Cent. Eur. J. Eng. • 3(4) • 2013 • 620-643
DOI: 10.2478/s13531-013-0118-5

VERSITA

Central European Journal of **Engineering**

ASIC and FPGA based DPWM architectures for single-phase and single-output DC-DC converter: a review

Review article

Subhash Chander[1]*, Pramod Agarwal[2], Indra Gupta[2]

1 *Govt. College of Engineering and Technology, Jammu (J&K) Jammu, Jammu and Kashmir, India*

2 *Electrical Engineering Department, Indian Institute of Technology, Roorkee-INDIA*

Received 22 September 2011; accepted 31 October 2012

Abstract: Pulse width modulation (PWM) has been widely used in power converter control. This paper presents a review of architectures of the Digital Pulse Width Modulators (DPWM) targeting digital control of switching DC-DC converters. An attempt is made to review the reported architectures with emphasis on the ASIC and FPGA implementations in single phase and single-output DC-DC converters. Recent architectures using FPGA's advanced resources for achieving the resolution higher than classical methods have also been discussed. The merits and demerits of different architectures, and their relative comparative performance, are also presented. The Authors intention is to uncover the groundwork and the related references through this review for the benefit of readers and researchers targeting different DPWM architectures for the DC-DC converters.

Keywords: DPWM architectures • DC-DC Converters • FPGA based DPWM • ASIC • DPWM • PWM

Nomenclature

ADC	Analog to Digital Converter
ASIC	Application Specific Integrated Circuits
Clk	Reference clock
$d[n]$	Duty cycle command/value
DCM	Digital Clock Manager
dexc	Executable duty cycle
DLL/PLL	Delay/Phase Locked Loop
DPWM	Digital Pulse width modulation/modulator
DSP	Digital Signal Processor
EMI	Electromagnetic Interference
f_{clk}	Clock frequency
FF-DPWM	Flip-Flop of DPWM
FPGA	Field Programmable Gate Array
f_s/T_s	Switching frequency/time
HSM/LSM	High/Low side MOSFET
LUTs	Look Up Tables
MUX	Multiplexer
n	No of bits
PWM	Pulse Width Modulator/Modulation
Seg i	Segment i
SIDO	Single Input Double Output
SMPS/SMPC	Switched Mode Power Supply/Converter
t_{ON}	Output Pulse width or ON-time of Switch
t_d	Cell delay

*E-mail: subrag11121@gmail.com

$V_{control}$	Control voltage of each segment
$V_{q,DPWM}$	Output quantization step of DPWM
Δd	Resolution measured as the minimum increment in ON-time step between two successive duty cycle values
Δt_{min}	DPWM time resolution
ZCS/ZVS	Zero current/voltage switching

1. Introduction

There has recently been an increased focus on research in digital control of switched mode power supplies (SMPS) due to their advantages over the analogue control [1–8]. However, the main disadvantages of digital control are the sampling delay and the limited resolution [9] due to constraints of ADC and the DPWM. The windowed ADC technique has helped to circumvent the issues pertaining to resolution of ADC [3, 10]. However, resolution of DPWM, which has to be higher than that of ADC to avoid limit cycling, remains a bottleneck [1, 9]. The use of constant-frequency pulse-width modulation is widespread in switching power converters, although a sigma-delta scheme for modulation [11–15] and varying switching frequency schemes, such as sliding-mode and hysteretic control [16–26], for control can also be used. The pulse width modulation (PWM) principle is widely used in power electronics applications for controlling power converters (DC-DC, DC-AC, etc.) [27]. In addition to hard-switching DC-DC converters, the PWM control strategy has also been applied in zero voltage (ZVS) and zero current switching (ZCS) resonant converters [28]. In many applications it is desirable to use a microcontroller, DSP or FPGA to switch a converter with the help of sophisticated control schemes such as fuzzy control [29–31], sliding mode control [16–21] etc. The block diagram of such a configuration is shown in Figure 1a. An application of DPWM control in a buck DC-DC converter is shown in Figure 1b. The duty cycle command is in digital form, which is processed by the DPWM to create a driving pulse to control the ON-time of the main switch S, in the power stage and modulate the DC input voltage into a high-frequency wave, V_{iD}, to produce DC output voltage (V_o) through a low pass L-C filter. Regulation of V_o to the desired value is achieved by adjusting the duty cycle command value as shown in Figure 1c. A high-resolution DPWM circuit is one of the critical blocks for successful practical realization of digital control for switching power supplies. Hence, developing DPWM architecture and techniques is an important part of the entire controller design. Today, DC-DC converters are operating at switching frequencies exceeding tens of MHz [32], resulting in a new set of challenges for research in the field of DPWMs.

Regardless of the architecture, all DPWM operate by first quantizing the switching period T_s into individual time slots. The total number of time slots depends on the required resolution of the DPWM. Once the switching period is quantized, digital logic is used to achieve the required duty ratio $d[n]$ through an appropriate time slot that sets and resets the output based on the input. Figure 2a and Figure 2b show the standard analog trailing-edge pulse width modulator and its associated waveform, in which control voltage is compared to a carrier wave to provide amplitude to time domain conversion to achieve a linear relationship between the control voltage and the output pulse width [33]. The implementation of a pulse width modulator with a digital signal $d[n]$ as the input requires the proper means to provide the digital to time domain conversion. Figure 2c and Figure 2d show a conceptual DPWM realization and associated waveforms [33]. In DPWM, time is quantized into a number of discrete time slots of length t_d and a particular slot is selected by the duty cycle command $d[n]$. The digital comparator, which is used to select the time slot in the DPWM, serves the purpose of the analog comparator in Figure 2a. There are many practical methods by which this time step has been generated, shown in Figure 2b [2, 3, 9, 10, 34–57], with different characteristics.

In the last few years many alternatives have been reported to increase the resolution of DPWMs. In the first group of architectures an attempt has been made to obtain the minimum time step in the DPWM by changing the hardware architecture of the DPWM. In the second group of architectures the effective duty cycle resolution has been increased using a software approach that changes the pattern used in the generation of the output signal i.e Digital dither [9, 51, 58, 59] and sigma-delta [11–13]. A new set of architectures has been presented to reduce the minimum time step of DPWM by using the FPGA's advance resources and hybrid architectures with software approach. A number of implementations for DPWM are possible with different characteristics in terms of high-frequency capability, complexity, area, power consumption, linearity, monotonicity etc.

The DPWM is the most critical component in the digitally controlled switching power converter. A literature reviews can showthat the single chip DC-DC converter circuits have been around 15 years; an early traceable contribution of Stratakos *et.al.* appeared in 1994 [60]. It was motivated by emerging battery operated applications that required a compact, lightweight and highly efficient

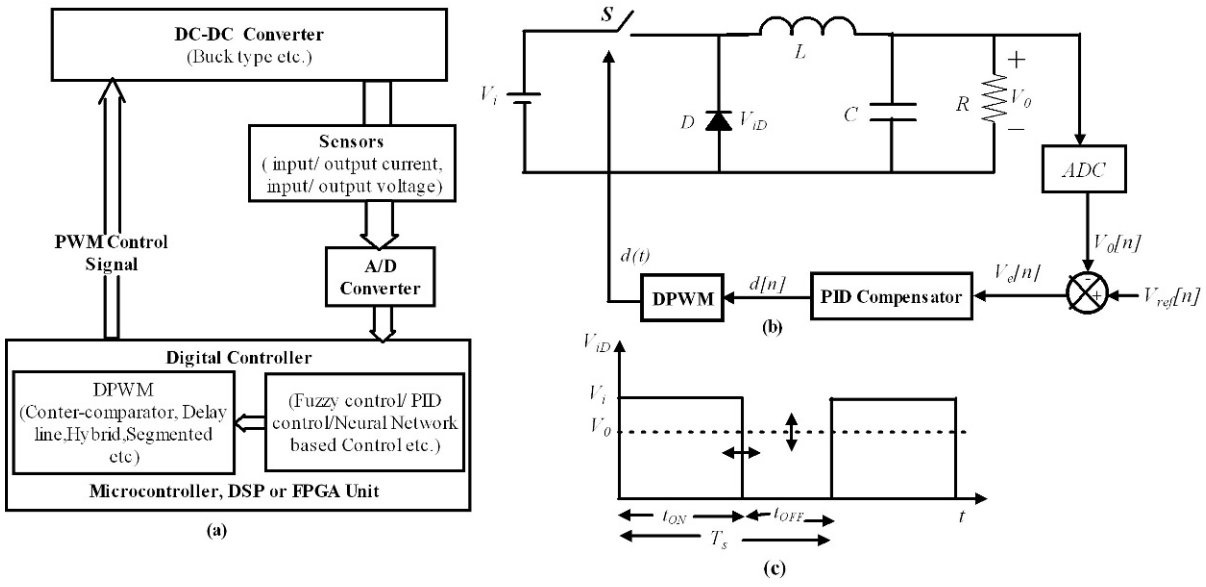

Figure 1. Implementation of digital PWM, DC-DC Converter control using a FPGA, DSP or Microcontroller unit (a) General block diagram (b) PWM control of buck converter with PID compensator and (c) associated waveform.

Buck converter IC chip. Later in 1995, Arbetter *et.al.*, presented an optimized design of DC-DC converters for energy limited battery operated systems [61]. Use of analog PWM schemes prevailed till 1997. Parallely, Dancy et.al [55], presented the design for a Digital PWM controller and the two architectures for DPWM using fast clocked counter and delay lines for IC implementation. Use of these architectures for a variable voltage DC-DC converter was presented in 1998 [56]. This started the development of IC implementation in DPWM architecture for single phase DC-DC converters. The first architecture on DPWM for multi-output dc-dc conversion for low voltage system was based on counter-comparator and delay line and appeared in 2000 [54]. Syed, Ahmed and Maksimovic had first presented the survey and classification of architectures for IC implementation of DPWM for DC-DC converter in 2004 [33] and presented first segmentation based DPWM to optimize the resources. Later, they introduced the input-voltage feed-forward compensation into the digital controller IC through a delay-line based DPWM [10] in 2005.While, DPWM modules for SMPC digital controller ICs were successfully designed and implemented as a part of Integrated circuits (ICs), some of the crucial issues, such as programmability of switching parameters like resolution, duty cycle, versatility in the PWM output waveforms e.g. dead times and switching frequencies beyond 1 MHz [57], remained unresolved. These issues were addressed by O'Malley and Rinne [57], through the implementation, in IC form, of highly versatile DPWM composed of delay-locked loop (DLL) and programmable DPWM module. It allowed the generation of high resolution, high switching frequency PWM signals with the DPWM IC support switching frequencies beyond 15 MHz. Further improvement in the resolution, over the classical methods, was provided by the low power consumption of the FPGA implementation, this allowed for high switching frequencies. First time FPGA based DPWM was reported in 2005 by Foley *et.al.* [38, 40]. The paper [38] describes the DPWM design for FPGA implementations and a novel multi-output PWM scheme with versatile output waveforms with high resolution over a small area. While, in [40], a single and 3-phase heterogeneous DPWM for FPGA based implementation was presented. Further increases in the resolution, for high frequency applications, were introduced through FPGA's advanced resources like DLL and Digital Clock Manager (DCM) in the architectures. Having increased the resolution without significantly increasing in the clock frequency, Huerta *et.al.* [49] proposed new DPWM architecture combining a synchronous block (counter based) and asynchronous block using available FPGA's DLL to achieve a fine time resolution of lower than 2ns. The clock multiplying and phase shifting properties of DLL were partially utilized in these architectures. Another architecture, which fully utilizes inherited phase shifting properties of the DCM blocks, simplifies the duty cycle generation capabilities as reported by Batarseh *et.al.* [46].

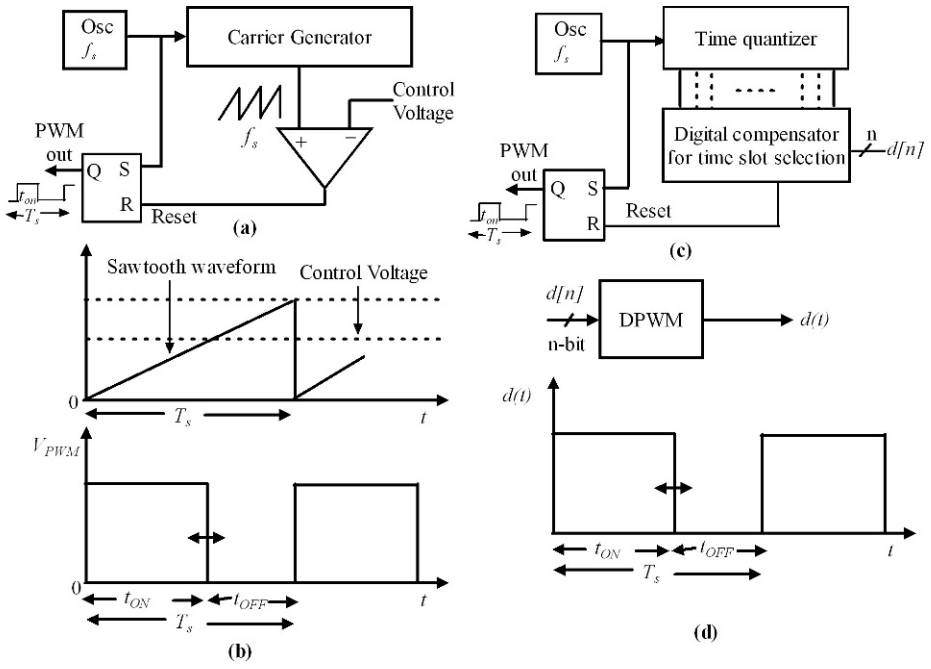

Figure 2. Block diagram of conceptual; (a) Analog PWM and its (b) associated waveforms (c) DPWM and its (d) associated waveforms.

