Design of a Digitally Controlled Pulse Width Modulator

for DC-DC Converter Applications

by

Amit Kumar

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved November 2013 by the
Graduate Supervisory Committee:

Bertan Bakkaloglu, Chair
Hongjiang Song
Jennifer Kitchen

ARIZONA STATE UNIVERSITY

December 2013

ABSTRACT

Synchronous buck converters have become the obvious choice of design for high efficiency voltage down-conversion applications and find wide scale usage in today's IC industry. The use of digital control in synchronous buck converters is becoming increasingly popular because of its associated advantages over traditional analog counterparts in terms of design flexibility, reduced use of off-chip components, and better programmability to enable advanced controls. They also demonstrate better immunity to noise, enhances tolerance to the process, voltage and temperature (PVT) variations, low chip area and as a result low cost. It enables processing in digital domain requiring a need of analog-digital interfacing circuit viz. Analog to Digital Converter (ADC) and Digital to Analog Converter (DAC). A Digital to Pulse Width Modulator (DPWM) acts as time domain DAC required in the control loop to modulate the ON time of the Power-MOSFETs. The accuracy and efficiency of the DPWM creates the upper limit to the steady state voltage ripple of the DC – DC converter and efficiency in low load conditions.

This thesis discusses the prevalent architectures for DPWM in switched mode DC – DC converters. The design of a Hybrid DPWM is presented. The DPWM is 9-bit accurate and is targeted for a Synchronous Buck Converter with a switching frequency of 1.0 MHz. The design supports low power mode(s) for the buck converter in the Pulse Frequency Modulation (PFM) mode as well as other fail-safe features. The design implementation is digital centric making it robust across PVT variations and portable to lower technology nodes. Key target of the design is to reduce design time. The design is tested across large Process (+/- 3σ), Voltage (1.8V +/- 10%) and Temperature (-55.0 °C to 125 °C) and is in the process of tape-out.

Dedicated to my Family and Friends

ACKNOWLEDGEMENTS

First and foremost, I convey sincere gratitude to my adviser, Dr. Bertan Bakkaloglu, without whom this thesis would not have been possible. I am very grateful for his guidance, continuous support and friendly attitude for the entire duration of this thesis. He has been easy to approach and has always shown trust and confidence in me that kept me motivated.

Along with Dr. Bertan Bakkaloglu, I would like to wholeheartedly thank Dr. Hongjiang Song and Dr. Jennifer Kitchen for being a part of my defense committee in spite of their busy schedule. I benefit a lot from insightful discussions with Dr. Song on the design of Delay Locked Loop (DLL), and his invaluable suggestions for the entire duration of the project.

I want to thank all my professors whose coursework helped me in enhancing my knowledge which proved essential in completing this work.

I would also like to thank my colleagues Mr. Vivek Parsuram, Mr. Karthik Pappu, Mr. Abhiram Mummareddy, and Mr. Navankur Beohar for the frequent discussions throughout the duration of project that helped shaping it. The course of the thesis helped in building a friendship and trust I will always cherish. I want to thank my dear friends Mr. Ashish Senapati and Mr. Daipayan Chakrabarti for their help and support. I want to thank Mr. James Laux for his endless support in systems and software.

Last but not the least, I would like to extend special thanks to Dr. Philippe Adel and Jet Propulsion Laboratory - NASA for providing financial support to the project.

TABLE OF CONTENTS

Page

CHAPTER

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

1 INTRODUCTION

1.1 What is a DC – DC Converter?

A DC – DC converter is a class of power converter circuit(s) that converts the source of direct current from one voltage level to another. They provide a stable supply voltage, independent of the variation in the input voltage, load current, temperature and time. A battery is a convenient means to provide a portable supply voltage. The output voltage of a battery changes (decreases during discharging, increases when charging) while using it. As a result we need a DC – DC converter that helps in decoupling the effect of voltage variation of the battery output and provides a constant voltage supply to the load. In portable devices we require to change the supply voltage to a lower value to reduce the power dissipation in circuits. A DC – DC converter is used to achieve such lowering (or step – down) of supply voltages and increases efficiency of the system. Applications like harvesting solar energy, driving Liquid Crystal Display (LCD) panels, Light Emitting Diode (LED) lighting, and electric cars require supply voltage conversion from low voltage to high voltage. In all such conversions we want to change the supply voltage while maintaining the highest operating efficiency.

1.2 Types of DC – DC Converters

Many methods exist to achieve supply voltage conversions. A linear conversion of a supply voltage can be done using a Low Dropout Regulator (LDO) to a lower supply voltage level. With the absence of any switching behavior in the LDO, it has no switching ripple at the output voltage (as seen in switched mode converters). LDO also demonstrates very good supply noise immunity.

But they have two major limitations:

1. The voltage conversion with the LDO is always step – down. It means that the output voltage of the LDO is always lower than input voltage creating a limitation to its usage.
2. Low efficiency as the effective power utilized in the LDO is limited by the ratio of output voltage to the input voltage.

The supply conversion can also be done by storing the input energy temporarily in inductors or capacitors and later releasing it at different voltage. Such converters are commonly referred as switched mode voltage converters. They show higher efficiency (up to 98%) especially at high load conditions compared to their linear counterparts. Some of the primary challenges posed by such a topology are design complexity, switching noise and cost.

1.2.1 Buck Converter and Boost Converter

A buck converter is a switched mode DC – DC converter where we require lowering of the supply voltage without affecting the efficiency. It is often referred as a step down converter as the output voltage is lower than the input voltage. Figure 1 shows the conceptual diagram of a buck converter. The steady state output of the converter is controlled by the pulse width of the signal applied to the switch S1.

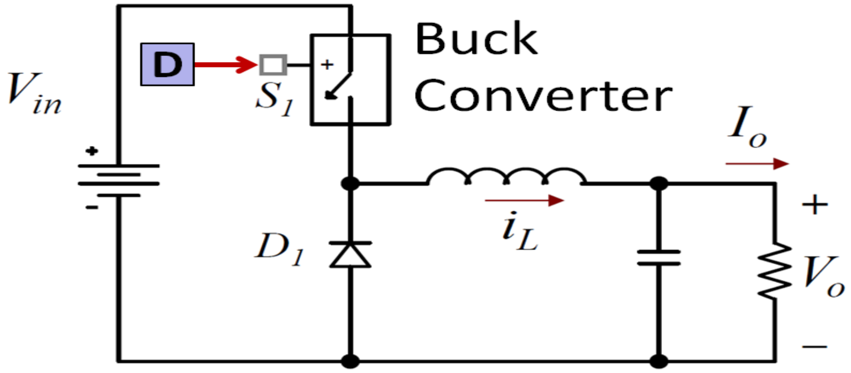

Figure 1: Conceptual diagram of a buck converter

For applications like harvesting solar energy, driving LCD panels, LED lighting, and electric cars that require supply voltage conversion from low voltage to high voltage, also referred to as step-up conversion, we require a boost converter. The conceptual diagram of a boost converter is shown in Figure 2. Table 1 lists major design parameters of buck converter and a boost converter. In steady state, input voltage (V_{IN}) and output voltage (V_O) are related by the duty cycle 'D'.

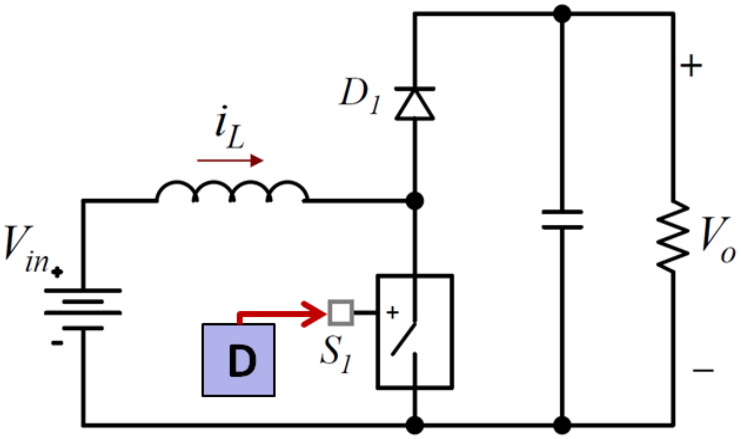

Figure 2: Conceptual diagram of a boost converter

	Buck Converter	**Boost Converter**
$\frac{V_O}{V_{IN}}$	D	$\frac{1}{(1-D)}$
L_{MIN}	$\frac{V_O \times (1-D) \times T_S}{\Delta I_L}$	$\frac{D \times T_S \times V_{IN}}{\Delta I_L}$
C_{MIN}	$\frac{\Delta I_L \times T_S}{8 \times \Delta V_O}$	$\frac{I_O \times D \times T_S}{\Delta V_O}$

Table 1: Buck and boost converter design equations

We can see that ON time or duty-cycle of the signal connected to switch, S1, controls the switched mode converters' operation and dictates steady state and transient characteristics of the converter. This ON time or duty cycle control is conveniently achieved using Pulse Width Modulation (PWM) generator and is explained at length in the following sections.

1.3 PWM in DC – DC Converters

Pulse width modulation is essential in the realization of a switched mode DC – DC converter. The duty cycle (or pulse width) of the signal controlling the Power-MOSFET (represented by S1 in the Figure 1 and Figure 2) provides a way to control the steady state as well as the transient behavior of the converter. Since the input/control to the pulse width modulator (PWM) comes from the controller of the converter, the implementation of PWM block depends closely on the implementation of the controller. As an example, a DC – DC converter with a digital controller will provide binary data to the PWM block, which is then required to be converted to a proportional pulse width. Hence, we require a digital to PWM converter for this DC – DC converter.

1.3.1 Analog PWM Generator

Traditionally, switching DC – DC converters were controlled using analog techniques as they were simple to implement and helped in maximizing the power efficiency. They also offer high linearity in the PWM conversion. The digital counterparts were much slower and much less efficient in terms of power to offer a competitive solution. The emergence of low power, fast and inexpensive silicon processing power over the past decade has kindled the interest in processing in digital domain and hence the use of a digital controller in the switched mode DC – DC converters. Figure 3 shows model of an analog controller in a switched mode buck converter. The key circuits of an analog PWM controller are high gain operational amplifier, voltage controlled oscillator, ramp generator and a comparator. Figure 4 demonstrates how the output of an analog PWM generator (shown in Figure 3) changes with the change in input. As the output of the error amplifier changes from $V_{O,\ MIN}$ to $V_{O,\ MAX}$, the pulse width of the output of the DPWM changes from 0% to 100%.

