *IEEE Transactions on Power Electronics*, 2003, **18**, 438–446.

14. Xuzhen Shen, Xiaobo Wu, and Jing Lu, Lin Qin, Proceedings of the International MultiConference of Engineers and Computer Scientists, 2010, Vol II, IMCS 2010, March 2010.

15. J. Jasielski, S. Kuta, W. Machowski, W. Kołodziejski, *Microelectronics Journal*, 2014, **45**, 1202–1211.

16. J. Jasielski, S. Kuta, W. Machowski, W. Kołodziejski, “Hybrid LBDD PWM Modulator for Digital Class-BD Audio Amplifier Based on STM32F407VGT6 Microcontroller and Analog DLL”, Mixed Design of Integrated Circuits & Systems (MIXDES), 2015 Proceedings of the 22st International Conference, 25–27 June 2015.

17. J. Jasielski, W. Kołodziejski, S. Kuta, W. Machowski, “An Analog Dual Delay Locked Loop Using Coarse and Fine Programmable Delay Elements”, Proceedings of the 20-th International Conference Mixed Design of Integrated Circuits & Systems (MIXDES), 2013, Gdynia, Poland, June, Pgs. 20–22.

18. RM0090 Reference manual. STM32F40xxx, STM32F41xxx, STM32F42xxx, STM32F43xxx advanced ARM-based 32-bit MCUs.

19. STM32F405xx STM32F407xx, Datasheet, June 2013, DocID022152 Rev 4.

20. UM1472 User manual Discovery kit for STM32F407/417 lines, January 2014 DocID022256 Rev 4.