It is a hybrid approach, combining the linear counter-comparator and the FPGA on-board DCM resources, offering reduced external clock frequency and thus, allowing high switching frequency operation. Further reducing the power dissipation in the architecture, Batarseh et.al [53] introduced the window mask concept in the architecture of [46]. A very high resolution DPWM was proposed by Castro and Todorovich in 2010 [52] which took advantage of advanced clock management capability called the fine phase shifting of the clock. This feature allows very small and programmable delays between input and output clocks, thereby increasing the resolution of DPWM to 19.5 ps in vertex-5 FPGA. This is the highest resolution reported by any of the architectures

2. Classical Digital PWM architectures

2.1. Counter based DPWM architectures

The simplest DPWM architecture, offering the best linearity, is the direct implementation of an analog PWM in the digital domain [37]. The simplest method to create a PWM signal from a duty command is to use fast-clocked counters. A clock frequency (f_{clk}) is chosen to be 2^n times the switching frequency (f_s), where n is the number of bits in the duty command. Figure 3a illustrates typical fast-counter digital PWM generation scheme as reported in [54–56], which uses a preset-able counter and zero-detector, recommended for low power switching power supplies [3, 54]. This method requires the use of very fast clocks. The architecture of Figure 3a is based on the principle that a counter is triggered by a clock signal (f_{clk}) with frequency equal to some multiple of the converter switching frequency (f_s).

$$f_{clk} = 2^n \cdot f_s \tag{1}$$

The PWM output signal is set at the beginning of the switching period and, simultaneously, the duty cycle command data is loaded in the preset able counter. The counter is then decremented on every clock cycle; the zero state is detected by the zero detector circuit. The zero state of the counter resets the SR flip flop after a number of clock cycles equal to the integer value of the $n-$bit input duty command.

A second type of implementation, separately reported in [38, 41], uses a cycling counter and a comparator, as shown in Figure 3b. It uses a cyclic counter operating at many times the required switching frequency. The output of SR-flip-flop is set when the counter value is zero and is reset when the counter reaches a chosen duty

cycle command value $d[n]$, thereby generating the required DPWM.

The third type of implementation [42] is shown in Figure 3c which uses n-bit free-running synchronous counter, $n-$bit register and comparator. The overflow signal of $n-$bit counter sets the output flip-flop and at the same time loads the duty cycle command value to the $n-$bit register. The $n-$bit register output is compared with the output value of an n-bit counter in an $n-$bit comparator. When these two values become equal, the comparator output is used to reset the SR flip flop output, thus producing the PWM wave. The counter overflow signal changes the duty cycle of new PWM wave period in order to avoid any frequency and/or phase jitter of the output waveform. The PWM frequency is given by (2), while its duty cycle is given by (3).

$$f_{PWM} = f_{clk}/2^n \tag{2}$$

$$d = \text{data value}/2^n \tag{3}$$

where, data value is the input data word integer value. If an 8-bit input is used, then the duty cycle is in the range $0 \leq d \leq 255/256 = 99.6\%$.

An excellent linearity is achieved through the use of a $2^n \cdot f_s$ clock to divide the time period $T_s = 1/f_s$ for an $n-$bit DPWM, which is required in order to maintain tight regulation of the output voltage with a feedback loop. The propagation delay of each increment is constant, provided the counter is designed properly, and is relatively easy due to the digital nature of the circuit. These arrangements have a simple construction and size [41, 62]. The architectures can be easily implemented either on ASIC or FPGA. However, the high switching frequency and high DPWM resolution may result in impractically large clock frequency requirement and thus large power consumption [53, 63], which turns out to be the disadvantage.

2.2. Tapped delay line based DPWM architectures

The tapped delay-lines, is an alternative to the counter based structure. In this structure, the propagation time of delay cells are used to resolve the pulse width modulated signal. It consists of 2^n delay cells that divide the switching period (T_s) into 2^n quantized time periods, a 2^n : 1 multiplexer, and a SR-latch. The fine time resolution and much reduced power consumption can be achieved using a tapped delay-line [3, 55].The delay cells in the delay line can also be designed to accomplish feed-forward compensation of the input voltage [10]. The delay lines architecture is the first selection when DPWM is implemented on ASIC [13]. The delay line scheme needs a very accurate delay component usually in the range of 20 ps to 200 ps, which is impractical in FPGA [13]. Though the tapped delay line based architecture circumvents the high-frequency clock problem of the counter-based DPWM [33, 55], the designs are susceptible to process and temperature variations.

This method uses the propagation delay of a pulse from a reference clock Clk, through a cascade connected chain of similar cells, to generate a time interval by selecting a given pulse width which is quantized as a function of the selected number of cells. The delay through a single cell is the fundamental time step, and the output of each cell is tapped into a multiplexer to facilitate its selection. This selection can be obtained from the $d[n]$ by means of a $2^n : 1$ multiplexer driven by the digital input code $d[n]$. The multiplexer selects the signal that resets the output PWM signal, thus performing the function of selecting one of the time slots generated by the delay-line time quantizer. The delay through the delay cell is the function of supply voltage (V_{dd}). This technique uses switching frequency as clock frequency; decreasing the power consumption at the expense of large chip size [5, 6, 63].

A delay-line may be arranged in two possible ways, either in open-loop [33, 55], where the line delay is designed to equal the switching period of an external clock (Figure 4a), or as a ring-oscillator (Figure 4b), where the delay-line sets the switching period [3, 33, 43, 64]. In open-loop delay line architecture, Figure 4a as reported in [33], the switching frequency (f_s) is imposed by an external oscillator, in which the total delay of the line should be designed in order for the maximum delay to match the switching period. As cell delay (t_d) varies with temperature because of semiconductor material properties, this condition cannot be satisfied at all corners. Consequently, the executed duty cycle ($dexc$) is not always the same as the duty cycle command $d[n]$ presented. As an alternative, in Figure 4b, the delay line can be used to form a ring oscillator which generates the clock at the switching frequency, imposed by the delay line. Thus the maximum duty cycle can be guaranteed. In this case, the process temperature variation may cause a drift in the switching frequency (f_s) but the $dexc$ is always equal to $d[n]$.

Arrangements similar to that in Figure 4a are used to implement input-voltage feed-forward compensation, by designing delay cells in the delay lines, shown in Figure 4c. The proposed approach in [10] to high-frequency, high-resolution DPWM introduced the feed forward compensation technique. Consider the basic delay-line DPWM configuration shown in Figure 1 of [55], which can be applied, to other DPWM realizations

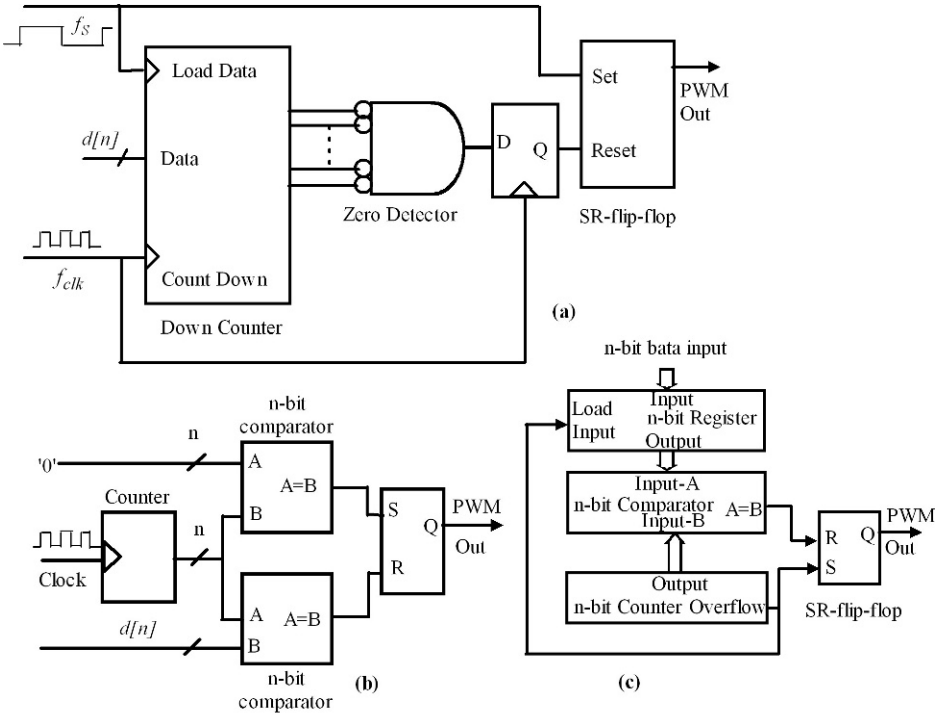

Figure 3. Counter-comparator type DPWM architectures using: (a) Presettable down counter and Zero-detector; (b) cycling counter and comparator; (c) Free running synchronous counter and comparator with data input resistor.

reported in [2, 3, 65]. At the start of a switching cycle, a clock signal sets the SR flip flop and starts the clock signal propagation through a chain of delay cells. When the signal reaches the delay cell selected by the duty-cycle command $d[n]$, the flip-flop is reset. The output pulse width t_{ON} is equal to

$$t_{ON} = k \cdot t_d \tag{4}$$

where, k is the numerical value of the duty command $d[n]$, and t_d is the cell delay. Feed-forward compensation is achieved for which the product of the converter input voltage V_{in} and the pulse width t_{ON} should be independent of the input voltage [10]. In case of delay line based DPWM, feed-forward compensation can be achieved if the cell delay t_d is inversely proportional to V_{in} as shown in (5),

$$t_d = A/V_{in} \tag{5}$$

where, A is ideally a constant. In a buck converter, the output quantization step V_{qDPWM} of the DPWM, i.e. the least-significant bit (LSB) value in volts, is given by (6), where, $T_s = 1/f_s$ is the switching period. FF-DPWM, has a constant quantization step, independent of the input voltage V_{in}.

$$V_{qDPWM} = V_{in} t_d / T_s \tag{6}$$

With the constant quantization step and the loop gain independent of the V_{in}, the limit-cycle conditions [9] are also independent of the V_{in} , which is another advantage of the feed forward DPWM. In the practical ASIC implementation of Figure 4c, it is assumed that the fclk. is constant, independent of the V_{in} or other operating conditions.

Another delay-line-based DPWM circuit, as reported in [54, 55], is shown in Figure 4d. The circuit is same as that in Figure 4a, with a delay matched network which is used to match the propagation delay experienced through the multiplexer. This eliminates any mismatched delay in the path of delay cell output to the reset terminal of the flip-flop.

These architectures use clock frequency as a switching frequency unlike in that used in counter based architectures, this helps to decrease the power consumption at the expense of large area due to area consuming multiplexer [3, 53, 55, 63]. The size of the multiplexer grows exponentially with the increase in the resolution bits (n). This makes

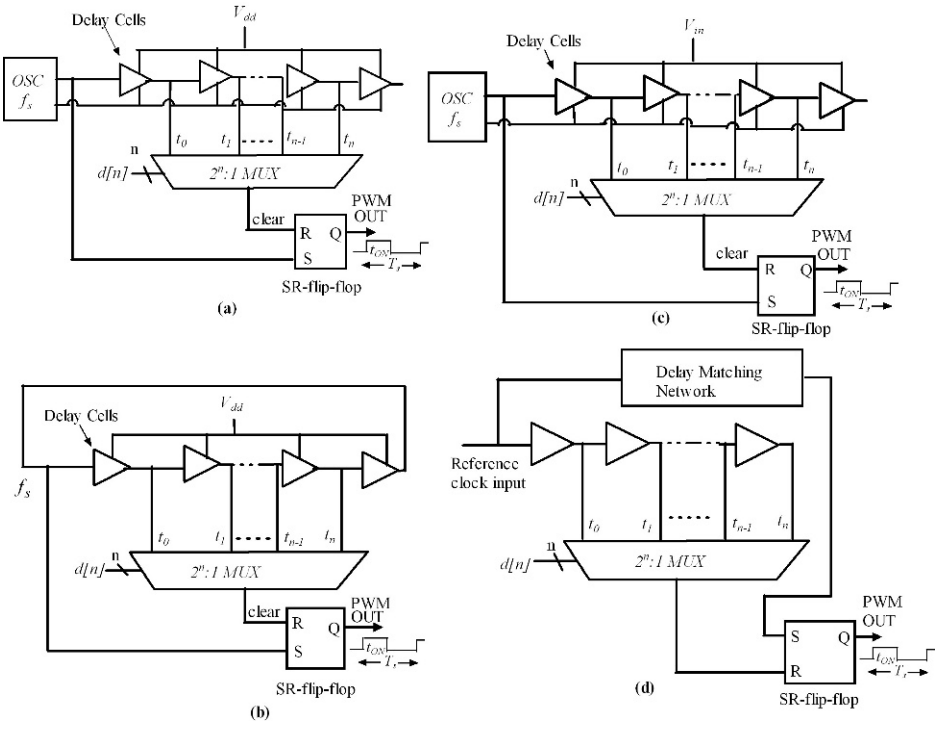

Figure 4. Trapped delay line based DPWM architectures; (a) Using Open-loop delay lines; (b) Closed loop delay lines forming a ring oscillator; (c) Feed forward DPWM architecture using open-loop delay lines; (d) Open-loop architecture with delay matching network.

this architecture unsuitable for high resolution DPWM. Device mismatching between different delay elements is the issue, which leads to the non-linearity in such architectures. These drawbacks can be mitigated using combination of delay-line and counter-comparator based hybrid scheme [54].

2.3. Hybrid approach based DPWM architectures

Keeping in mind the various disadvantages of the delay-line and the counter based DPWM; a compromise can be made between the two, resulting in Hybrid DPWM. In Hybrid DPWM, some of the bits are resolved using the delay lines, and the remaining using a counter [66]. Architectures of Figure 5a & Figure 5b provide a good solution to the area resolution trade-off, incorporating both a ring-oscillator delay-line and a counter, are detailed in [2, 40, 42, 54]. The hybrid approach uses either the counter with the MSB's to provide a coarse pulse width and the finer pulse width using the delay line selected by the LSB's, or vice versa. The addition of coarse and fine pulse widths is accomplished by a series connection of the two sub-circuits [33]. The high resolution is presented by the delay-line, the counter frequency is low. Hence, it has low power consumption. The long delay chains and large multiplexers are not required to set switching periods and therefore have a smaller area. A power and area efficient solution can be implemented by considering a hybrid counter-cum-delay-line based design, trading-off multiplexer area versus counter clock frequency [2, 54, 56, 57, 67, 68].