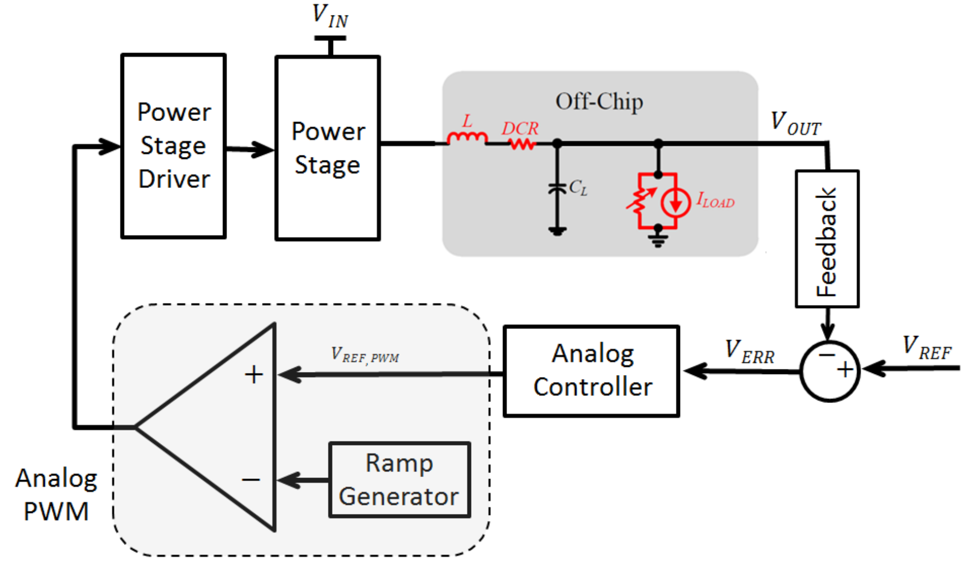

Figure 3: Block level diagram of a buck converter with analog control loop

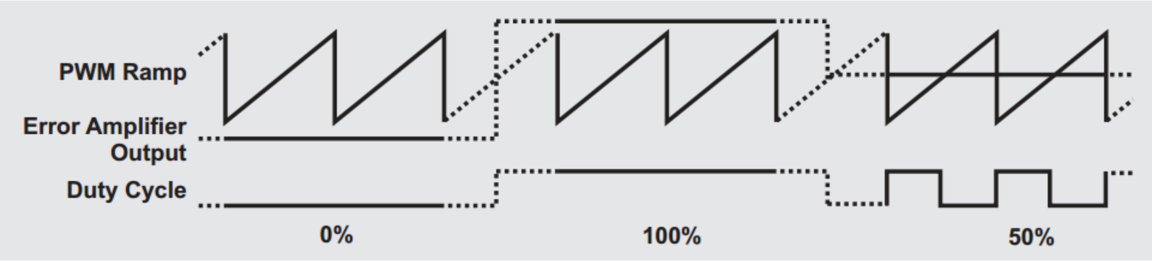

Figure 4: Pulse width modulation by an analog PWM generator [8]

As Figure 4 shows, the output of the Analog PWM stays high as long as the output of the error amplifier stays higher than the PWM ramp signal.

Figure 5 shows the conceptual block level diagram of an Analog PWM [2][9]. F_S is the switching frequency of the DC – DC converter.

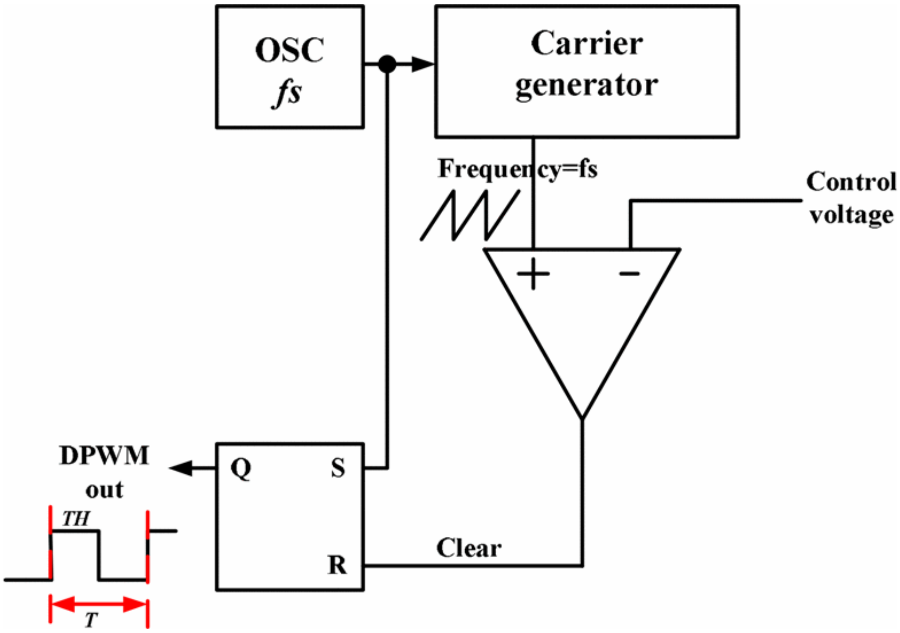

Figure 5: Conceptual block diagram of analog PWM [9]

Simplicity, linearity and ease of realization are primary traits of this architecture. However, this architecture suffers from sensitivity to noise and process, voltage and temperature (PVT) variations.

1.3.2 Digital PWM Generator

The block level diagram of a digitally controlled power converter is shown in Figure 6. Since the processing is done in digital domain, an ADC is required between the output of the converter and the controller. Hence, adding some silicon and design overhead. The accuracy requirement of the ADC depends on the output voltage ripple specification of the converter as explained in Section 2.1.1. Digital compensator and DPWM are the other important blocks when a digital controller is used in a converter.

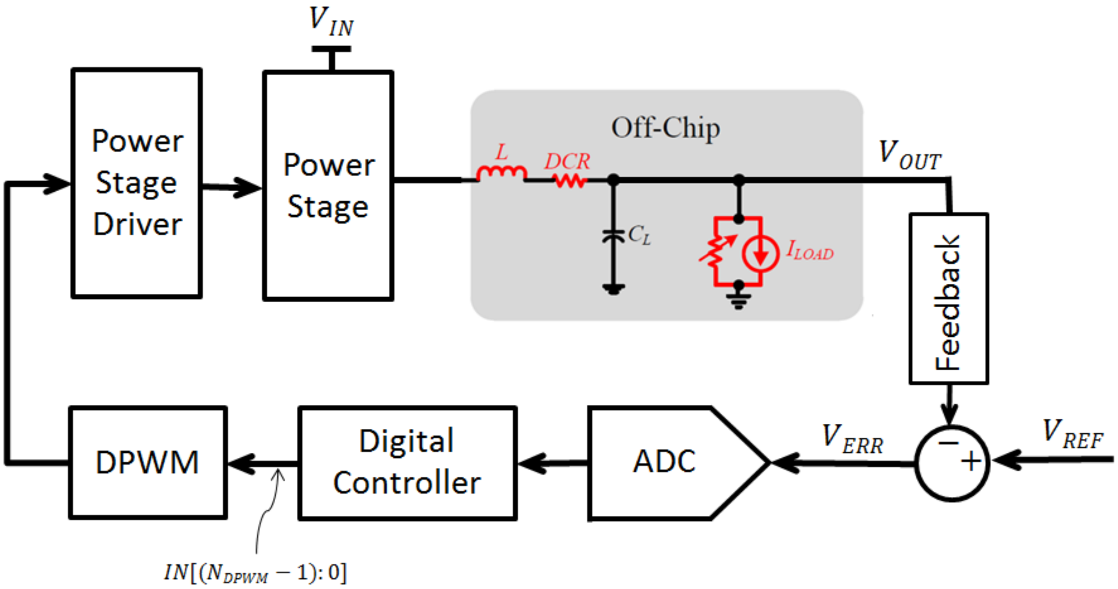

Figure 6: Block diagram of a digitally controlled switching power converter [3]

Digital PWM acts as a time domain DAC for the controller. One of the major advantages of the digital controller over their analog counterpart is that it allows programmability in the controller 'on the fly' to compensate the changes in the operating environments, component aging, capacitive loading etc. In addition, it is relatively much easier to incorporate features like debug modes to enable failure analysis, and switching to low power modes in low load conditions with the digital controller. The digital implementation also benefits from technology scaling and becomes increasingly efficient in terms of area and power and as a result reduces the cost of implementation. There are two major sections of a digital PWM:

1. A block, or a group of blocks that generate $2^{N_{DPWM}}$ equally spaced transitions in the switching time period, T_{SW}, and
2. A supporting circuit that integrates number of unit delays (of time-steps) and generates a pulse-width corresponding to the input of the DPWM.

Figure 7 shows the conceptual block level diagram of a digitally controlled switching power converter [2][9].

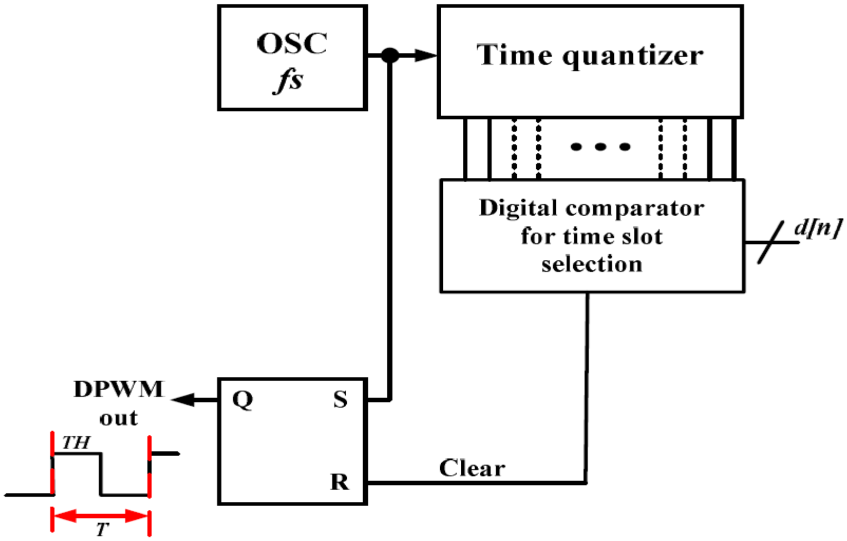

Figure 7: Conceptual block diagram of digital PWM [9]

There are four prevalent architectures of digital PWM [2]:

1. DPWM Type I : Counter Method
2. DPWM Type II : Delay Line
3. DPWM Type III : Hybrid DPWM
4. DPWM Type IV : Σ-Δ – DPWM

Each type is explained in the following sections. Area, power, linearity and output voltage ripples are some of the major matrix while evaluating the architectures.

1.3.2.1 DPWM Type I: Counter Method

A simple way to implement a digital to time conversion is by using a counter. The idea is to count the time-steps using a fast clock. The clocking frequency (F_{CNT}) of the counter depends on two factors:

1. Switching frequency (F_{SW}) of the converter
2. Resolution (Number of bits, N_{DPWM}) of the DPWM.

The frequency of the clock used in the counter (CK_{CNT}) is given by

$$F_{CNT} = 2^{N_{DPWM}} \times F_{SW} \qquad \text{Equation 1}$$

Figure 8 shows the block level design of a counter based DPWM. At the rising edge of the switching clock, CK_{SW}, the counter starts counting, output flip-flop gets a 'set' signal and DPWM output goes to high. The counter output is continuously compared with (here it is subtracted from) the DPWM binary input and comparator output is used to generate the reset for the output flip-flop. DPWM output stays high until the counter output is less than the input DWORD [N_{DPWM} -1:0], generating a pulse width equal to DWORD [N_{DPWM} -1:0] * T_{CNT}, hence generating a pulse – width proportional to the DPWM input.

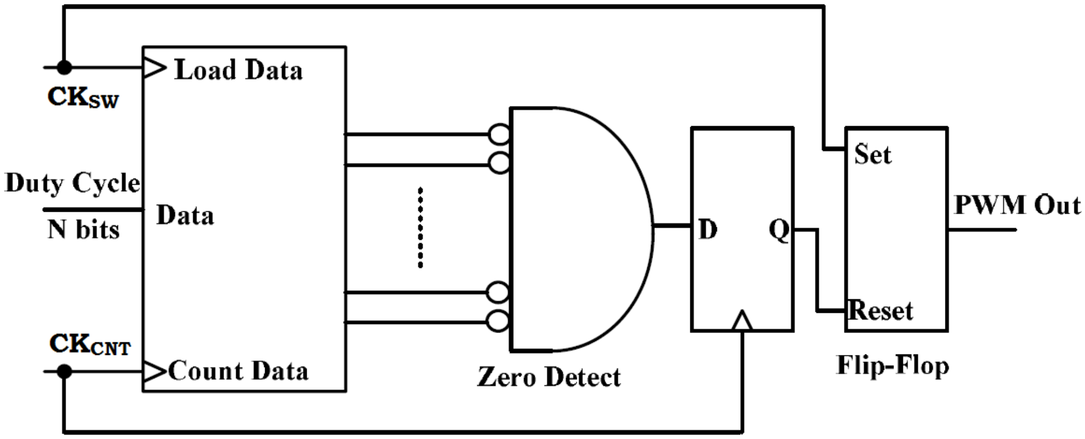

Figure 8: Counter based DPWM [2]

The architecture is inherently simple, efficient and robust to PVT variations due to digital implementation. It requires an N-bit counter and an N-bit digital comparator to implement an N-bit DPWM. As explained in the Equation 1, with the increase in the resolution (N_{DPWM}) and switching frequency, it requires a very fast clock. This is a major drawback of this architecture. For example, if we want to design a 10-bit linear DPWM for a DC – DC converter with a switching frequency of 1.0MHz, we need a 1.024 GHz clock. As a result, we will need a clock generator (using PLL) inside the chip that adds to the complexity and the efficiency of the design in terms of area and power.

1.3.2.2 DPWM Type II: Delay Line

Another simple way to implement time-steps is by using $2^{N_{DPWM}}$ equally spaced transitions generated from matched delay cells. Each delay element generates a time-step or delay equal to $T_{SW}/2^{N_{DPWM}}$. So, we no longer require a really fast clock in the delay line based design as was the case with the counter based approach. This architecture takes advantage of the linear propagation of the clock (CK_{SW}) through the delay cells. Figure 9 shows the block level diagram of a delay line based DPWM.

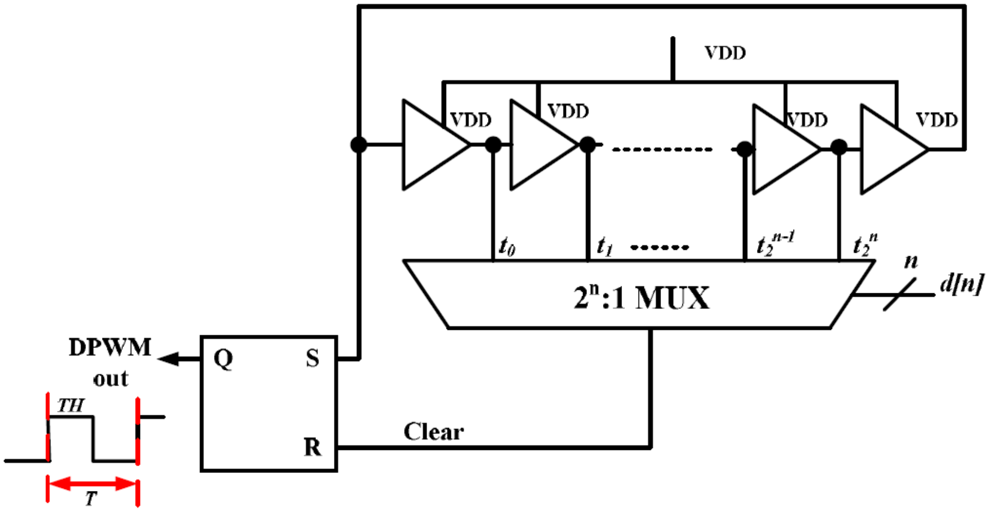

Figure 9: Delay Line based DPWM [2]

At the rising edge of the CK_{SW}, the output flip-flop gets set and then a correct phase of the CK_{SW} is selected by the multiplexer corresponding to the select input of the multiplexer and is used to reset the output flip-flop. Matching of the delay elements, PVT variations and area are some of the major challenges in the delay line based DPWM. The area and power increases exponentially with the increase in the DPWM resolution. With the increase of the number of bits in DPWM, a larger multiplexer is required. Increased skew from input to output and mismatch of the different paths connecting input to output pose some of the major limitations on the design which limits the linearity of the design.

1.3.2.3 DPWM Type III: Hybrid DPWM

The Hybrid DPWM incorporates the merits of counter based and delay line based approaches. This approach divides the design into two sets of delay elements to achieve high DPWM resolution without increasing area and power requirement significantly. It uses a counter for coarse delay generation and a delay locked loop to achieve finer delay resolution between each coarse delay step. Figure 10 shows the block level diagram of a Hybrid DPWM.

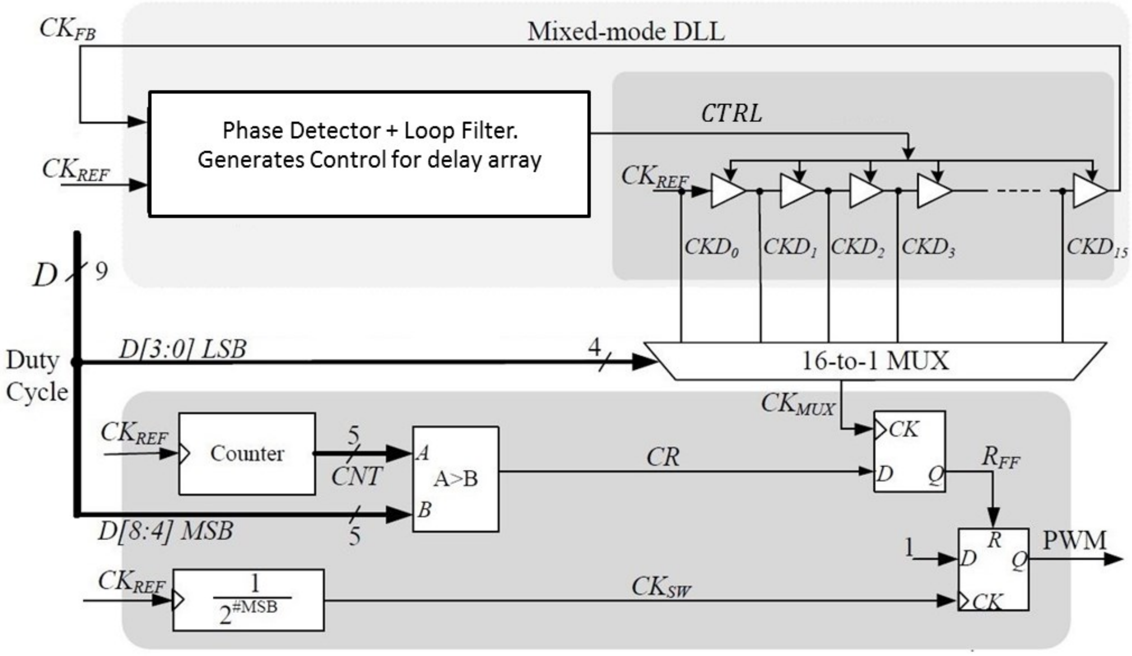

Figure 10: Block level diagram of a Hybrid DPWM

An appropriate balance of resolution is kept between coarse and fine delays such that

1. Coarse delay element is kept large enough so that we don’t require a fast clock (CK_{REF}).
2. The number of delay stages in the delay locked loop (DLL) is small enough that the area and matching requirements are easy to achieve.
3. Smaller number of stages ensures that the multiplexer is small enough and does not deteriorate the linearity of the DPWM due to the mismatch of the input to output delay of the multiplexer.

The use of DLL enables correction of delay variation of the delay arrays due to the PVT variations. This architecture significantly improves area, power and clock requirement for a high resolution DPWM design.

1.3.2.4 DPWM Type IV: Σ-Δ – DPWM

An elegant way to achieve additional higher accuracy requirement is done using time domain averaging. Figure 11 demonstrates the time domain averaging using a 2-bit dither. When averaged over time (here 4*T_{SW}), we are able to achieve sub LSB level (0.25*LSB, 0.5*LSB, and 0.75*LSB) accuracy without increasing the frequency of the switching clock CK_{SW} [7][11]. So, a block that controls the dither is added between the controller and the DPWM to implement the operation. A major challenge to this approach is presence of switching noise at sub-switching frequencies (e.g. $F_{SW}/2$, $F_{SW}/4$ in the case shown in Figure 11) due to this averaging. This noise needs to be filtered the loop band-width of the converter. So, we would need the loop bandwidth of the converter to be much lower than the switching frequency of the converter. As we add more number of bits in the dither, the loop-bandwidth will need to be increasingly lower compared to the switching frequency in order to control the voltage ripple at the converter output. Absence of sufficient filtering of sub-harmonics of CK_{SW} also leads to EMI/EMC issues that need to be taken care of. Use of sigma delta modulator (as shown in the Figure 12) enables shaping of the switching noise and reduces the EMI/EMC issues. Even with sigma-delta based noise shaping, it still required that the bandwidth of the loop be much lower than CK_{SW} due to averaging over time.

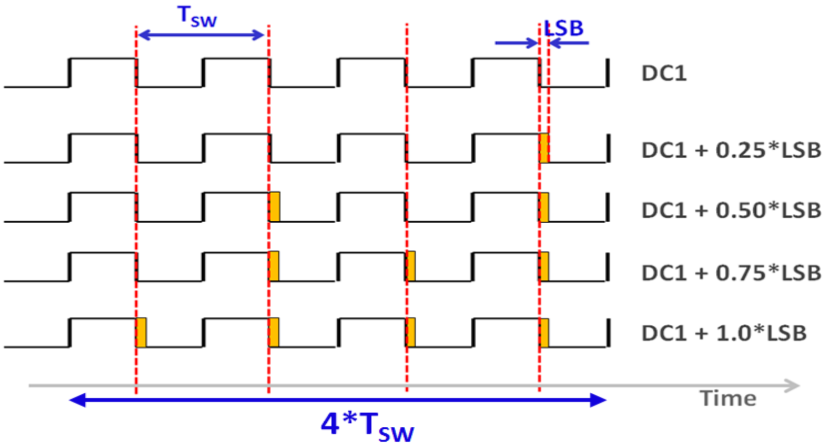

Figure 11: Sub-LSB resolution using a 2-bit time domain averaging (dither)

Additionally, due to time domain averaging, transient response of the sigma-delta or a dither based DPWM get affected. This gets reflected on the over-shoot/under-shoot performance of the converters.