A Hybrid DPWM architecture reported in [2, 69] is similar to the design described in [54]. In this approach, an n—bit resolution is achieved using an n_c—bit counter, whereas the remaining $n_d = n - n_c$ bits of resolution are obtained from a tapped delay line. Figure 5a shows a simplified diagram of a hybrid DPWM where 4-bit ($n = 4$) resolution is obtained using a 2-bit counter (n_c) and a four-cell ring oscillator ($n_d = 2$, $2^n_d = 4$), which consists of resettable flip flops as delay cells. At the beginning of a switching cycle, the output SR flip flop is set, and the DPWM output pulse $d(t)$ goes high. $d(t)$ goes high. The pulse that propagates through the ring at the frequency ($2^{n_c} = 4 \cdot f_s$) serves as the clock for the counter. The complete switching period is divided into $2^{n_d} \cdot 2^{n_c} (= 16)$ slots. At the time when the counter output matches the top nc MSB's of the $d[n]$, a pulse reaches the tap selected by n_d, the

LSB's of $d[n]$, and the output flip-flop is reset making the output pulse low. The cell delay and the number of cells in the ring determine the switching frequency f_s. Figure 5a, achieves a 1 MHz PWM frequency with an 8-bit resolution using a 3-bit counter and a 32-cell long ring oscillator. Resolution equal to that achieved with 256 MHz clock, in counter based architecture is accomplished with the 8 MHz clock in hybrid architecture. However, both the 32-cell ring oscillator and the 32:1 multiplexer result in large implementation area.

A circuit similar to that in Figure 5a is reported in [42] as shown in Figure 5b. An $n-$bit PWM resolution is achieved using an n_c-bit counter and an n_d-bit delay-line, designed such that $n = n_c + n_d$. The delay-line has been built with D-type flip-flops comprising of a ring oscillator. The PWM output frequency, in this case, is also given by (2). The hybrid architecture shown in Figure 5b requires a relatively low frequency counter clock with a short delay-line and thus, a reduced-area multiplexer [2, 54]. Also, the frequency of the counter clock tap on the delay-line should be kept to a minimum due to power consumption concerns. On the other hand, the architecture is compact and power-efficient. These architectures have been primarily developed developed for low output voltage (<2.7 V) and low output power (<3 W) converters. The similar architectures to Figure 5a and Figure 5b have been presented in [70–72] for different resolutions and in the modified architecture for SIDO [73].

Another simple fully synthesizable hybrid DPWM architecture [36, 74], that is suitable for both FPGA and ASIC implementations, is shown in Figure 5c. It requires relatively small hardware resources, and can be synchronized to an external clock over a range of process or temperature variations. It is basically a hybrid 5-bit DPWM with an open loop delay lines with 3-bit counter and 2-bit delay line. This architecture utilizes MSB and LSB of duty command similar to that in Figure 5a. The output, PWM_out, is set at the zero value of the counter (cnt = "000"). The output of the counter is compared with the 3 MSBs of the duty cycle command which results in the signal $delclk$. The width of $delclk$ is equivalent to one clock period of the input clk signal. The signal $delclk$ is then propagated through the delay line. The output of each delay cell is tapped out and connected to a L : 1 multiplexer whose output is activated by LSB portion of the duty cycle command, LSB (duty). The appropriate input of the multiplexer is then connected to the output, R. The total delay of the delay line should be equal to one clock period, to guarantee monotonicity and near optimum linearity. Furthermore, the desirable delay through the delay line is achieved using active control scheme to control the delay through each individual delay cell as proposed in [36]. A hybrid DPWM with programmable dead times is also introduced in [36], as shown in Figure 5d. The DPWM provides two complementary outputs, D1 and D2. The output D1 is a periodic square wave signal with a duty cycle determined by the input duty command. The input dead-times t_{d1} and t_{d2} are applied to the output signal D2. The dead-time programmability provides the optimum efficiency in a synchronous converter [75]. The hybrid structure of the DPWM is DPWM is similar to what is shown in Figure 5c, but with an adjustable delay through the delay line, using a digital delay-locked-loop (DLL).

Another hybrid 10-bit DPWM architecture with the dead-time programmability is presented in [76]. It is composed of a 5-bit counter and a 5-bit delay line. The clock period, T_{clk} is divided into 32 time steps by 5-bit delay lines as shown in Figure 5e. The required clock frequency, 22.7MHz, is 32 times smaller than that for the equivalent counter. The architecture is composed of 3 delay lines with a 32:1 multiplexer and two SR latches. The delay lines are composed of the same delay cells so that the total propagation delay time of delay cells matches the period of clock. The counter provides the MSB portion of the command $d[n]$. The architecture provides dead-time programmability to achieve optimum efficiency as shown in Figure 5d. The DPWM provides outputs for high side MOSFET (ON/OFF signal) Q1 and low side MOSFET (ON/OFF signal) Q2.

A 10-bit hybrid DPWM using ring oscillator MUX as illustrated in [64, 77] is shown in Figure 5f. The rising edge of the PWM signal is generated by a fixed clock signal and the falling edge is generated by combining a 5-bit counter-comparator and a 5-bit ring oscillator MUX DPWM. The ring oscillator runs at the frequency of 32 fs, and the 5-bit counter divides the switching period into 32 segments. In each segment, the ring oscillator generates 32 equally spaced square waves from symmetrically oriented taps. A synchronizer is used to combine the multiplexer and comparator output. The rising edge of the PWM signal is generated at the beginning of the switching cycle. The falling edge of the PWM signal is generated from the tap which is specified by the MUX according to the 5 LSB of the $d[n]$, after the counter reaches the count corresponding to the five MSB's of the $d[n]$. A drawback of this type of architecture is that a potential race condition occurs when the delay difference between the counter-comparator and the multiplexer is large enough such that the setup time requirement of the flip-flop is violated. A modified architecture is also presented to avoid this problem [77].

All previous delay-line based architectures require the number of unary delay cells, proportional to the total

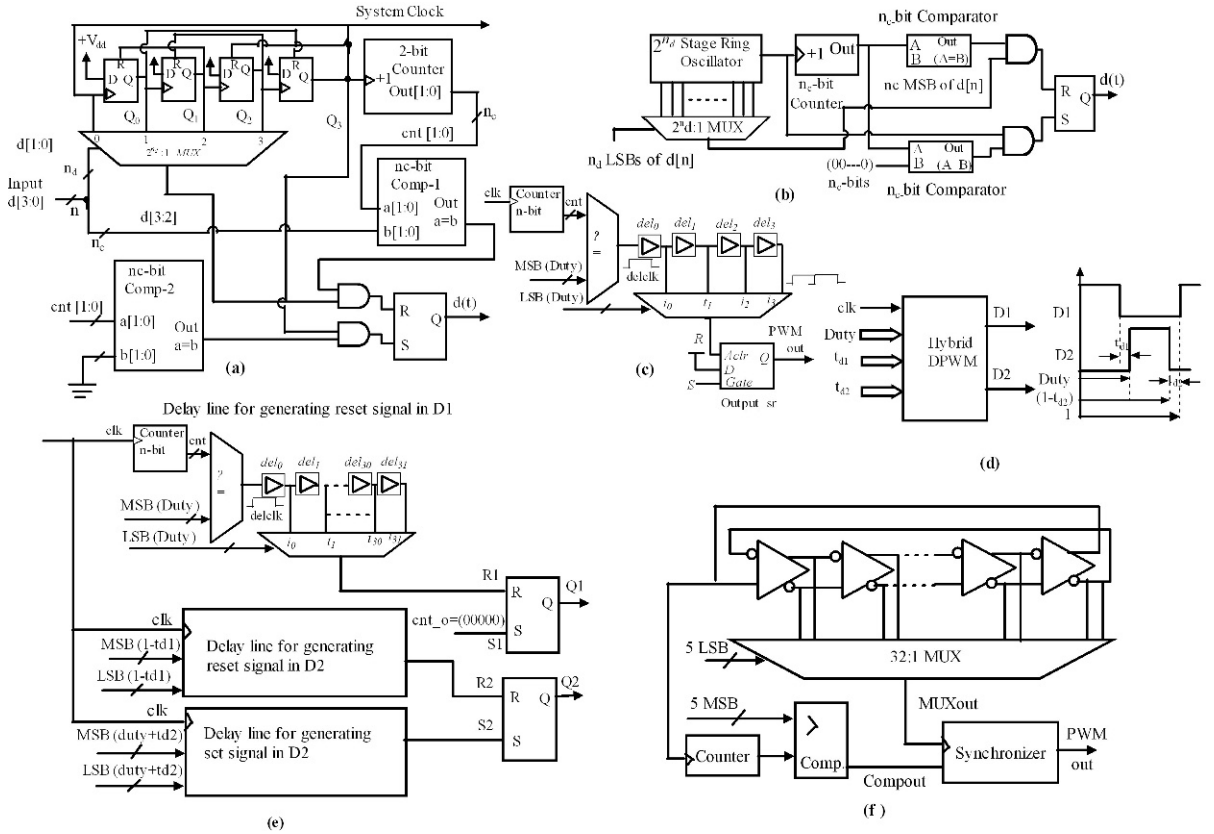

Figure 5. Hybrid DPWM architectures; (a) A 4-bit hybrid DPWM IC; (b) Hybrid counter-comparator with delay-line type architecture; (c) FPGA based hybrid architecture with external clock; (d) FPGA based hybrid architecture with programmable dead times; (e) Hybrid DPWM with dead-time programmability using counter and delay lines; (f) A 10-bit hybrid DPWM using ring oscillator MUX and counter comparator.

number of quantization levels, i.e. 2^n, which provides ultra fine resolution without suffering from a massive increase in physical area. However, in order to avoid DPWM nonlinearity, care must be taken to ensure that the maximum delay of the tapped delay line is appropriately matched with the delay associated with the counter's LSB. This may prove difficult due to the temperature variations that can affect the delay line.

2.4. Segmented Delay line based DPWM architectures

Improvement in the DPWM architectures, of section 2.3, is made by resolving the MSB bits instead of a counter using a more coarse set of delay cells i.e by segmenting the delay lines which further reduces the overall area of the DPWM [33, 35]. A fragmented version of the delay line architectures is proposed in the literature which, ultimately, will result in a smaller area compared to the delay line based DPWM [35]. An approach proposed in [33] considers binary-weighting of the delay cells so that they can be directly driven by the $d[n]$. This architecture is shown in Figure 6a, in which the delay line is composed of cells whose delays are scaled with binary weights. The architecture considers direct use of switches driven by the $d[n]$, that either bypasses or includes the delay of a given binary-weighted cell in the signal path, so that the output pulse width corresponds to the expected value of the digital-to-time conversion. Thus, area consumption reduces drastically due to the absence of multiplexer. But the linearity of the digital to time domain conversion degrades, particularly when the number of bits increases [33].

A segmented DPWM architecture proposed in [33] is shown in Figure 6b, in which the DPWM circuit is segmented into two sub-DPWM circuits. The $q-$bits of MSBs of the segmented $d[n]$ are thermometer decoded and selected from a delay line for a given pulse width with 2^q time slots. The output of this first segment which guarantee monotonic characteristic is connected in series

with a binary-weighted group of cells driven by the $n-q$ bits of LSBs. The cascade connection of the two sub-circuits results in time addition and hence the final pulse width is selectable with the resolution provided by the $n-$bit input of $d[n]$.

Another version of similar architecture presented in [35] is shown in Figure 6c for a 3-segment, 6-bit DPWM. The additional delay compensation circuit is not shown for brevity. Each segment has a 4-to-1 multiplexer controlled by two bits of $d[n]$. The delay in each of the four elements of the ith segment is given by (7):

$$\Delta t_i = 4 \cdot \Delta t_{i-1} = 2^{2^i} \Delta t_o \tag{7}$$

The Δt_i delays are created by simply replicating the Δt_0 delay cell, resulting in the same overall number of delay cells, neglecting the additional dummy load cells. The modest area reduction compared to the traditional delay line DPWM comes from the multiplexer, since a 64-to-1 multiplexer is replaced by four 4-to-1 multiplexers, a positive edge of the square wave clock sets the SR flip flop, which starts the output pulse. Simultaneously, the clock starts propagating through a series of delay cells: sixteen cells - sixteen cells - sixteen cells-dummy cell. According to the values of the $d[5,4]$, one of the taps in the delay line of seg 2 is selected, and the rising edge of the signal at that tap passes through the 4 to 1 multiplexer to the next segment. At this stage, a similar transfer of the signal occurs through the selection of the desired tap. Finally, the positive edge of the signal resets the SR flip flop marking the end of the output pulse. The length of the path for the signal through the segmented structure is determined by the inputs duty command $d[5,0]$ to the DPWM. Based on the segmented architecture, a 6-bit proof-of-concept DPWM IC was designed and fabricated [39] as mentioned in [33]. Similar to Figure 6c, a segmented structure of the DPWM is shown in Figure 6d. An extra delay cell at the beginning, and one dummy delay cell at the end of these stages provides a similar driving and loading environment to each of the delay cells [33, 78]. The delay compensation unit is also provided.

Further reduction in area and power consumption in segmented DPWM is achieved by using identical, voltage controlled tunable delay elements instead of replicating Δt_0 [35]. Each of the larger delays can be achieved by using the same delay cells with different control voltages, thereby saving on the total area of the delay line [35]. In addition, DLL is used to tune the delays such that the non-linear and monotonic effects due to delay mismatches are smaller. The control voltage of each segment $V_{control}$ is adjusted to meet (7) using an embedded delay-locked loop (DLL) as shown in Figure 6e for an 8-bit DPWM to linearize the segmented DPWM. Using this self-calibrated architecture each segment is identical and differs only in its $V_{control}$. Each segment's DLL calibrates the delay cells to be four times longer than the previous segment. This approach claims a good linearity with a minimum number of delay cells ($2n$ instead of 2^n). The use of four DLL consumes a slightly large area. A modified architecture, shown in Figure 6f [35], has single DLL and therefore a compromise between the architectures of Figure 6d and Figure 6e. Each delay element in Seg3 and Seg1 is made using four delay cells from Seg2 and Seg0 respectively. The Seg0 and Seg1 blocks do not require a cal output. Therefore, the total number of 4:1 MUXs is reduced from 8 to 6. The total number of delay cells is increased from 16 to 40 compared to the DPWM shown in Figure 6e. The delay cells in Seg0 and Seg1 do not require an explicit $V_{control}$. The oscillation frequency can be simply adjusted by varying the DPWM V_{dd} reference. To achieve a constant switching frequency in all DPWM, the Vdd should be regulated. The DLL adjusts the Seg2 and Seg3 delays to maintain linearity while the V_{dd} is adjusted to set the LSB delay. The delay calibration between Seg2 and Seg1 is achieved by synchronizing the row1, and cal2 rising edges. The DLL uses a precision phase comparator and a charge pump to regulate the delay [35]. In ASIC solutions, it is possible to control the supply voltage of the delay cells to control its delay, but it is not possible in FPGA solutions.

The core of the High-frequency segmented ring DPWM shown in Figure 6g is the modification of Figure 6d, with no external clock [32]. The architecture presented in [79], also resembles the DPWM architecture based on a ring oscillator, with a difference in the size of the structure. The architecture of Figure 6g is an 8-bit DPWM which consists of two 16:1 multiplexers and two sets of delay lines connected as a ring. The first delay line consists of 16 fast delay elements in series. Each intermediate node is passed onto a 16:1 multiplexer (MUX-A), whose select signals are the 4 LSBs of the digital input $d[n]$. MUX-A is responsible for the fine resolution of the DPWM. The second delay line consists of 15 slow delay elements in series. These delays are 16 times slower than the fast delay elements. The outputs of slow delays are connected to a second 16:1 multiplexer (MUX-B). The select signal for the MUX-B are the 4MSBs of $d[n]$ connected in reverse order, such that the MSB of $d[n]$ is tied to the LSB of MUX-B. The 4MSB of $d[n]$ define the start point of the pulse-width modulated signal that sets the SR latch. The 4LSBs of $d[n]$, connected to the select input of MUX-A, are not reversed and define the end point of the fraction of switching period during which the PWM signal is high. The DPWM can be implemented with very small hardware; it takes only 1/15 of the resources needed for

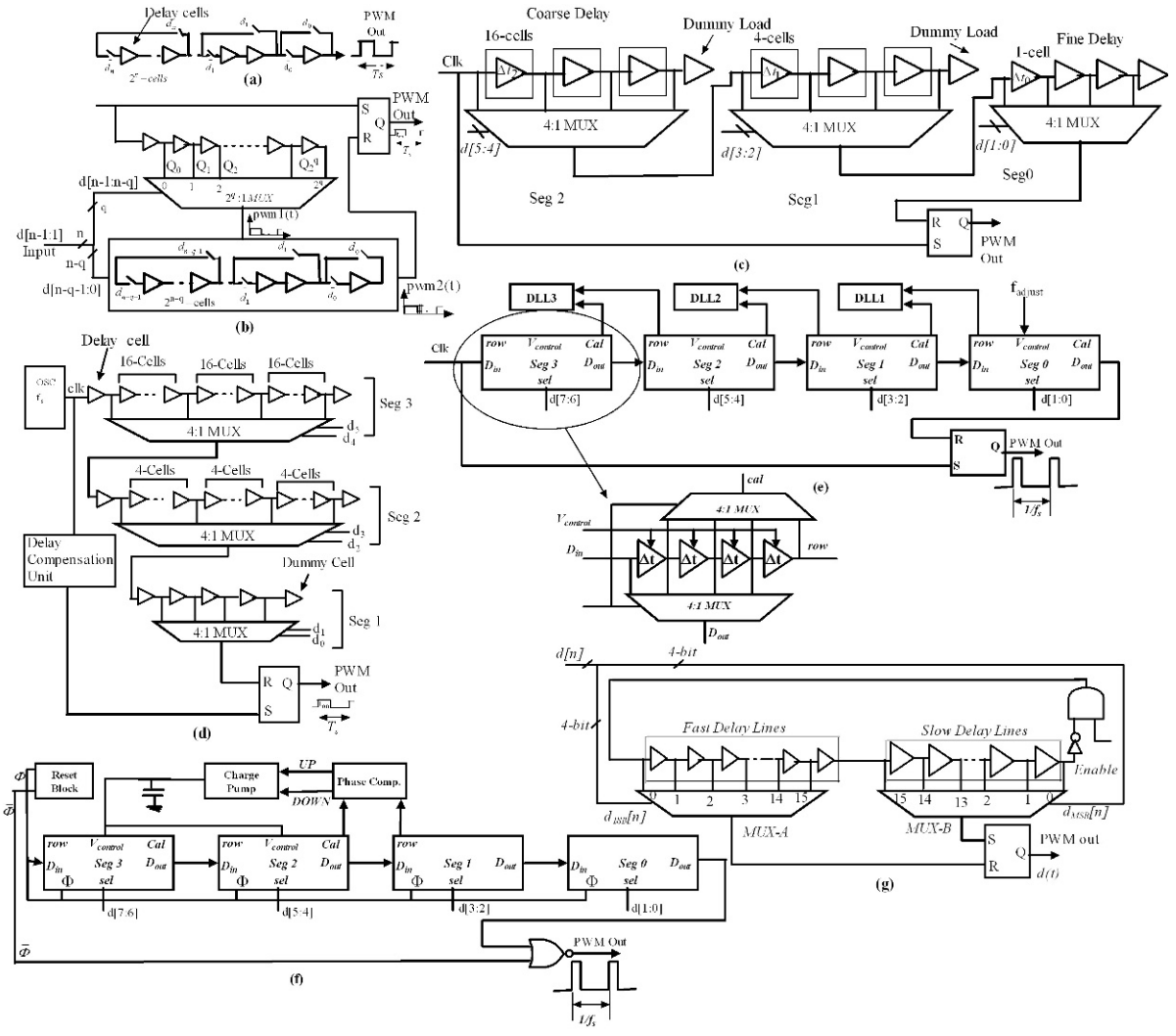

Figure 6. Segmented delay lines based DPWM architectures; (a) Binary-weighted delay-lines DPWM.; (b) Segmented DPWM architecture with Binary-weighted delay-lines; (c) Simplified block diagram of a 3-segment, 6-bit DPWM; (d) Segmented DPWM IC prototype; (e) Segmented DPWM by using identical, voltage controlled tunable delay elements using DLL; (f) Modified Segmented DPWM of Figure 6e with single DLL; (g) Ring based segmented DPWM architecture with an external clock.

the implementation of a conventional 8-bit ring-oscillator based DPWM that employs a large 256:1 multiplexer.

The main advantage of segmenting delay cells DPWM is that it can be realized with much smaller area than the delay-lined DPWM. Instead of having 2^n : 1 MUX, it is logarithmically divided into two or more MUXs, which are much smaller than original MUX. If the delay cells are binary-weighted, the total number of delay cells is much smaller. However, there are a few disadvantages; firstly, it requires an external clock to synchronize the output of DPWM. Secondly, linearity is degraded due to the mismatches in the delay cells as in delay line DPWM. There is also an additional source of non-linearity due to the additional MUXs. As the number of segments increases, linearity decreases. Also, the amount of non-linearity is code-dependent and is more problematic for larger duty ratios. Thirdly, and more importantly, monotonosity can't guarantee if the delay cells are binary-weighted due to mismatch.

2.5. Hetrogeneous DPWM architecture

As proposed in [40] the heterogeneous-DPWM, shown in Figure 7, is an extension of the hybrid-DPWM that is

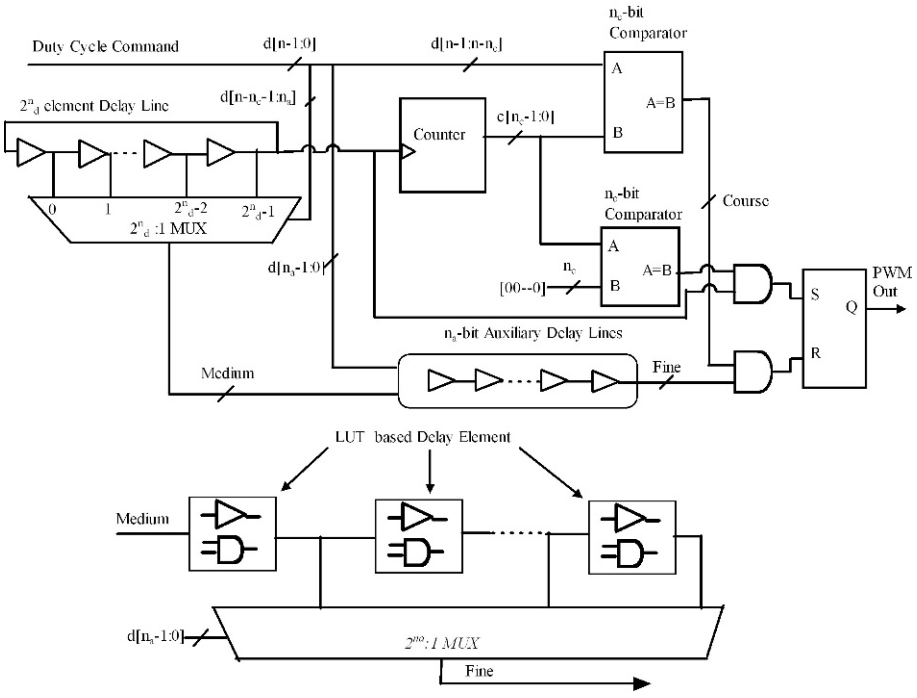

Figure 7. Single Phase heterogonous-DPWM architecture with auxiliary delay-line structure, using LUT-based delay elements.

modified by adding a na-bit auxiliary delay stage at the multiplexer output, and elements that have a finer latency than those of the main delay-line. A single pulse cycles around the ring-oscillator and increments the counter on each pass through the ring-in the same way as in hybrid-DPWM. The SR-latch is set each time the counter output returns to zero. The duty-cycle command bus, $d[n-1:0]$ is divided into three sub-busses: $d[n-1:n-n_c]$ sets the coarse time comparator reference, $d[n-n_c-1:n_a]$ selects the appropriate tap of the $2n_d:1$ multiplexer, and $d[n_a-1:0]$ selects the delay of the auxiliary delay-line. The SR-latch is reset when the specified duty-cycle command value is reached. Here,

$$n = n_c + n_d + n_a \tag{8}$$

where, n is the number of bits of resolution of the pulse width modulated waveform, n_c is the counter output-bus width, n_d is the number of select inputs to the ring-oscillator multiplexer and na is the number of select inputs to the auxiliary delay line. There are three time-interval grains-coarse, medium and fine-generated in turn by the counter comparator, the ring oscillator delay-line and the auxiliary open-loop delay-line respectively. Each time interval divides the time-interval of the previous stage into smaller steps. The main delay line uses flip-flop delay elements, the auxiliary delay line uses look up tables ($LUTs$) to generate the delays, shown in Figure 7. This architecture generates high-resolution PWM waveforms using a minimum area, while limiting the counter clock frequency to an acceptable value. This shows the DPWM to be more area-efficient. The primary objective of this heterogeneous-DPWM is to generate high-resolution PWM waveforms with a minimum area, while limiting the counter clock frequency to an acceptable value. The relative performance comparison of classical DPWM architectures is presented in Table 1.

3. FPGA's advanced characteristics based DPWM architectures

Modern day SMPS operate at frequencies in the range of 1-10MHz [2, 3]. As frequency increases, T_s decreases, Δd (i.e $\Delta d = t_{on}/T_s$) and becomes coarser (lower resolution). A number of solutions have been suggested to increase the DPWM resolution. Classical methods provide many solutions but do not provide an adequate resolution. In this section new architectures for increasing the resolution of the DPWM by using advance resources of the FPGA and the hybrid architecture with software

Table 1. Relative performance comparison of classical architectures.

DPWM Type	Different Comparison Parameters for DPWM					
	Silicon Area	Power	Guaranteed Monotonic	Required external clock	No.of delay elements	Max.clock frequency
1.Counter	Lowest	Highest	Yes	Yes	0	$2^n \cdot f_s$
2.Delay lines	Highest	Low	Yes	No	$\approx 2^n$	f_s
3.Hybrid	Low	High	No	No	$\approx 2^{n-n_c}$	$2^{n_c} \cdot f_s$
4.Segmented	High	Lowest	No	No	$\approx 2 \cdot n$	f_s
5.Hetrogenous	Low	Low	No	No	$\approx 2^{n-n_c}$	$2^{n_c} \cdot f_s$

whereas, n =total number of duty command bits, n_c =no. of bits resolved by counter

approach [13, 59] are included. These architectures achieve a resolution higher than the methods discussed in Section 2. Two attractive digital techniques are available for implementation of DPWM in FPGA-based system, one the Delay-Locked Loop (DLL) [44, 45, 47–49, 51, 52, 59] and second the segmented DCM [46, 59]. Some of these DPWM architectures uses partial FPGA resources [44, 45, 47–49, 51, 52, 59], while others use full FPGA resources [46, 53]. All of these architectures belong to the class of hybrid architectures.

3.1. Hybrid DPWM using DLL asynchronous block

This DPWM is based on advanced DLL features of FPGA that are available in almost every FPGA. In new hybrid DPWM the synchronous block is counter based. However, the asynchronous block is not based on delay-line structure (as in Section 2), but using FPGA's available DLLs with phase-shifting features. Taking advantage of these resources, the architecture achieves an excellent tradeoff between linearity and time resolution. This DPWM architecture is, in principle, intended for FPGA implementation. Since DLL block is already available in FPGAs, an FPGA implementation is very simple.