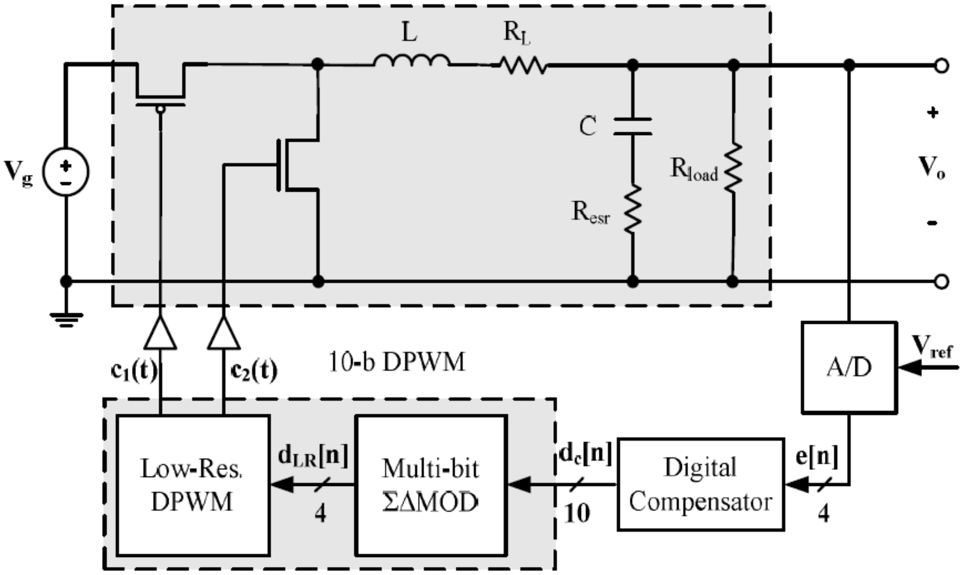

Figure 12: Block diagram of switching converter with Σ-Δ DPWM controller [4]

1.4 Scope of the thesis

The goal of the thesis is to come up with a simple design of a DPWM for switched mode buck converter that is robust and can support a low power mode for the low load and PFM mode. The thesis focuses on the following things:

1. Study of the prevalent architectures of Digital PWM
2. Development of the DPWM architecture for the switched mode buck converter that is efficient and simple to implement
3. Low power mode for the converter in PFM mode
4. Fail safe and debug features

1.5 Thesis outline and organization

In the following chapters, system level constraints, and implementation of a Digital PWM has been discussed.

Chapter 2 discusses the analysis of the design specification and the system architecture of the DPWM used in the design.

Chapter 3 discusses the design specification, design choices and trade-offs of the Delay Locked Loop used in the DPWM. The chapter also discusses and modeling and analysis of the DLL used in the DPWM.

Chapter 4 explains the implementation of the DPWM and presents the results.

2 DIGITAL PULSE WIDTH MODULATOR (DPWM)

This design is targeted to be a part of a switched mode buck converter. As discussed in the previous section, DPWM acts as a digital to analog converter in the switched mode buck converter.

2.1 Design Specification

Digital pulse width modulation produces discrete set of pulse widths (duty cycles) depending on the input codes and the resolution of the DPWM. The pulse width of the DPWM is equal to

$$T_{ON} = \frac{Decimal\ value\ of\ DWORD[N_{DPWM}: 0]}{2^{N_{DPWM}}}\ .\ T_{SW} \qquad \text{Equation 2}$$

Where,

N_{DPWM} = Number of bits in the input word for the DPWM.

DWORD = Input word for the DPWM

T_{SW} = Time period of the switching clock of the converter

For a switched mode buck converter operating in steady state,

$$V_{OUT} = D\ .\ V_{BAT} \qquad \text{Equation 3}$$

$$V_{OUT} = \frac{T_{ON}}{T_{SW}}\ .V_{BAT} \qquad \text{Equation 4}$$

Where,

D = duty cycle of the output of the DPWM

V_{OUT} = Output voltage of the converter

V_{BAT} = Input voltage of the converter

As it is evident from Equation 2, Equation 3, and Equation 4 only a discrete set of output voltage values can be obtained in the steady state from a discrete set of pulse widths from the DPWM. If the desired output voltage does not belong to one of these discrete values, the feedback controller will switch among two or more discrete values of the duty cycle [1].

Figure 13 shows a DPWM used in a digitally controlled voltage mode buck converter. The design of the DPWM is targeted for a switched mode buck converter with the following specifications:

Input Voltage V_{BAT} = 3.3V to 5.0V

Output voltage V_{OUT} = 0.9 V

Output ripple V_{RIPPLE} = 1.0 – 2.0 % of V_{OUT}

Switching Frequency F_{SW} = 1.0 MHz

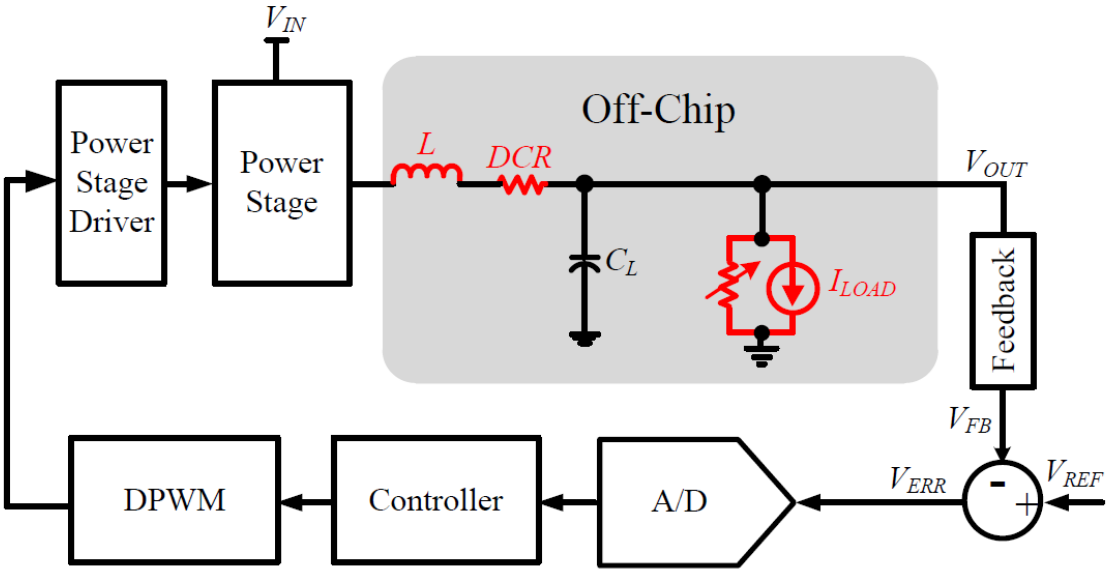

Figure 13: Generic digitally controlled voltage mode Buck converter [6]

2.1.1 Calculation of number of bits for DPWM

Based on the specifications provided for the typical output voltage and the maximum allowed steady state voltage ripple at the output of the converter, we can write:

$$V_{RIPPLE,MAX} = VOUT \times \frac{2}{100} = 0.9 \times \frac{2}{100} = 18.0\, mV \qquad \text{Equation 5}$$

Where, V_{RIPPLE} is the output voltage regulation accuracy.

Since, we don't want the ADC to track the steady state ripple we want the ΔV_{ADC} to be bigger than $V_{RIPPLE,\ MAX}$. The ΔV_{ADC} represents the accuracy of the ADC defined as the minimum change required at the input voltage to the ADC to change the output code by 1. It can be represented as,

$$\Delta V_{ADC} = \frac{FSR}{2^{N_{ADC}}} \qquad \text{Equation 6}$$

Where,

FSR = Full scale range of the ADC, and

N_{ADC} = Number of bits in the ADC.

So,

$$\Delta V_{ADC} > V_{RIPPLE,MAX} > 18.0\, mV \qquad \text{Equation 7}$$

In general, the converter is designed such that the ripple at the output is smaller than $V_{RIPPLE,\ MAX.}$

$$\Delta V_{ADC} \sim 20\, mV \qquad \text{Equation 8}$$

Let $\Delta T_{ON,\ DPWM}$ be the change in the duty cycle of the DPWM with unit change in input of the DPWM then ΔV_{DPWM} is the corresponding change in the target output voltage of the DC – DC converter.

$$\Delta T_{DPWM} = \frac{T_{SW}}{2^{N_{DPWM}}}$$ Equation 9

Where, T_{SW} represents the time period of the DPWM and is also equal to the switching time period of the DC – DC converter.

Thus Equation 4 can be written as,

$$\Delta V_{DPWM} = \frac{\Delta T_{ON,DPWM}}{T_{SW}} \times V_{BAT}$$ Equation 10

The term ΔV_{DPWM} indicates how accurately the output voltage can be regulated by the buck converter in the steady state.

2.1.1.1 Limit Cycles

Limit cycle refers to steady state periodic oscillations seen at the output voltage of the DC – DC converter which is not caused by the PWM activities of the loop. It is hard to predict the amplitude and frequency of the oscillations in the steady state due to limit cycles [7]. As a result, limit cycles are highly undesirable and every attempt is made during design to ensure that it does not occur during the converter operations. One of the scenario that can result limit cycles is when the voltage resolution of the ADC is more than that of the DPWM and there is no DPWM level that maps onto the ADC bin corresponding to the reference voltage V_{REF} (ADC bin corresponding to V_{REF} is referred to as zero error bin). The situation is demonstrated in Figure 14. In steady state, the controller tries to drive V_{OUT} to zero error bin. Since, no DPWM level exists in the zero error bin, V_{OUT} will vacillate between the DPWM levels around zero error bin. The issue gets resolved when we increase the resolution of the DPWM by adding more bits to it as shown in Figure 15.

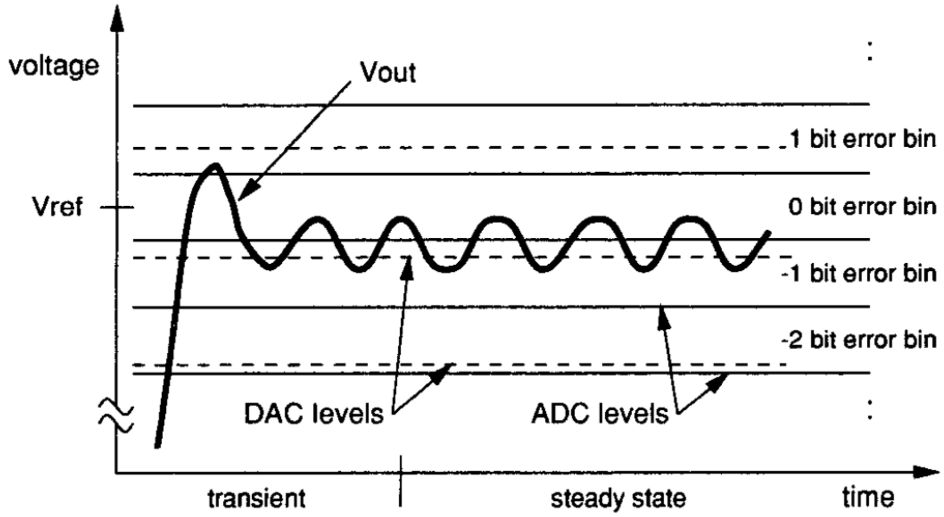

Figure 14: Limit cycles when DPWM resolution is lower than ADC resolution [7]

So, to prevent limit cycle oscillation, we need (assuming same full scale range)

$$N_{DPWM} \geq N_{ADC} + 1 \qquad \text{Equation 11}$$

So, the resolution of the DPWM from Equation 8 and Equation 11

$$\Delta V_{DPWM} = \frac{\Delta V_{ADC}}{2} = 10.0\, mV \qquad \text{Equation 12}$$

So, the minimum number of bits required for DPWM,

$$N_{DPWM} = \log_2(\frac{V_{BAT}}{\Delta V_{DPWM}}) = \log_2 \frac{5.0V}{10.0mV} \sim 9\, Bits \qquad \text{Equation 13}$$

When resolution of the DPWM is increased such that there always exist a DPWM level in the zero error bin, we no longer observe the issue of limit cycles as shown in Figure 15. The limit cycles may occur even when the resolution of the DPWM is lower than that of ADC. But, such discussions are beyond the scope of this document. Please refer to literature [7] for further details.