3.1.1. FPGA resources: Delay-Locked Loop (DLL)

FPGAs have specific blocks i.e. DLL that can manage clock signals. Using DLLs, it is possible to multiply or divide the clock frequency. Many of these DLLs can also generate four phase-shifted clocks (shifted 0°, 90°, 180° and 270°) directly [80–82]. The first feature of these DLLs is used in this DPWM i.e. multiplying the clock frequency. of the advantage of this is that a high clock frequency can be internally used in the DPWM, while an external lower frequency is generated and also used in the rest of the digital controller. DLL based clock multiplication is used in the DPWM as reported in [45, 48, 49]. As shown in Figure 8a, a 32 MHz external clock is multiplied by 4 for a 128 MHz internal clock in the DPWM. Using the multiplied clock, time resolution increases from 31.25 ns to 7.81 ns. However, the rest of the controller modules i.e. control algorithm can work at the low clock frequency for decreased power consumption and easier design. Therefore, different clock frequencies are proposed in the architecture. However, the main contribution of the DPWM comes from second feature of DLL. Most DLLs in FPGAs also generate phase-shifted versions of the output clock. The clock shifting feature allows multiplication of time resolution by 4 (2-additional bits) beyond the maximum resolution achievable with a counter-based technique [45, 48]. The DPWM could obtain a higher resolution, multiplying the resolution of a counter-based solution by the number of available clocks. Using the phase-shifted clocks it is necessary to use both synchronous and asynchronous techniques as explained below.

3.1.2. Synchronous block

The synchronous block is a counter-based DPWM that uses the most significant bits (MSBs) of the duty cycle, $d[n-1,2]$, where n is the total number of bits. As shown in Figure 8b, the Synchronous block is based on a counter and comparison structure; capable of working at high clock frequencies due to its simplicity. In the synchronous block, resolution is given by both the clock and the switching frequency, obtained as [45, 48, 49]:

$$\text{Resolution} = f_{clk}/f_s \tag{9}$$

where, f_{clk} is the clock frequency and f_s is the switching frequency.

3.1.3. Asynchronous block

The two LSBs, $d[1,0]$, are used in the asynchronous block. These two bits are used to select between the four phase-shifted clocks generated by the DLL These signals are combined using AND gates (Figure 8b) in order to obtain other four phase-shifted signals that are high only a quarter of a cycle (Figure 8c). These signals

obtain four possible switching instants during each clock cycle, thus, resolution is quadrupled. In general, using m asynchronous bits (and $2m$ phase-shifted clocks), the total resolution is calculated as [45, 48, 49]:

$$\text{Resolution} = 2^m \cdot f_{clk}/f_s \tag{10}$$

where, m is the number of asynchronous bits f_{clk} is the clock frequency and f_s is the switching frequency.

3.1.4. Functionality of the Hybrid DPWM architecture

DPWM shown in Figure 8b [45, 48, 49] is formed by two blocks: a synchronous block and an asynchronous block, as already explained. The output of the synchronous block (counter-based with external frequency multiplied by 4) sets the HSM or output of the DPWM depending on the MSBs of $d[n]$. The synchronous DPWM uses only the 0° shifted clock, while the asynchronous block needs all four phase-shifted clocks. The DPWM architecture of Figure 8b creates driving signals for both the high side MOSFETs (HSM) and low side MOSFETs (LSM), as is intended for a synchronous buck converter. The turn-on instant of the HSM is always coincident with a 0° clock edge, but the turn-off instant depending on the LSBs, can be at any of the four clocks edges. The opposite is done for the LSM. In Figure 8b, the asynchronous block generates the signal named QuarterCycle, which corresponds to a quarter of the clock cycle, starting in the rising edge of one of the four clocks, depending on the value of the 2 LSBs of the $d[n]$ (Figure 8c). Output is reset for HSM when the synchronous block is already off and QuarterCycle arrives. Therefore, the output is active for integer number of clock cycles plus 0 to 3 quarters of a cycle. However, using asynchronous techniques, the DPWM suffers from non-monotonic behavior. In order to avoid this behavior some modifications in the DPWM architecture have also been proposed [45, 48, 49].

3.2. Hybrid DPWM using DCM asynchronous block with single/multiple clocking

FPGA's other advanced clock management capability, the fine phase shifting of the clock, is utilized, to develop a high resolution hybrid DPWM architecture [52]. Thus, allowing very small and programmable delays between the input and output clocks as shown in Figure 8c. An original use of this fine phase shifting pushes the limits of DPWM resolution.

As reported in [52], shown in Figure 8d, a FPGAs DCM block is utilized to shift the phase of the clocks in small increments. The resolution obtained by this method is much higher than that with the classical architectures [52]. Apart from other features (to be discussed in Section 3.5.1), the ability of DCM to shift the phase of the output clock with respect to the input clock using some additional control signals, called fine phase shifting (see Figure 8c). It is utilized in the architecture of Figure 8d. Its resolution is greater than the 1/256 of the input clock cycle and one tap delay of its internal DLL. The delay of a tap is the range of 7-30 ps [80, 81]. This resolution is finer than those of the classical DPWMs.

Phase shifting in a DCM works with input clock which is used as a time reference. Using the appropriate control signals, the relative phase of the input clock CLKIN and the output clock CLK0 can be changed in steps of 1/256 of the clock period (Figure 8c). The DCM uses delay taps for adjusting the relative phase and is adjusted to a number of delay taps that approaches the desired solution. Any resulting phase, obtained by single increment steps, from 0° to 360° is possible, as long as the clock period is below the total delay of all the delay taps together. The basic structure is also a hybrid structure shown in Figure 8d and is similar to Figure 8b, in which the MSBs (say M—bits) of the $d[n]$ are handled in a synchronous block (counter-based) and the LSBs in an asynchronous block. The synchronous block manages M-bits, the internal counter ranges from 0 to $2^M - 1$. Whenever, the M MSBs is below the value of the counter; the main output of this block [Sync HSM] is high, otherwise it is low. Dead-times are included to manage topologies with more than one switch, like the synchronous buck converter. The synchronous block uses the input clock CLKIN and not the shifted clock generated by the DCM. The asynchronous block manages the LSBs. It uses 8 bits because the DCM of Xilinx FPGAs has 256 steps of fine phase shift. The rest of the asynchronous block uses the clock CLK0, which is shifted with respect to CLKIN in all the possible range, from 00 to 3600, depending on the eight LSBs. The working of the circuit can be explained using a simplified version shown in Figure 8d. The way in which the DPWM output for the HSM is generated is shown in Figure 8e. The MSBs determine the number of clock cycles that the output has to be active through Sync HSM, which is also the set signal of the RS-latch at the output. The eight LSBs are used for defining the fraction of clock cycle to be added at the output. These bits go to the block in charge of controlling the fine phase-shift process, which is a finite-state machine that shifts the CLK0 clock in step by step, until the phase between CLKIN and CLK0 is equal to the fraction of clock cycle to be added. The RS latch is reset when the CLK0 signal is active, but only after the set signal is already inactive (and gate, inverter). Therefore, the output signal (HSM) is the active integer number of clock cycles (determined by the MSBs) plus a fraction

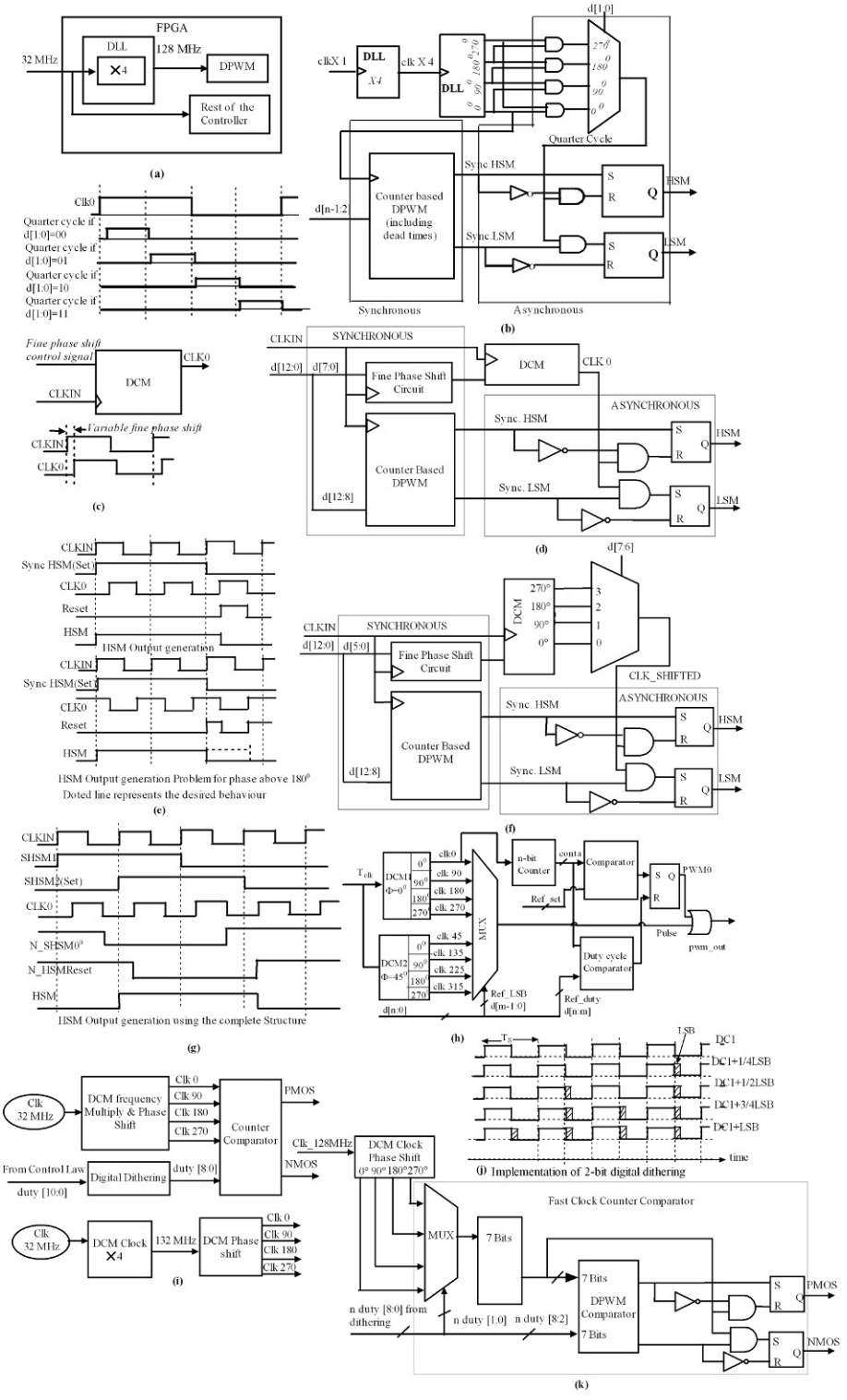

Figure 8. Hybrid DPWM architectures using FPGAs advance resources DLL and DCM; (a) Different clock domains for the DPWM and rest of the controller; (b) Basic structure of DPWM with DLL Asynchronous block and corresponding QuarterCycle signal depending on the value of $d[1:0]$; (c) Fine phase shifting-very small and programmable delays between the input and output clocks; (d) Basic structure of DPWM using DCM asynchronous block with single clocking; (e) HSM output generation problem for phases above 180°. Discontinuous line represents the desired behavior; (f) Basic structure of DPWM using DCM asynchronous block with Multiple clocking; (g) HSM output generation using the complete structure(modified); (h) Hybrid DPWM using DCM and counter-comparator; (i) Basic scheme of DPWM using DCM and counter-comparator with digital dithering; (j) Implementation of 2-bit digital dithering; (k) DPWM using DCM counter-comparator with digital dithering.

of clock cycle (determined by the LSBs). An asymmetric technique is used for generating the LSM output, using CLK0 for the set of LSM and the synchronous signal Sync LSM (synchronized with CLKIN) for the reset.

The drawback of this technique, represented in Figure 8d, is that it would not work for phases above 180° as shown in Figure 8e. In these cases, the reset signal would also be active at the beginning of the clock cycle, resetting the output before expected. In order to avoid this problem, a real modified architecture is also reported in [52], shown in Figure 8f and its corresponding waveforms are shown in Figure 8g. The main drawback of the architecture of Figure 8d is the phase-shift update time. The phase of the output clock of the DCM must be changed each time any of the eight LSBs of the duty cycle changes. This is done through a request acknowledge protocol in steps of 1/256 of the clock period [52]. If the update time is critical, the phase-shift update time can be decreased using more than one clock. These clocks must have fixed phase differences among them. Figure 8f shows such structure using four clock lines. The asynchronous block will be same as one shown in Figure 8d, but using CLK_SHIFTED instead of CLK0. In that case, the maximum phase shift will be 90° and only 64 fine phase steps are necessary. Each DCM in FPGAs includes four clock outputs phase-shifted by 0°, 90°, 180°, and 270° [80–82], and that most FPGAs have between two to eight DCMs [80–82], the maximum phase shift can be drastically reduced, if necessary. The phase-shift update time affects only the transient behavior as it produces relatively small transient error [52].

3.3. Hybrid DPWM using DCM and counter-comparator

Implementation of a hybrid DPWM using counter based architecture and a FPGA DCM block is shown in Figure 8h [47]. The time period of the system clock T_{clk} is sliced by means of the proper phase shifting of the clock signal. This solution provides a lower equivalent FPGA clock period; the architecture uses two DCM, to generate the phase-shifted clock signals. The DPWM should provide the power stage driving signals with the adequate duty cycle value encoded in $n-1$ bits. The $n-m$ MSBs, ($d[n,m]$), provide the coarse pulse width while, the $m-1$ LSBs and ($d[m-1,0]$) are used to obtain the finer pulse width. When the output code (conta) of the $n-$bits counter is equal to the ref_set code, the comparator sets the RS flip flop. It implies that a positive edge of the $pwm0$ signal is obtained ($pwm0 =' 1'$). In addition, when the value of conta is equal to ref_duty, the duty cycle comparator generates the reset signal of RS flip flop and thus, the negative edge of the $pwm0$ signal is obtained ($pwm0 =' 0'$). As a result, the coarse pulse width (t_{ON}, MSB) consists of an integer number of cycles of the reference clock, $clk0$. At the same time, the negative edge of the pulse signals (pulse90, pulse180 and pulse 270) are generated in synchronism with the first positive edge of the corresponding phase-shifted clock signal (clk90, clk180 and clk270). The output signal of the multiplexer (pulse) is chosen according to the LSB code. Finally, the negative edge of the selected pulse signal generates the negative edge of the DPWM output signal ($dpwm\_out$). Hence the pulse width value of $dpwm\_out$ corresponds to the addition of t_{ON}, MSB and t_{ON}, LSB, where t_{ON}, LSB is the time difference between the negative edges of $pwm0$ and the selected pulse signal. As a result, the DPWM is synchronous. Finally, the DPWM time resolution (Δt_{min}) is defined as the ratio between T_{clk} and the number of the phase-shifted clock signals (N_Q) and is given as:

$$\Delta t_{min} = T_{clk}/N_Q \tag{11}$$

Also, the DPWM voltage resolution (ΔV_{min}) is proportional to the DPWM time resolution and is given as:

$$\Delta V_{min} = V_{in} \cdot \Delta t_{min}/T_s \tag{12}$$

where, V_{in} and T_s are the input voltage and the switching period respectively. This architecture provides a lower equivalent FPGA clock period, avoiding the time resolution limitation of the counter-based DPWM approach.

3.4. Hybrid DPWM using DCM and counter-comparator with digital dithering

A method presented in [51] makes use of FPGA's DCM with frequency-multiply and phase-shift characteristics. It combines a counter-comparator block with digital dithering approach to increase the DPWM resolution. The structure is shown in Figure 8i and it includes three parts of DCM frequency-multiply and phase-shift block, digital dithering block and counter-comparator block. An external clock of 32 MHz is multiplied by two DCMs and then the internal clock reaches 128 MHz which can realize a 7-bit DPWM under the switching frequency of 1 MHz. The third DCM provides shifted phase based on 128 MHz clock such as 0°, 90°, 180°, 270°. The DPWM resolution is increased 2 bits by phase-shift without increasing the clock frequency by fourfold [51]. The digital dithering block is also important. The basic principle of programmed digital dithering is to use look up table according to LSBs of $d[n]$ and distribute the results of look up table with MSBs of PWM by some pre-scheduled rules. The

average of duty cycle values over several switching periods is equal to the $d[n]$ provided by the controller, so that a high resolution DPWM is achieved with a low clock frequency. In hybrid DPWM scheme, 2 LSBs from the control law are used for dithering, and the dithering mode is shown as Figure 8j. After a series of four switching periods, the average duty cycle over the four periods is obtained to implement Levels (1/4) LSBs, (1/2) LSBs and (3/4) LSBs. A look up table for 2-bit digital dithering for minimum-ripple dither pattern is given in [51]. The minimum ripple dither sequence is necessary to avoid poor output regulation, EMI and limit cycle [51].

The complete architecture presented in [51] is given in Figure 8k. In each switching period, control law provides an 11-bit duty [10:0] and its 2 LSBs duty [1:0] is realized by digital dithering block, duty [3:2] is realized by 4-to-1 multiplexer which has the input of four clock signals $clk-0$, $clk-90$, $clk-180$, $clk-270$ and passes one of them to the counter-comparator block, 7 MSBs duty [10:4] is realized by the counter-comparator. Finally, an 11-bit DPWM is achieved with a reasonable clock frequency of 32 MHz external clock and 128MHz internal clock. Four clock signals with different phases offer four switching times at every switching period, accordingly, the DPWM resolution is increased by fourfold. The presented hybrid DPWM achieves an 11-bit resolution, 1 MHz switching frequency in FPGA. The advantages of this hybrid method are its simple algorithm and loose demand for hardware. But relatively, it has higher internal clock which causes more power consumption and high-frequency noises.

3.5. Hybrid DPWM using segmented DCM and counter-comparator

These architectures use a hybrid approach which fully utilizes the DCM resources [46, 53]. The Segmented DCM DPWM architectures are low power architectures and allow for high switching frequency operation. The inherit phase shifting properties of the DCM blocks simplify the duty cycle generation. The architecture can be applied to achieve various numbers of bits for the DPWM resolution. These new approaches offer reduced external clock frequency, making full use of the on-board DCM resources and DLL capabilities, unlike the design presented in [45, 48, 49], where only the two LSBs, $d[1,0]$, are used to select among one of four phase shifted clock versions generated by the FPGA's DLL. In Segmented DCM, most of the resolution bits are achieved by the DCM blocks and the remaining bits are use to implement the counter-comparator technique. The savings in the oscillator clock frequency are more appreciable particularly when higher DPWM resolution is required.

3.5.1. FPGA resources: Digital Clock Manager Circuit (DCM)

Another advance FPGA block that shifts the phase of the clock is called DCM (Digital Clock Manager) [46, 53, 80–82]. Advance FPGA devices like Virtex-4, provide many on chip DCM circuits which offer zero propagation delay, lower jitter and better phase-shift resolution [53, 80–82]. DCM produces an input clock to the DCM Block i.e.Clk_in, the four different versions of clock phase shifts: $Clk\_shifted$ 0°, $Clk\_shifted$ 90°, $Clk\_shifted$ 180° and Clk_shifted 270°, and also the double and four times the input clock, Clk2x and Clk4x, along with their inverted signals, $Clk2x\_180°$ and $Clk4x\_180°$ respectively (Figure 9a). The phase shifted versions of the input clock corrects the duty cycle and locks it at 50% duty cycle. Full advantage of the DCM capabilities is taken and the same phase shifts are generated with duty cycles other than 50%. Two DCM block design can be used to meet the desired functionality. These are the $DCM\_1/4$ and $DCM\_1/2$ clock signals shown in Figure 9b. The $DCM\_1/4$ and $DCM\_1/2$ will generate phase shifts versions with 0.125/0.875 and 0.25/0.75, ON-time/OFF-time relationship respectively [46, 53]. In order to get the delayed waveform generated from both $DCM\_1/4$ and $DCM\_1/2$, at the same phase shifts locations of delays 0°, 90°, 180°, and 270° as the original DCM shown in Figure 9a, the delayed waveforms must go through a bank of AND gates. It uses both the true and the inverted forms of $Clk2x$ and $Clk4x$ previously generated from the original DCM block as shown in Figure 9b. The resolution can be measured as the minimum increment in ON-time step between two successive duty cycle values as depicted in Figure 9b. The resolution calculation using two successive duty cycle values (d_{in1} and d_{in2}) and the minimum increment in ON-time step (Δt) in one switching period T_s, are expressed in (13), as:

$$\begin{aligned}\Delta d &= (d_{in2} - d_{in1}) = (t_{on2}/T_s - t_{on1}/T_s) \\ &= (t_{on2} - t_{on1})/T_s = \Delta t/T_s\end{aligned} \tag{13}$$

where, smaller Δt between two successive duty cycle values (d_{in1} and d_{in2}) means higher resolution and thus enhanced performance. As seen from Figure 9b, different instances can be accessed within one Clk_in period; at both the rising and falling edges of the generated phase shifts. This will enable a reduced Δt when $DCM\_1/2$ is used compare to the original DCM block In other words $DCM\_1/2$ and $DCM\_1/4$ generates, four and eight digital values respectively, for the duty cycle between 0 and 1 in one Clk_in period as illustrated in Figure 9c. It shows that $\Delta t1/4$ is less than $\Delta t1/2$ which means that the resolution of

$DCM\_1/4$ is higher than $DCM\_1/2$ and it guarantees better performance since it allows for more duty cycle values in one clocking period. This DCM characteristic offers increased resolution, whilst maintaining the same input clock without the need for a higher clock frequency.

Furthermore, the increase in the resolution uses another attractive feature of DCM blocks. The two DCM blocks may be cascaded as shown in Figure 9d. The input clock, Clk_in, propagates through the first DCM block i.e $DCM\_1/2$, and the first phase shifted versions viz. $DLL1\_0°$, $DLL1\_90°$, $DLL1\_180°$ and $DLL1\_270°$ are generated together with double the input frequency, Clkx2, at which the second DCM block i.e $DCM\_1/4$ is operated, and further phase shifted signals of the clock viz. $DLL2\_0°$, $DLL2\_90°$, $DLL2\_180°$ and $DLL2\_270°$ are produced. The resolution is now increased by 16 times without the need of operating the whole system at 16 times higher Clk_in, as evident from Figure 9d. Two DCM blocks in series increases the resolution by 16 times as compare to four times ($DCM\_1/2$) or eight times ($DCM\_1/4$) when single DCM Block is used. Therefore, the feature of cascading DCM blocks will increase the resolution without operating the whole system at higher clocking. This will ultimately enhance the resolution, minimize the power dissipation, and pave the way to generate PWM signals with higher switching frequencies.

The comparison of two architectures show that, a 6-bit DPWM implementation using conventional DLL presented in [45] requires a $f_{clk} = 16 \cdot f_s$ as compared to $f_{clk} = 4 \cdot f_s$ in 6-bit Segmented DCM DPWM proposed in [46, 53]. The reduced clock requirements attained in DCM DPWM helps in minimizing the power consumption.

3.5.2. Segmented DCM based-DPWM operation

The segmentation of the available DCM blocks is utilized in the design of this DPWM to increase the effective resolution of the system. This approach is a hybrid technique, conceptually similar to the hybrid DPWM reported in [36], and the DLL-based DPWM in [45, 49, 52]. A 6-bit Segmented DCM based DPWM, $d[5, 4, 3, 2, 1, 0]$, redesigned by using two cascaded DCM blocks (4-bits) and a 2-bit counter, shown in Figure 10a. The duty command value generated by the compensator will vary from a minimum value of $d[000000]$, to a maximum of $d[111111]$. The 6-bit duty command goes into the DPWM block, where, the comparator will be set when the counterpart matches the 2-bit $d[5, 4]$, and the output of the DCM blocks will be decided by the 4-bits: $d[3, 2]$, $d[1, 0]$, as shown in Figure 10b.

The simplified block diagram of Figure 10a is shown in Figure 10b, where one block labeled $DCM1\_1/4$ represents the combination of $DCM1$ block and the bank of eight 3-input AND gates and $DCM2\_1/2$ represents combination of $DCM2$, 4x1 Multiplexer (MUX) and the bank of 3-input AND gates. The operation of the simplified Segmented DCM-based DPWM architecture of Figure 10b is as follows: First, the clock frequency from the oscillator goes through $DCM1\_1/4$ blocks. The output of $DCM1\_1/4$ is one of four different phase shifted versions of the input clock; $DLL1\_0°$, $DLL1\_90°$, $DLL1\_180°$ and $DLL1\_270°$ with 0.125/0.875 ON-time/OFF-time relationship. The value of the MUX select coming from the compensator duty cycle command $d[3, 2]$, selects among these $DCM1\_1/4$ output versions and the output of the MUX, Sx, is then further phase shifted by $DCM2\_1/2$ block. The counter-comparator block is operated at the same clock frequency of $DCM1\_1/4$ block as shown in Figure 10b. The SR latch is set at the beginning of each counting cycle (time required for the counter to count up starting from zero to $2^{n_c} - 1$). One counting cycle for the counter constitutes one switching period for the converter, $T_s = 1/f_s$. In this case, the counter cycle starts the sequence from 0 to $2^{n_c} - 1 = 2^2 - 1 = 3$. On the other hand, the SR latch is reset when both the comparators detect that the counter reaches the value of $d[5, 4]$ AND the clock is shifted by $DCM1\_1/4$ block followed by phase shifting in $DCM2\_1/2$ block. The resets the SR flip flop determines the converter duty cycle value $d[n]$.

More DCM blocks can be cascaded in the segmented DCM architecture to further decrease the clocking requirements. A new Segmented DCM DPWM is reported in [53] which help in further minimizing the power consumption.

3.6. Hybrid DPWM using window-segmented DCM and counter-comparator

A new windowing feature is added to the circuit, previously shown in Figure 10a [53]. Its purpose is to eliminate the propagation of the generated phase shifts through the whole switching cycle and limits it to a moving window set by the comparator. The operation of the reported window-segmented DCM-based DPWM architecture of Figure 10d is as follows: first, the counter ramps up from zero till it reaches $2N_{clk} - 1$ at every rising edge of the clock f_{clk}, making one counting cycle (Switching period for the converter $T_s = 1/f_s$, during which the counter cycle starts the sequence from 0 till $2^2 - 1 = 3$) as illustrated in Figure 10d. In other words, one switching period is composed of $2N_{clk}$ slots. As shown in Figure 10d, a 2-bit counter divides one switching cycle fs into four slots. The comparator compares between the counter value and the duty cycle command coming from the compensator $d[5, 4]$. The output of the comparator

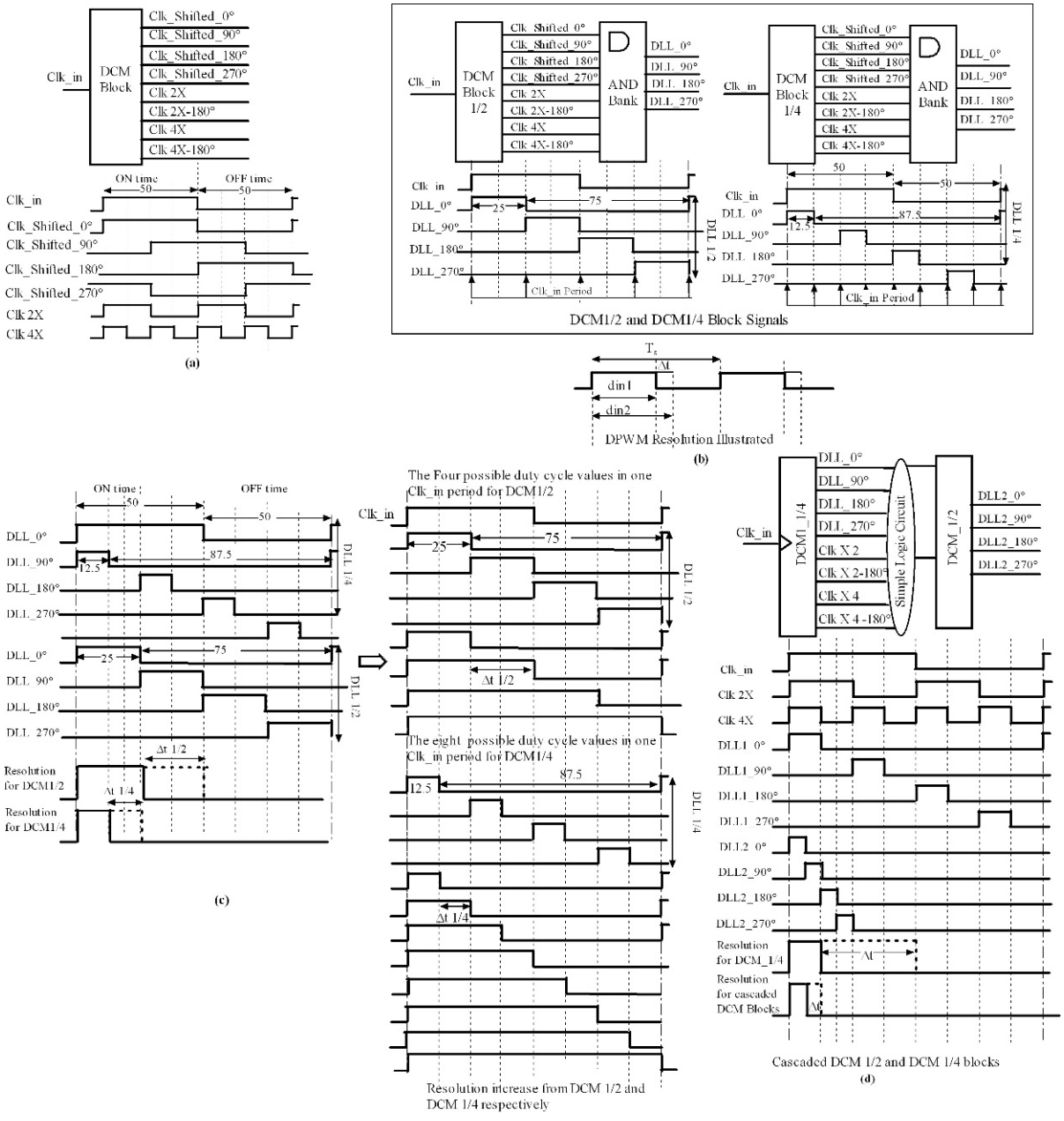

Figure 9. (a) Digital clock Manager (DCM) Block and its possible outputs; (b) DCM Blocks 1/2 and 1/4 blocks and possible outputs; (c) Resolution increase achieved from DCM1/2 and DCM 1/4; (d) Cascaded DCM blocks 1/2 and $\frac{1}{4}$ and corresponding outputs.