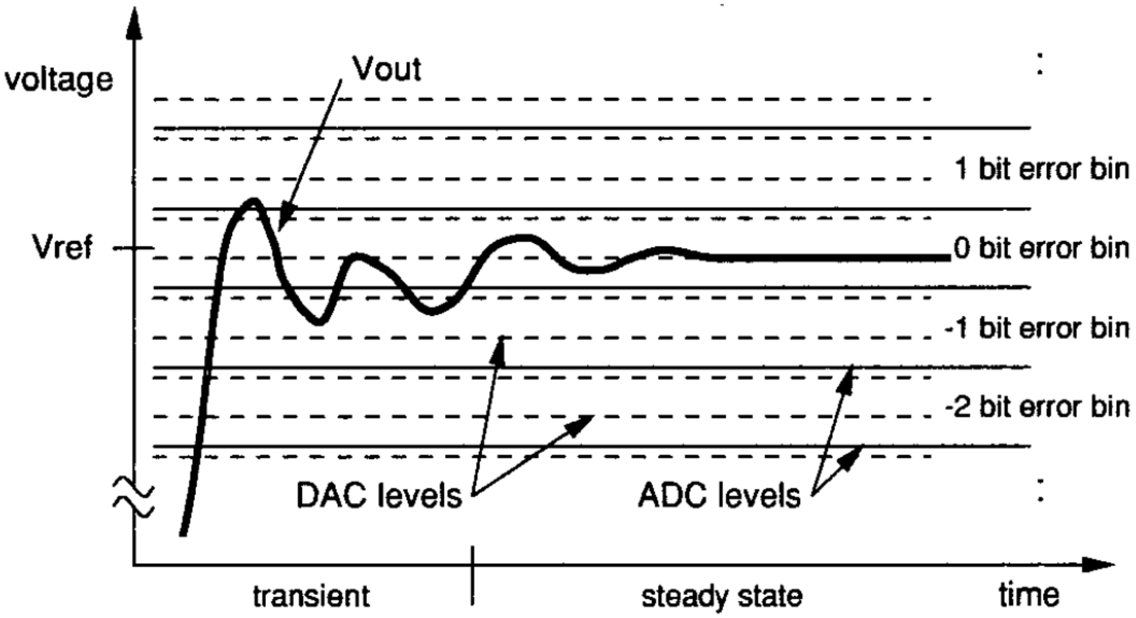

Figure 15: Limit cycle issue is resolved with increased resolution of DPWM

2.1.2 PVT specification of the DPWM

The PVT specification of the Digital to PWM converter is given in Table 2. The DC – DC converter is expected to work at extreme temperature corners (-55°C to 125°C) as indicated below. Sufficient supply margin (+/- 10%) is kept to ensure its performance. The design is targeted to work at +/- 3σ process corners to ensure the robustness across process variations.

Parameter	**Details**	**Min**	**Typ.**	**Max**	**Unit**
Process	Process Corner	-3σ	0σ	+3σ	
Supply Voltage		1.62	1.8	1.98	V
Temperature		-55	27	125	°C
Linearity	Linearity of the PWM	9			ENOB
Switching Frequency	Converter Switching Frequency (F_{SW})		1		MHz

Table 2: PVT corner specification of the DPWM design

2.2 System Architecture of the DPWM

As derived in Equation 13, the targeted DC – DC converter requires a 9-bit DPWM. As discussed in Section 1.3.2, achieving 9-bit accuracy requires 2^9 = 512 matching delay elements in the delay line based DPWM which in turn requires huge area. Delay line based DPWM also suffers from the delay variation due to PVT variations. The design will also require a 9-bit multiplexer with a select to output delay much smaller than ΔT_{DPWM}. PVT variations can be countered with the use of a DLL. However, this approach will require the DLL to have 512 phases, making the design unrealistic.

Counter based approach is relatively simple and efficient in terms of area and power. However, they are only suitable for a converter with lower switching frequency and lower resolution. For a 9-bit accurate DPWM with a switching frequency of 1.0MHz, it requires a counter clock frequency of 512MHz (as shown in the Equation 1). This will require an on-chip PLL adding to the area and power requirement and hence will be much less efficient.

Sigma – Delta DPWM, offer a better resolution at the cost of complexity with the additional sigma – delta block in the design. It also increases the output voltage ripple and slows the transient response of the converter. Considering the trade – offs between the architectures and target specifications a Hybrid DPWM was found to be the most suited choice.

Figure 16 shows the architecture of the DPWM used in the design. There are two major sections of the DPWM:

1. Delay locked loop (highlighted in red)
2. Supporting circuits (highlighted in gray)

The design incorporates a 5–bit counter to account for the 5-bits (higher significant bits, DWORD [8:4]) and a DLL to account for 4-bits (lower significant bits DWORD [3:0]) of the digital input (DWORD [8:0]). Thus, a 9-bit accurate digital-to-time conversion is achieved.

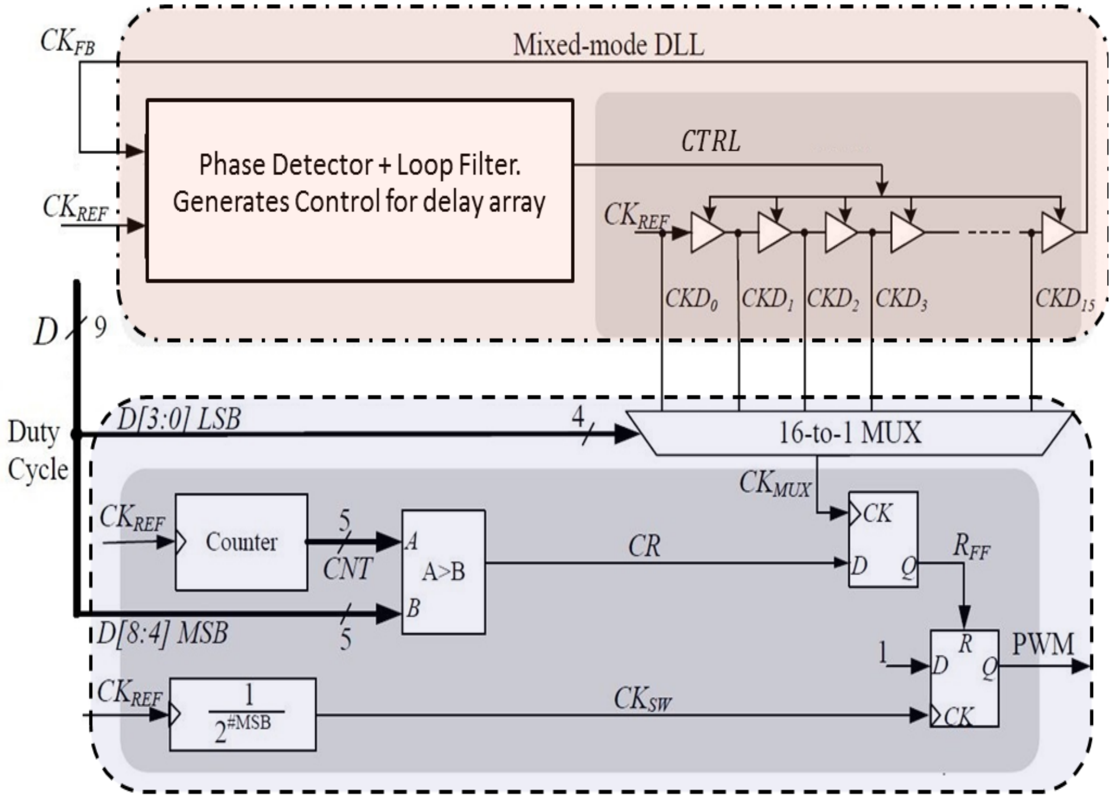

Figure 16: The Architecture of a Hybrid DPWM [6]

The timing diagram of the DPWM is shown in Figure 17. The following sequence of event occurs in DPWM operation:

1. Counter starts counting with the rising edge of the 1.0 MHz reference clock, CK_{SW} and the output of DFF_02 PWM goes high.
2. The output of counter CNT [4:0] is continuously compared with D [8:4] and when they are equal CR becomes high.
3. Based on D [3:0], the mux selects the correct phase of the clock, CK_{MUX}, which in turn is used to sample CR using DFF_01.
4. When CR goes high and CK_{MUX} samples it, R_{FF} goes high.
5. R_{FF} resets the DFF_02 generating a pulse proportional to D [8:0].

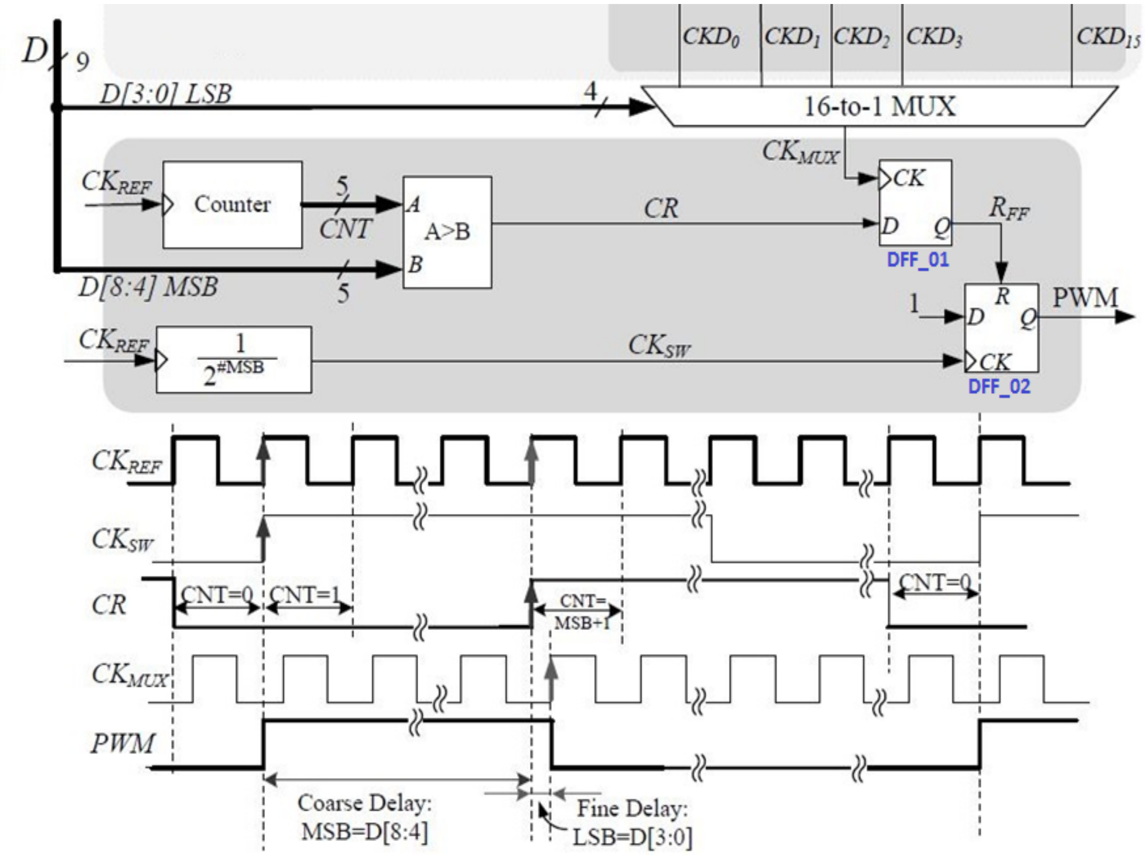

Figure 17: The timing diagram of the Hybrid DPWM

2.2.1 Delay Locked Loop (DLL)

2.2.1.1 Overview

Delay locked loop find wide range of applications in clock distribution, clock synthesis, memories, processors, clock and data recovery circuits, communication ICs where it reduces the buffer delay and as a result improves the timing margin of the interface. For the design of DPWM, DLL ensures that per-stage delay of the delay cells does not change with the PVT variations ensuring its robustness.

2.2.1.2 Expectations

The wish list of features for DLL is:

1. Design Simplicity
2. Since we require a 4 bit resolution from the DLL so the number of phases of the DLL = 16.
3. DLL should lock across PVT (as specified in the Table 2)
4. Since the reference clock of the DLL is fixed and the DLL only needs to correct the variation of delay with V and T, the bandwidth of the loop should be small.
5. When PFM mode is enabled, the DLL should go into a low power mode with minimum or no switching. The context of the DLL should be preserved when going into PFM mode to reduce the ramp-up time when coming back to PWM mode.

2.2.2 Supporting Circuits

2.2.2.1 Overview

The main functionality of the supporting circuit is to generate larger time steps using a counter and then integrate the delay contributed by the counter and DLL to generate the pulse-width proportional to the input of the DPWM.

2.2.2.2 Expectations

The design of the supporting circuit should ensure that:

1. Reset and input clock are glitch-free due to their edge alignment.
2. Delay steps produced by the counter are very accurate.
3. Delay corresponding to the counter and the DLL is integrated without adding significant errors that can limit the linearity of the design.
4. Design is glitch-free. Special care is taken to ensure that the output of the comparator followed by the counter, and the multiplexer (which selects the correct DLL phase) doesn’t introduce glitch in the design to ensure safe operation.
5. Output of the DPWM is pulled low when DPWM input is zero.

3 DELAY LOCKED LOOP (DLL)

Delay locked loop is used in the DPWM to provide 16 clock phases to generate a 4-bit resolution for each counter step. DLL ensures that the delay steps provided by the delay cells do not change with the process, temperature and supply voltages.

3.1 Design Specification

The design of the DPWM requires a DLL running at 32MHz. The DLL should have 16 phases. The PVT specification of the DLL is same as DPWM, given in Table 2. The design specifications of the DLL are:

1. Design simplicity and portability.
2. Reference clock of DLL = 32MHz
3. Number of phases of the DLL = 16.
4. DLL should lock across PVT (as specified in Table 2)
5. Since the reference clock of the DLL is fixed and the DLL only needs to correct the variation of delay with V and T, the loop bandwidth should be small.
6. Low power operation when converter in PFM mode.
7. Relaxed locking time since other blocks (e.g digital PID controller) take 15 – 20µs after reset to start functioning.
8. Relaxed jitter requirement from the DLL.

3.2 Implementation Selection: Analog DLL vs. Digital DLL

An analog DLL is a common approach to implement clock skewing/de-skewing. It offers extremely low jitter performance critical for all high speed I/O circuits. Implementation of the analog DLL depends heavily with the process capabilities. Hence the design usually changes when migrating from one process to another. In addition, the design is susceptible to PVT variations compared to their digital counterpart. So, analog DLL provides very low jitter performance at the cost of complexity [12].

Digital DLLs are becoming increasingly popular more applications because of their design simplicity and portability as a result reduced design time. They have better noise immunity and can operate at lower supply voltages making the design more attractive at higher technology node. With the advancement of the technology enables a higher resolution improving the performance of the digital DLLs. They have limited jitter performance due to the switching in the control loop.

Due to simplicity of design, robustness of implementation, better portability to higher process nodes and non-stringent requirement on jitter, digital implementation of the DLL is selected for the DPWM. It is easier to implement clock gating for PFM mode without losing the DLL lock in the Digital DLL. Figure 18 shows the conceptual diagram of a digital DLL.

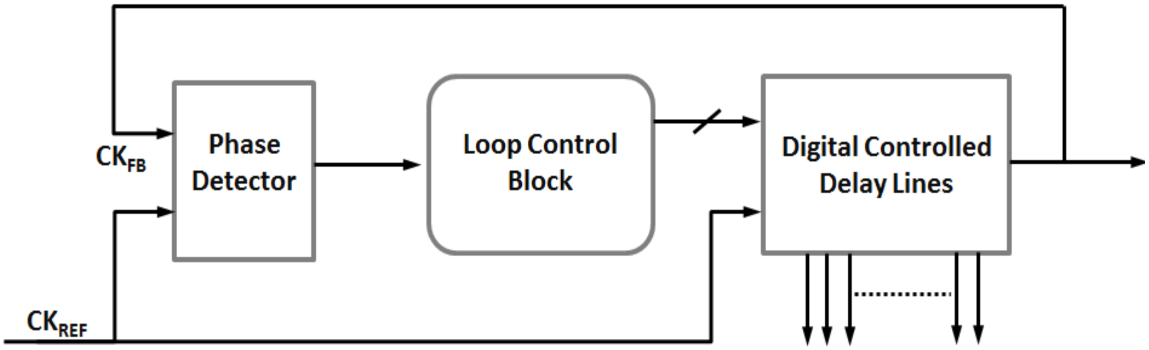

Figure 18: The conceptual diagram of a Digital DLL

Phase detector compares the delay generated by Digitally Controlled Delay Lines (DCDL). Phase detector output is then processed by control loop block and it generates a new command word controlling DCDL to correct the delay.

3.3 Architecture of the DLL used in the DPWM

Figure 19 shows the architecture of DLL used in the design of the DPWM.

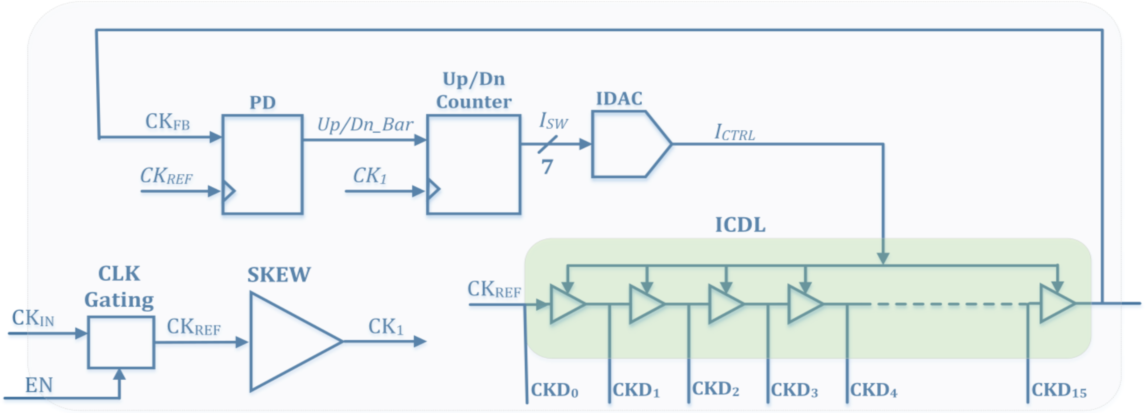

Figure 19: The block level diagram of the mixed mode DLL used for the DPWM

The locking mechanism of the DLL is explained below:

1. The phase detector (PD) compares delay provided by the 16 delay cells (T_{16}) with the time period (T_{REF}) of the reference clock (CK_{REF}). It generates 'high' ('low') when T_{REF} is smaller (larger) than T_{16}.
2. When PD output is 'high' ('low') counter output is decremented (incremented).
3. Counter output controls the output current $I_{CONTROL}$ of Current DAC (IDAC). $I_{CONTROL}$ increases (decreases) with the counter output.
4. Here delay stage is implemented using a Current Controlled Delay Line (ICDL). Delay in ICDL is inversely proportional to control current, $I_{CONTROL}$.
5. When T_{REF} is smaller than T_{16}, PD output goes 'low'. The counter output is then incremented which in turn increases $I_{CONTROL}$. And, since current in the delay lines is copied from the $I_{CONTROL}$, it also increases. As a result, the delay in the ICDL reduces and approaches target delay (T_{REF}).
6. Here, counter holds the context (state) of the DLL.

When buck converter goes into the PFM mode, it uses a comparator based approach to control the loop and bypasses the main controller. So, most of the blocks are in the main controller path are kept in low power mode during PFM. Clock gating is used to achieve minimize power requirement from the DPWM (here DLL). Since, switching power is major contributor to the power requirement of this DLL, clock gating enables large power saving during PFM mode.

Figure 20 shows Simulink Model of DLL loop used in the DPWM. The modeling of the design is done in time domain. The definitions of the blocks/symbols used in the model are listed below:

Di = Delay associated with the reference clock.

Do = Delay associated with the output clock.

Dr = Delay skew need to be added to the input clock to generate the skewed clock at the output of the DLL. So, we want, Do = Di + Dr.

Derr/Qerr = Output of the phase detector.

kD2I = Gain of the IDAC.

ITRIM = Trim bits in the IDAC to take care of the extreme PVT variations (details in the section 4.2.3).

kICDL = Gain of the delay lines, represents the change in the delay per unit change in the current controlling the delay line.

X_PVT = Models PVT variations.

Stage delay of a delay cell implemented using current starved buffer can be expressed as,

$$T_D = \frac{\mathrm{V_{DD}}}{2} \times \frac{\mathrm{C_L}}{\mathrm{I_{CONTROL}}} \qquad \text{Equation 14}$$

Note that the model is non-linear. There are two sources of non-linearity:

1. Phase detector (PD) only outputs 'high' or 'low', and so phase detector output is not proportional to the phase difference of the reference clock (CK_{REF}) and the feedback clock (CK_{FB}).
2. The delay is inversely proportional to the current used to control the delay line (as explained in the Equation 14).

Figure 21 shows the linearized Simulink model for the DLL. Non-linear phase detector is replaced by an attenuation stage. The $\mathrm{I_{CONTROL}}$ and delay relationship (Equation 14) is linearized as:

$$T_D = \mathrm{A} - \mathrm{B} \times \mathrm{I_{CONTROL}} \qquad \text{Equation 15}$$

Where, A and B are constants.