sets the window signal high when the count value matches the duty cycle $d[5,4]$. The generated window signal goes through *DCM*1_1/4 block. The output of *DCM*1_1/4 is one of the four different window phase shifted versions of the input clock viz. *DLL*1_0°, *DLL*1_90°, *DLL*1_180°, and *DLL*1_270° with 0.125/0.875 ON-time/OFF-time relationship. The value of MUX1 select from the duty cycle command, $d[3,2]$, selects from the four output versions of *DCM*1_1/4. The output of MUX1, *Sx*, is then further phase shifted by *DCM*2_1/2 block. The windowing feature added to the architecture confines the operation of all high frequency DCM blocks to only one slot of the switching period according to the value of the comparator bits as illustrated in Figure 10c. Therefore, with two

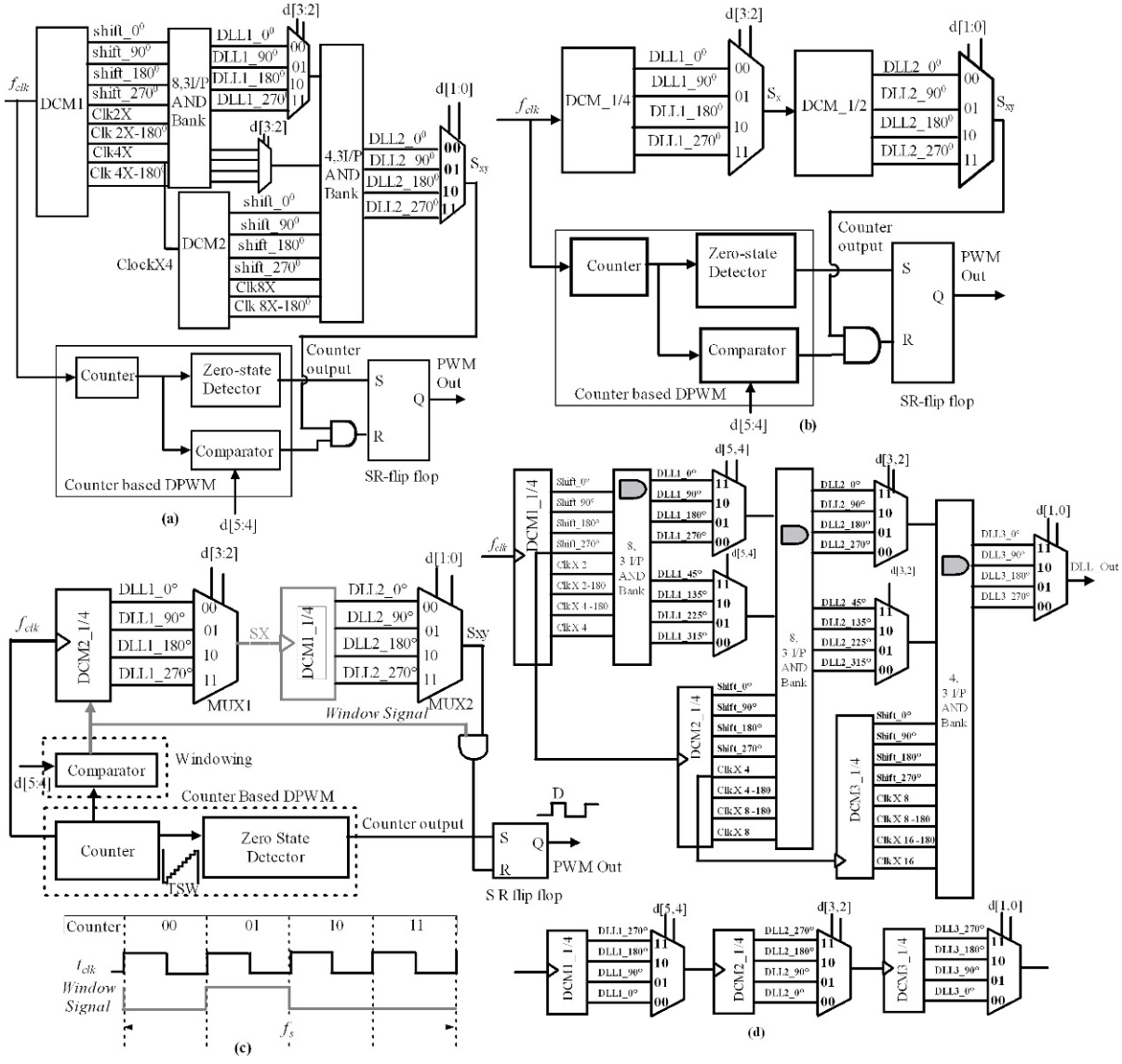

Figure 10. A 6-bit segmented DCM DPWM architecture; (a) Architecture implemented in 4-bit DLL using two DCM blocks and 2-bit counter; (b) Simplified circuit of Figure 10a; (c) Architecture with windowing effect included and corresponding waveforms; (d) The DCM part of Window-segmented DCM based DPWM implementation using three DCM blocks.

cascaded DCM blocks, the maximal clock frequency from *DCM2_½* is twice that of DLL presented in [45]. However, this high frequency clock is only confined to only one slot ($1/2N_{clk}$) of the switching period for *DCM2_½* and the four 3 I/P AND bank. This eventually contributes to the power efficiency as compared with almost all other FPGA-based DPWM architectures.

In DCM based architecture, no clock multiplication is required for either the counter comparator or the first DCM and only twice the input clock operates the second DCM block. Whereas, the DLL block and the counter comparator part in [45] (Figure 8b) operate at a frequency four times higher than the input clock. Secondly, the counter comparator part and the DCM blocks are operated at the same clocking source in DCM DPWM. That means the SR flip flop is set and reset using the same clock at the same rising edge of the clock which eliminates any non linearity aroused in DLL based DPWM [45] due to different clocking. In addition, DCM DPWM benefits from the high resolution and high precision phase shifting

capabilities of the DCM blocks rather than DLL circuits, which determine the linearity of the duty cycle. Therefore, a highly linear duty cycle generation is guaranteed using the accurate DCM circuits.

A numerical example illustrating the operation of the circuit is shown in Figure 10c [53]. Figure 10d shows the complete and simplified DCM (with window-segmented DCM) part for a three DCM-based DPWM with the synchronous part of the counter comparator is basically the same as shown in the full and simplified circuit in Figures 10a and Figure 10c, respectively.

4. Conclusions

Review of DPWM architectures reveals that, a high-frequency; high-resolution DPWM is a key component for successful realization of practical digital control for high-frequency switching power converters. The DPWM resolution is problematic mostly due to two reasons. The first is the need of resolution of DPWM higher than that of ADC to avoid limit cycling effect. The second is that DPWM resolution is inversely proportional to the switching frequency. This limits the use of digital control for high switching frequencies (over 1 MHz). Furthermore, it is noticed that the resolution obtained with classical techniques is not enough for high frequency switching. New techniques need to be developed to increase the resolution of the DPWMs for making them suitable for high frequency applications for better control of output. FPGA based implementations requires advance resources like DLL and DCM to increase the resolution of PWMs. The latest techniques used to increase the resolution of PWMs and consequently, the new DPWMs based on the FPGA advanced resources are explained in detail. These DPWMs provided high resolution, low power consumption for achieving efficient high frequency switching.

Authors through this review have brought out the groundwork and the related references for the benefit of readers and researchers targeting at the different DPWM architectures, the critical component of the digital controller, of the DC-DC converters.

References

[1] Peng H., Prodic A., Alarcon E., Maksimovic D., Modeling of quantization effects in digitally controlled DC-DC converters, IEEE Trans. Power Electronics., 22 (1), 2007, 208-215.

[2] Patella B. J., Prodic A., Zirger A., Maksimovic D., High-Frequency Digital PWM Controller IC for DC-DC Converters, IEEE Trans. Power Electronics., 18(1), 2003, 438-446.

[3] Peterchev A. V., Xiao J., Sanders S. R., Architecture and IC Implementation of a Digital VRM controller, IEEE Trans. Power Electronics., 18(1), 2003, 356-364

[4] Duan Y., Jin H., Digital controller design for switching mode power converters, In: Proceeding of the IEEE Applied Power Electronics Conference and Exposition (APEC'99), 1999, 2, 667-673.

[5] Martin T. W., Ang S. S., Digital control for switching converters, In: Proceeding of the IEEE International symposium on Industrial Electronics, 1995, 2, 480-484

[6] Liu Y.F., Sen P. C., Digital control of switching power converters, In: Proceeding of the IEEE International Conference on control applications, 2005, 635-640

[7] Al-Atrash H., Batarseh I., Digital Controller Design for a Practicing Power Electronics Engineer,In: Proceeding of the IEEE Applied Power Electronics Conference and Exposition (APEC'07), 2007, 34-41

[8] Prodic A., Maksimovic D., Mixed-signal simulation of digitally controlled switching converters, In: Proceeding on IEEE workshops on Computer in Power Electronics, 2002, 100-105

[9] Peterchev A. V., Sanders S. R., Quantization resolution and limit cycle in digitally controlled PWM converters, In: Proceeding of the IEEE Power Electronics Specialists Conference (PESC'01), 2001, 465-471

[10] Syed A., Ahmed E., Maksimovie D., Digital PWM controller with feed-forward compensation, In: Proceeding of the IEEE Applied Power Electronics Conference and Exposition (APEC'04), 2004, 60-66

[11] Lukic N, Rahman N, Prodic A., Multibit $\Sigma - \Delta$ PWM Digital Controller IC for DC-DC Converters Operating at Switching Frequencies Beyond 10 MHz, IEEE Trans. On Power Electronics., 22(5), 2007, 1693-1707

[12] Lukic Z., Wang K., Prodic A., High-Frequency Digital Controller for DC-DC Converters Based on Multi-Bit Sigma-Delta Pulse-Width Modulation, In: In: Proceeding of the IEEE Applied Power Electronics Conference and Exposition (APEC'05), 2005, 1, 35-40

[13] Wang Q., Zhang D. L., All digital DC-DC converters on FPGA, In: Proceeding of the IEEE International Conference on Intelligent Computation Technology and Automation, 2008, 11-15

[14] Guo S., Lin-Shi X., Allard B., Gao B. L. Y., Ruan Y., High-Resolution digital PWM Controller for High-Frequency Low-Power SMPS, In: Proceeding of the IEEE European conference on Power Electronics and Applications (EPE'09), 2009, 1-9

[15] Guo S., Gao Y., Xu Y., Lin-Shi X., Allard B., Digital PWM Controller for High-Frequency Low-Power DC-DC Switching Mode Power Supply, In: Proceeding of the IEEE Power Electronics and Motion Control Conference (IPEMC'09), 2009, 1340-1346

[16] Tan S. C., Lai Y. M., Cheung M. K. H., On the practical design of a sliding mode voltage controlled buck converter, IEEE Trans. Power Electronics., 20 (1), 2005, 425-437

[17] Orosco R., Vazquez N., Discrete Sliding Mode Control for DC/DC Converters, In: Proceeding of the IEEE International Power Electronics Congress (CIEP'00), 2000, 231-236

[18] Ahmed M., Kuisma M., Tolsa K., Silventoinen P., Implementation of Sliding Mode control for Buck Converter, In: Proceeding of the IEEE Power Electronics Specialists Conference (PESC'03), 2003, 2, 634-637

[19] Lentijo S., Pytel S., Monti A., Hudgins J., Santi E., Simin G., FPGA Based Sliding Mode Control for High Frequency Power Converters, In: In: Proceeding of the IEEE Power Electronics Specialists Conference (PESC'04), 2004, 5, 3588-3592.

[20] Tan S-C., Lai Y. M., Tse C. K., Cheung M. K. H., A Fixed-Frequency Pulse width Modulation Based Quasi-Sliding-Mode Controller for Buck Converters, IEEE Trans. on Power Electronics., 20 (6), 2005, 1379-1392

[21] Tan S-C., Lai Y. M, Cheung M. K. H., Tse C.K., On the Practical Design of a Sliding Mode Voltage Controlled Buck Converter, IEEE Trans. on Power Electronics., 20 (2), 2005, 425-437

[22] Abu-Qahouq J. A., Mao H., Batarseh I., Multiphase Voltage-Mode Hysteretic Controlled DC-DC Converter with Novel Current Sharing, IEEE Trans. Power Electronics., 19(6), 2004, 1397-1407

[23] Li P., Bashirullah R., A DLL Based Multiphase Hysteretic DC-DC Converter, In: Proceeding of the IEEE International symposium on Quality Electronic Design (ISQED'07), 2007, 98-101.