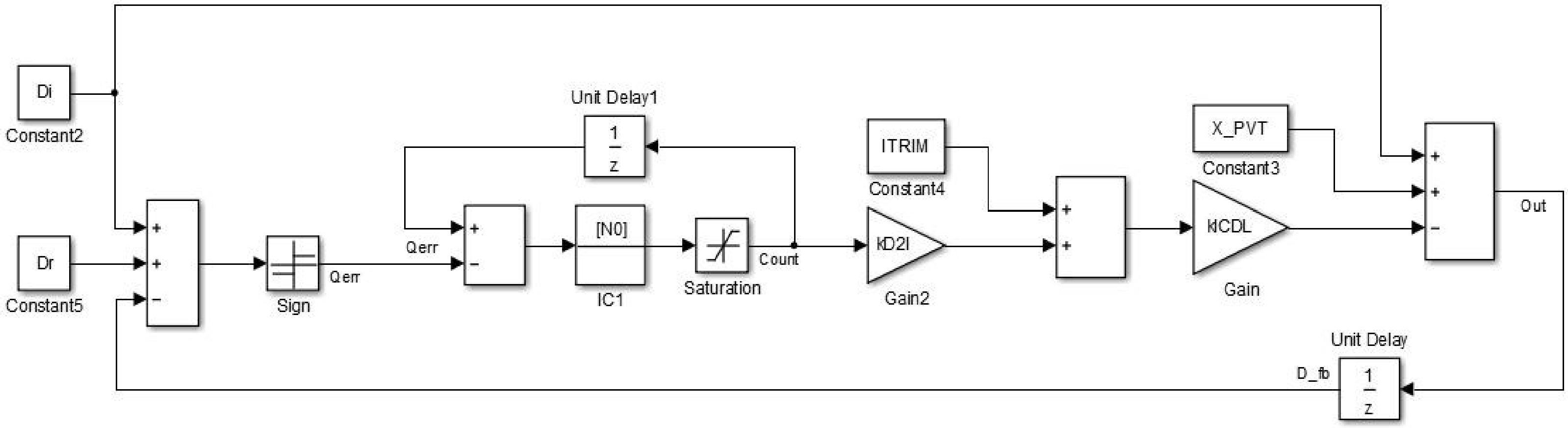

Figure 20: Non-linear model of the Delay Locked Loop (DLL)

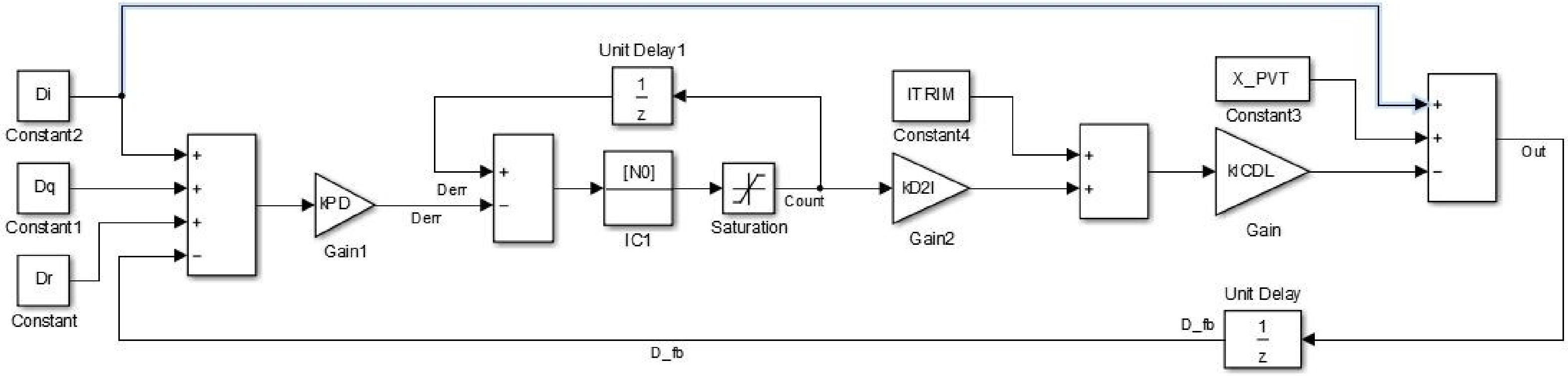

Figure 21: Linearized model of the Delay Locked Loop

Figure 22 shows signal flow diagram of the linearized DLL shown in Figure 21.

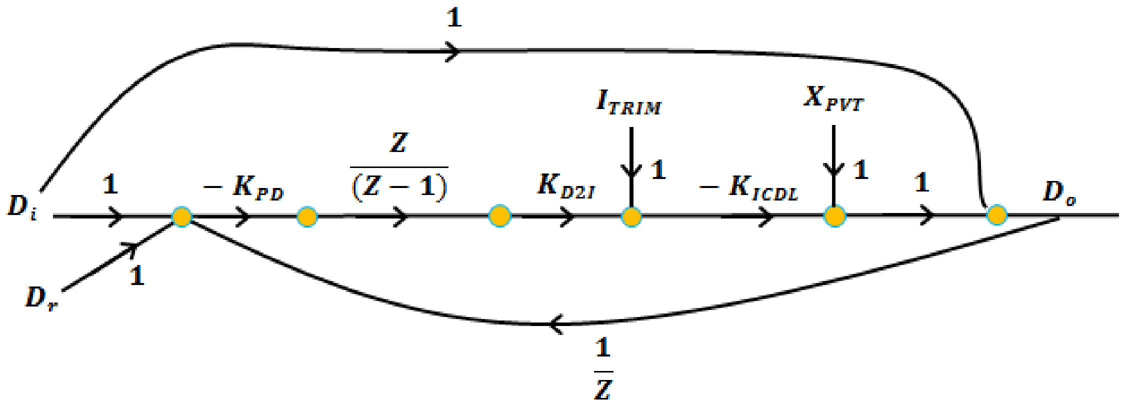

Figure 22: Signal flow diagram of the linearized DLL

The transfer function of the DLL (shown in Figure 22) can be expressed as:

$$D_o = \frac{\mathrm{Z} \times (\mathrm{K}_1 + 1) - 1}{\mathrm{Z} - (1 - \mathrm{K}_1)} \times \mathrm{D}_\mathrm{i} + \frac{\mathrm{Z} - 1}{\mathrm{Z} - (1 - \mathrm{K}_1)} \times \mathrm{D}_1 + \frac{\mathrm{Z} \times \mathrm{K}_1}{\mathrm{Z} - (1 - \mathrm{K}_1)} \times \mathrm{D}_\mathrm{r} \qquad \text{Equation 16}$$

Where,

$$K_1 = \mathrm{K}_{\mathrm{PD}} \times \mathrm{K}_{\mathrm{D2I}} \times \mathrm{K}_{\mathrm{ICDL}} \qquad \text{Equation 17}$$

$$D_1 = \mathrm{X}_{\mathrm{PVT}} - \mathrm{I}_{\mathrm{TRIM}} \times \mathrm{K}_{\mathrm{ICDL}} \qquad \text{Equation 18}$$

D_1 represents the delay (or delay uncertainty) introduced by the blocks internal to the DLL. Gate delays, thermal noise of MOSFETs, supply noise, substrate noise, V_{TH} variation are some of the factors contributing to D_1.

Figure 23 shows the transfer characteristics of the DLL. DLL acts as a high pass filter for the jitter in the input clock (represented by Di), low pass jitter for the jitter in the reference (represented by Dr), and doesn’t filter the jitter internally generated (represented by D1) in the DLL as shown in the Figure 23 as well as explained in the Equation 16.

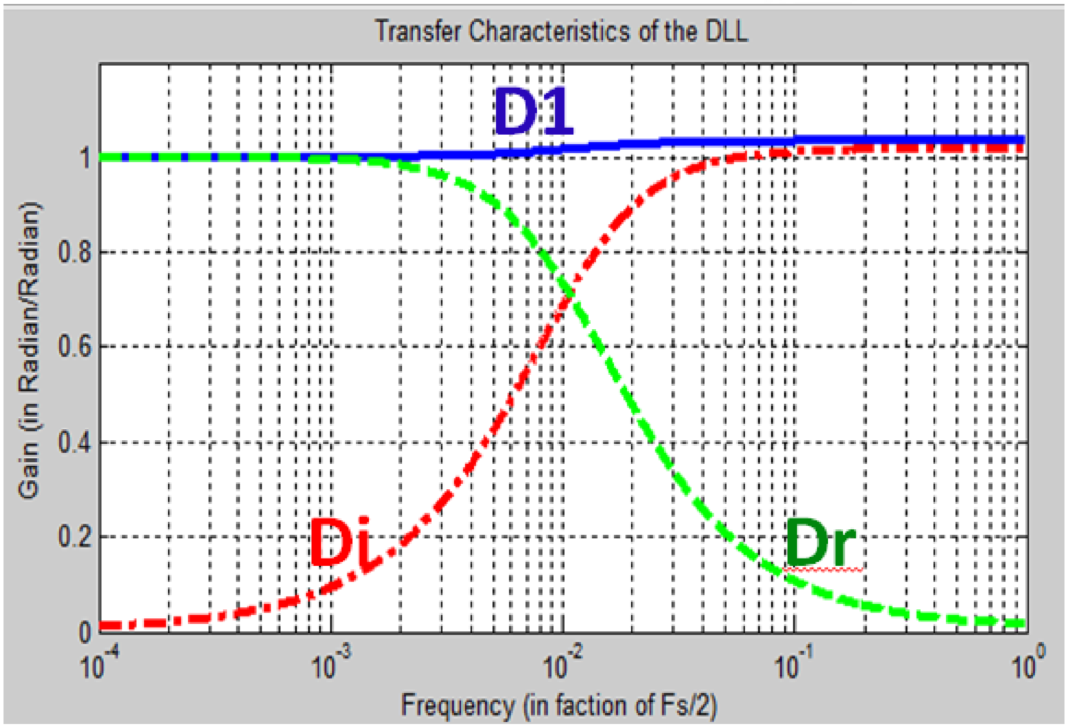

Figure 23: Jitter transfer characteristics of the delay locked loop

Figure 24 shows the locking behavior of the DLL. Count represents the counter output and Qerr represents the phase detector output. Initially, total delay (Do) provided by 16 delay cells is equal to 25.8 ns, which is much lower than the target delay (Dr) of 31.25 ns (= time period of a 32MHz Clock). As a result phase detector outputs a 'high' requesting the loop control to decrease the current controlling the delay lines. So, DLL control loop decreases the current controlling the delay cells by decreasing the counter output. At each clock cycle, the counter output is changed by 1 count.