[24] Corradini L., Bjeletic A., Zane R., Maksimovic D., Fully Digital Hysteretic Modulator for DC-DC Switching Converters, IEEE Trans. on Power Electronics., 99, 2010, 1-1

[25] Schrom G., Hazucha P., Hahn J.,Gardner D. S., Bloechel B. A., Dermer G., Narendra S. G., Karnik T., De V., A 480-MHz, multi-phase interleaved buck DC-DC converter with hysteretic control, In: Proceeding of the IEEE Power Electronics Specialists Conference (PESC'04), 2004, 6, 4702-4707

[26] Abu-Qahouq J. A., Lou J., Batarseh I., Voltage regulator module with interleaved synchronous buck converters and novel voltage-mode hysteretic control,In: Proceeding of the IEEE International Midwest Symposium on Circuits and Systems (MWSCAS'01), 2001, 2, 972-975

[27] Mohan N., Undeland T., Robbins W., Power Electronics: Converters, Applications and Design, 2nd ed., Wiley, 1995.

[28] Dananjayan P., SriRam V., Chellamuthu C., A flyback constant frequency ZCS-ZVS quasi-resonant converter, Journal of Microelectronics, 29, 1998, 495-504

[29] Ofoli A., Rubaai A., Real-Time Implementation of a Fuzzy Logic Controller for DC-DC Switching Converters., In: Proceeding of the IEEE Annual meeting Conference on Industrial Applications(IAS'05), 2005, 1292-1299

[30] Ramos R., Roset X., Mhuel A., Implementation of Fuzzy Logic Controller for DC/DC Converters Using Field Programmable Gate Array (FPGA), In: Proceeding of the IEEE conference on Instrumentation and Measurement Technology (IMTC'00), 2000.1, 160-163

[31] Gupta T., Boudreaux R. R, Nelms R. M., Hung J. Y., Implementation of a Fuzzy Controller for DC-DC Converters Using an Inexpensive 8-b Microcontroller, IEEE Trans. on Industrial Electronics, 44(5), 1997, 661-669

[32] Wang K., Rahman N., Lukic Z., Prodic A., All-Digital DPWM/DPFM Controller for Low-Power DC-DC Converters, In: Proceeding of the IEEE Applied Power Electronics Conference and Exposition (APEC'06), 2006, 719-723

[33] Syed A., Ahmed E., Maksimovic D., Alarcon E., Digital pulse width modulator Architectures, In: Proceeding of the IEEE Power Electronics Specialists Conference (PESC'04), 2004, 6, 4689-4695

[34] Djemouai A., Sawan M. and Slamani M., New CMOS Integrated Pulse Width Modulator for Voltage Conversion Applications, In: Proceeding of the IEEE International conference on Electronics Circuits and Systems (ICECS '00), 2000, 116-119

[35] Olivier T., Wei G., Wai T. N., A Segmented Digital Pulse Width Modulator with Self-Calibration for Low-Power SMPS, In: Proceeding of the IEEE conference on Electron Devices and Solid-state Circuits (EDSSC'05), 2005, 367-370

[36] Yousefzadeh V., Takayama T., Maksimovi D., Hybrid DPWM with Digital Delay-Locked Loop, In: Proceeding of the IEEE workshops on Computers in Power Electronics(COMPEL'06), 2006, 142- 148

[37] Wei G.-Y., Horowitz M., A low power switching power supply for self-clocked systems, In: Proceeding of the IEEE International Symposium on Low Power Electronics and Design (ISLPED'96), 1996, 313-318
[38] Foley R. E, Kavanagh R. C., Marnane W. P., Egan M. G., An Area-Efficient Digital Pulse width Modulation Architecture Suitable for FPGA implementation, In: Proceeding of the IEEE Applied Power Electronics Conference and Exposition (APEC'05), 2005, 3, 1412-1418
[39] Syed A., Digital pulse width modulators: architectures and feed forward compensation, M.S. Thesis, University of Colorado at Boulder, 2004
[40] Foley R. F., Kavanagh R. C., Marnane W. P., Egan M. G., A Versatile Digital Pulse width Modulation Architecture with Area-Efficient FPGA implementation,In:Proceeding of the IEEE Power Electronics Specialists Conference (PESC'05), 2005, 2609-2615
[41] Wei G.-Y., Horowitz M., A fully digital, energy-efficient, adaptive power-supply regulator, IEEE Journal of Solid-State Circuits., 34(4), 1999, 520-528
[42] Koutroulis E., Dollas A., Kalaitzakis K., High-frequency pulse width modulation implementation using FPGA and CPLD ICs, Journal of Systems Architecture, 52, 2006, 332-344
[43] Xiao J., Peterchev A. V., Zhang J., Sanders S. R., A 4-uA Quiescent-current dual-mode digitally controlled Buck Converter IC for Cellular Phone applications, IEEE Journal Solid-State Circuits, 39(12), 2004, 2342-2348
[44] de Castro A., Todorovich E., DPWM based on FPGA Clock Phase Shifting with Time Resolution under 100 ps, In: Proceeding of the IEEE Power Electronics Specialists Conference (PESC'08), 2008, 3054-3059
[45] Huerta S. C., Castro A. D, Garcia O., Cobos J. A., FPGA based Digital Pulse Width Modulator with Time Resolution under 2 ns, In: Proceeding of the IEEE Applied Power Electronics Conference and Exposition (APEC'07), 2007, 877-881
[46] Batarseh M. G., Al-Hoor W., Huang L., Iannello C., Batarseh I., Segmented Digital Clock Manager-FPGA based Digital Pulse Width Modulator Technique, In: In: Proceeding of the IEEE Power Electronics Specialists Conference (PESC'08), 2008, 3036-3042
[47] Quintero J., Sanz M., Barrado A., Lázaro A., FPGA based Digital Control with High-resolution Synchronous DPWM and High-Speed Embedded A/D Converter, In: Proceeding of the IEEE Applied Power Electronics Conference and Exposition (APEC'09), 2009, 1360-1366
[48] de Castro A., Santa G. S, Huerta C., Cobos J. A., High resolution pulse width modulators in FPGA, In: Proceeding of the IEEE Southern conference on Programmable Logic (SPL'07), 2007, 137-142
[49] Huerta S. C, Castro Ad., Garcìa O., Cobos J. A., FPGA-Based Digital Pulsewidth Modulator With Time Resolution Under 2 ns, IEEE Trans. on Power Electronics, 23(6), 2008, 3135-3141
[50] Ahmad H., Bakkaloglu B., A Digitally Controlled DC-DC Buck Converter Using Frequency Domain ADCs, In: Proceeding of the IEEE Applied Power Electronics Conference and Exposition (APEC'10), 2010, 1871-1874
[51] Gao Y., Zhang S., Xu Y., Gao S., Analysis and Comparison of Three Implementation Methodologies for High-Resolution DPWM, In:Proceeding on the IEEE International Conference on Power Electronics Systems and Applications (PESA'09), 2009, 1-7
[52] Castro Ad., Todorovich E., High Resolution FPGA DPWM Based on Variable Clock Phase Shifting, IEEE Trans. on Power Electronics., 25, 2010, 1115-1119
[53] Batarseh M. G., Al-Hoor W., Huang L., Iannello C., Batarseh I., Window-Masked Segmented Digital Clock Manager-FPGA-Based Digital Pulse width Modulator Technique, IEEE Trans. on Power Electronics, 24(11), 2009, 2649-2660
[54] Dancy A. P., Amirtharajah R., Chandrakasan A. P., High efficiency multiple-output DC-DC conversion for low voltage systems, IEEE Trans.Very Large Scale Integration (VLSI) Systems, 8(3), 2000, 252-263
[55] Dancy A. P., Chandrakasan A. P., Ultra low power control circuits for PWM converters, In: Proceeding of the IEEE Power Electronics Specialists Conference (PESC'97), 1997, 1, 21-27
[56] Goodman J., Dancy A. P., Chandrakasan A. P., An energy/security scalable encryption processor using an embedded variable voltage dc-dc converter, IEEE Journal of Solid-State Circuits, 33(11), 1998, 1799-1809
[57] O'Malley E., Rinne K., A Programmable Digital Pulse Width Modulator Providing Versatile Pulse Patterns and Supporting Switching Frequencies Beyond 15 MHz, In: Proceeding of the IEEE Applied Power Electronics Conference and Exposition (APEC'04), 2004, 1, 53-59
[58] Peterchev A. V., Digital Pulse-Width Modulation Control in Power Electronics Circuits: Theory and Application, Ph.D. thesis, Berkeley, CA, 2005
[59] Gao Y., Guo S., Xu Y., Lin S.X., Allard B., FPGA-Based DPWM for Digitally Controlled High-Frequency DC-DC SMPS, In: Proceeding on the

IEEE International Conference on Power Electronics Systems and Applications (PESA'09), 2009, 1-7

[60] Stratakos A., Sanders S., Broderson R., A low-voltage CMOS DC-DC converter for a portable battery-operated system, In: Proceeding of the IEEE Power Electronics Specialists Conference (PESC'94), 1994, 619-626

[61] Arbetter B., Erickson R., Maksimovic D., DC-DC Converter Design for Battery-Operated Systems, In: Proceeding of the IEEE Power Electronics Specialists Conference (PESC'95), 1995, 1, 103-109

[62] Wu A.M., Xiao J., Markovic D.,Sanders S. R., Digital PWM control: Application in voltage regulation modules, In: Proceeding of the IEEE Power Electronics Specialists Conference (PESC'99), 1999, 77-83

[63] Li J., Qiu Y., Sun Y., Huang B., Xu M., High Resolution Digital Duty Cycle Modulation Schemes for Voltage Regulators, In: Proceeding of the IEEE Applied Power Electronics Conference and Exposition (APEC'07), 2007, 871-876

[64] Xiao J., Peterchev A., Zhang J., Sanders S. R., An ultra-low-power digitally-controlled buck converter IC for cellular phone applications, In: Proceeding of the IEEE Applied Power Electronics Conference and Exposition (APEC'04), 2004, 383-391

[65] Prodić A., Erickson R. W., Maksimović D., Digital Controller Chip Set for isolated DC-DC Power Supplies, In: Proceeding of the IEEE Applied Power Electronics Conference and Exposition (APEC'03), 2003, 866 - 872

[66] Gwee B-H., Chang J., Li H., A Micropower Low-Distortion Digital Pulsewidth Modulator for a Digital Class D Amplifier, IEEE Trans on circuits and systems-II: Analog and Digital Signal Processing, 49(4), 2002, 245-256

[67] McDermott H., A programmable sound processor for advanced hearing aid research, IEEE Trans. Rehabilitation Engineering, 6(I), 1998, 53

[68] Gwcc B. H., Chang J. S., Li H., A micro-power low-distortion digital pulse width modulator for a digital class D amplifier, IEEE Trans. Circuits and Systems-II, 49(4), 2002, 245

[69] Maksimovic D., Zane R., Erickson R., Impact of digital control in power electronics, In: Proceeding of the IEEE International Symposium on Power Semiconductor Devices and ICs (ISPSD'04), 2004, 13-22

[70] Chi-Wai Leng C-W., Yang C-H., Tsai C-H., An Integrated GUI Design Tool for Digitally Controlled Switching DC-DC Converter, In: Proceeding of the IEEE International Conference on Communications,Circuits and systems (ICCCAS'08), 2008, 1324-1327

[71] Liu L-J, Kuo Y-C., Cheng W-C., Analog PWM and Digital PWM Controller IC for DC-DC Converters,In: Proceeding of the IEEE International Conference on Innovative Computing Information and Control (ICICIC'09), 2009, 904-909

[72] Liu Y-F., Meyer E., Liu X., Recent Developments in Digital Control Strategies for DC/DC Switching Power Converters, IEEE Trans. on Power Electronics, 24(11), 2009, 2567-2577

[73] Leng C-W., Yang C-H., Tsai C-H., Digital PWM Controller for SIDO Switching Converter with Time-Multiplexing Scheme, In: Proceeding on the IEEE International Symposium on VLSI Technology, systems and Applications (VLSI-DAT'09), 2009, 52-55

[74] Yousefzadeh V., Advances in Digital Power Control, In: Proceeding of the IEEE International Conference on Telecommunications Energy (INTELEC'10), 2010, 1-8

[75] Yousefzadeh V., Maksimovic D., Sensorless optimization of dead times in DC-DC converters with synchronous rectifiers, IEEE Trans. Power Electronics, 21(4), 2006, 994-1002

[76] Nakamura K., Naka T., Kamata Y., Taguchi T., Shimizu T., Ikeda Y., Nakagawa A., Maksimovic D., 10A 12V 1 chip digitally-controlled DC/DC converter IC with high resolution and high frequency DPWM, In: Proceeding of the IEEE International Conference on Power Electronics and Motion Control (EPE-PEMC'08), 2008, 498-503

[77] Zhang J., Advanced Pulse Width Modulation Controller ICs for Buck DC-DC Converters, Ph.D. Thesis, Berkeley, CA, 2006

[78] Sun A., Tan M., Siek L., Segmented Hybrid DPWM and Tunable PID Controller for Digital DC-DC Converters, In: Proceeding of the IEEE International Symposium on Next-Generation Electronics (ISNE'10), 2010, 154-57.

[79] Rahman N., Wang K., Parayandeh A., Prodić A., Multimode digital SMPS Controller IC for Low-Power Management, In: Proceeding of the IEEE International Symposium on Circuits and Systems(ISCAS'06), 2006, 5327-5330

[80] Virtex FPGA user guide, UG190 (u5.0), June2009 , http://www.xilinx.com

[81] Virtex-5 Libraries Guide for HDL design ISE 10.1, http://www.Xilinx.com

[82] Spartan-3A and Spartan-3A DSP Libraries Guide for HDL Designs ISE10.1, http://www.Xilinx.com