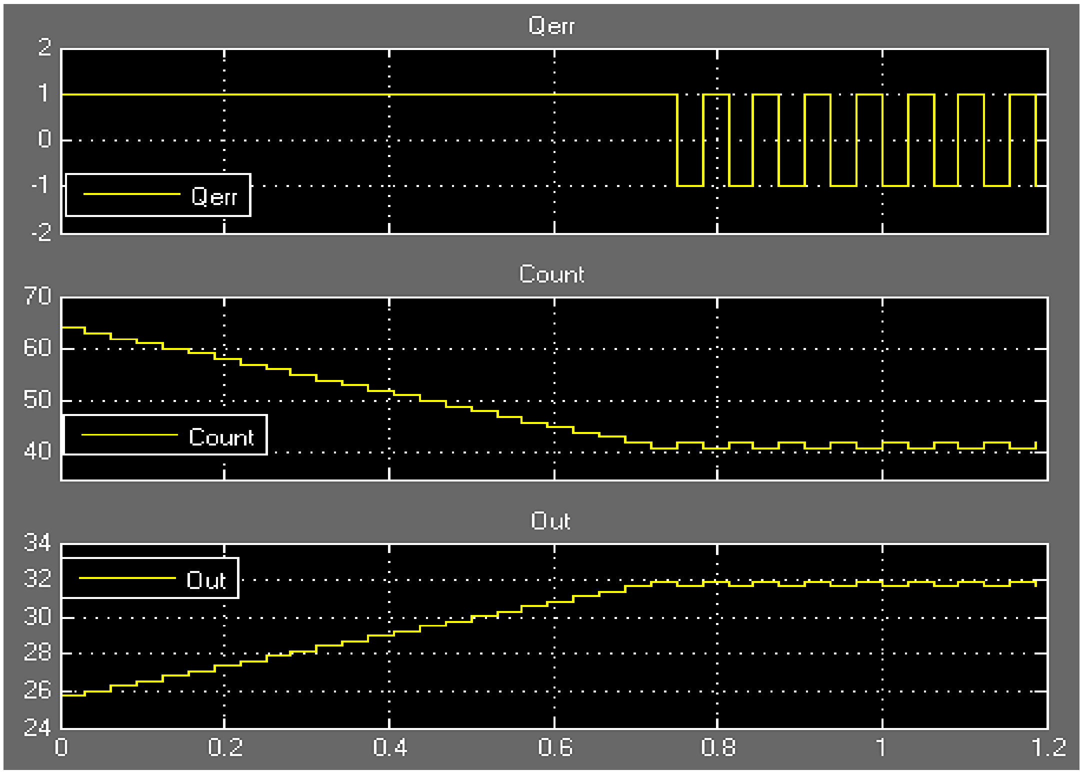

Figure 24: Locking behavior of the DLL in the fast corner

Since the control loop of the DLL is implemented digitally, the phase detector can only output a 'high' or a 'low'. It means that PD can only request to either increase or decrease the current controlling the delay of the delay cells. As a result, the output of the counter will go back and forth even when the DLL is in the locked condition. This introduces switching jitter in the DLL. Sources and analysis of the DLL jitter are discussed at length in the following section.

3.4 Jitter Analysis

As we discussed in the last section, due to the digital nature of the DLL control loop, delay provided by the delay cells go back and forth even when the DLL is in the locked state. The switching behavior of DLL in locked condition is shown in Figure 25. This variation in the delay introduces jitter in the DLL and is referred as 'Switching Jitter' of the DLL. The magnitude of the switching jitter depends on the delay resolution of the DLL. Since, IDAC output current, $I_{CONTROL}$, controls delay in the delay-cell; resolution of the $I_{CONTROL}$ determines the switching jitter in this DLL. The resolution of the IDAC is primarily determined by the gain of the IDAC and the number bits in the counter controlling the IDAC.

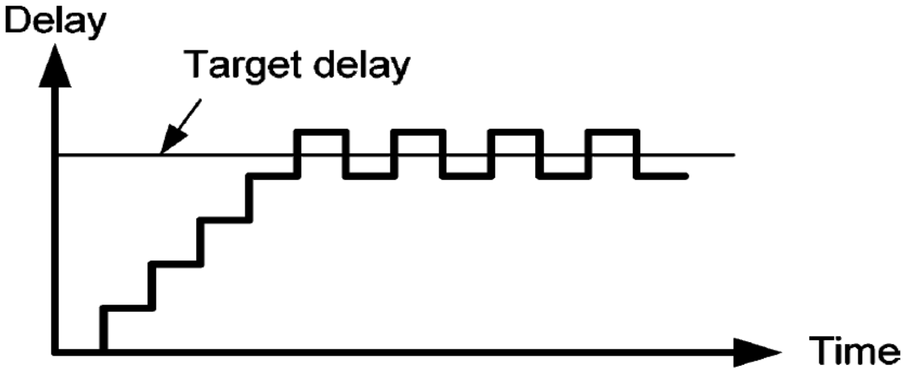

Figure 25: DLL output jitter due to digital loop with no reference clock jitter [16]

For ease of analysis let us assume that IDAC current, $I_{CONTROL}$ switches between N*I_{BAIS} and (N+1)*I_{BIAS}. Then the switching jitter of the DLL in the locked can be expressed as,

$$T_D = \frac{T_{REF}}{16} \times \frac{1}{N} \qquad \text{Equation 19}$$

In the computation so far it is assumed that the reference clock doesn't have any jitter. If the reference clock has jitter, the reference delay, T_{REF} will also vary with time. And if the point of transition of the switching of delay in DLL coincides with the band of the target delay (due to the jitter in the reference clock) then the DLL will switch between more than two delay levels as shown in the Figure 26. In the literature [16] more details on switching jitter is provided and also a circuit is proposed to limit the multi-level delay switching due to the reference clock which is beyond the scope of the material presented here.

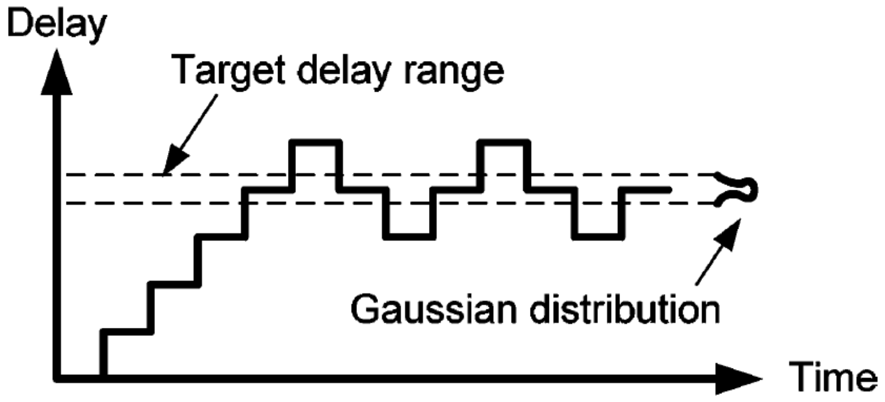

Figure 26: DLL output jitter due to digital loop with reference clock jitter [16]

In addition to the switching jitter and reference jitter, there are jitter sources ($T_{J,\ pk-pk}$) internal to the DLL. They also get added to the total output jitter in the DLL. Thermal noise of the MOSFET, supply & ground noise, substrate noise, and V_{TH} variation are to name a few. Figure 27 shows the total output jitter of the DLL.

$$T_{J,TOTAL} = 2 \times T_D + T_{J,pk-pk} \qquad \text{Equation 20}$$

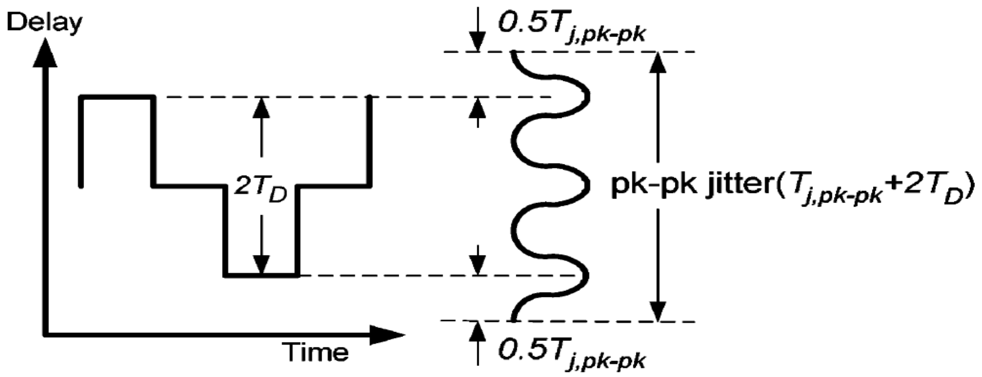

Figure 27: Total output jitter of the DLL considering all the sources [16]

4 IMPLEMENTATION AND RESULTS

4.1 Digital to Pulse Width Modulator (DPWM)

The block level diagram of the designed DPWM is shown in Figure 28. It consists of 2 major sections:

1. A delay Locked Loop (DLL)
2. Supporting circuits consisting of counter, comparator and pulse generator.

Since, robustness and accuracy of the design is mainly determined by the DLL, implementation of the DPWM was started with designing the DLL.

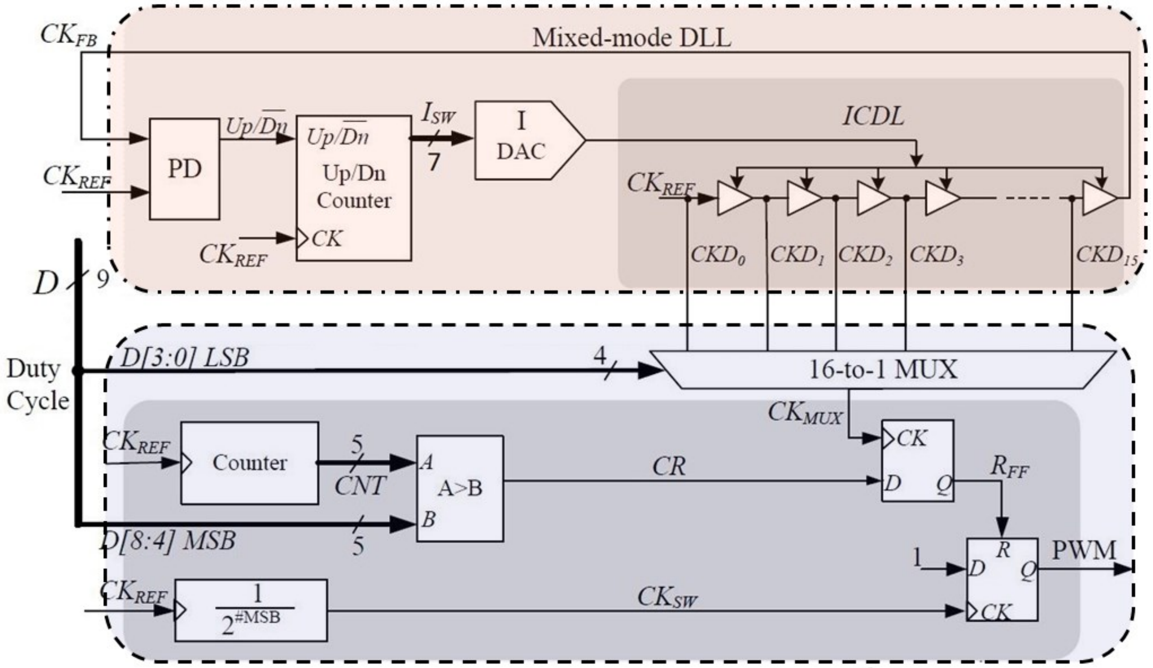

Figure 28: Block level diagram of the designed DPWM

4.2 Delay Locked Loop

Figure 29 shows the schematic of the DLL designed for the DPWM.

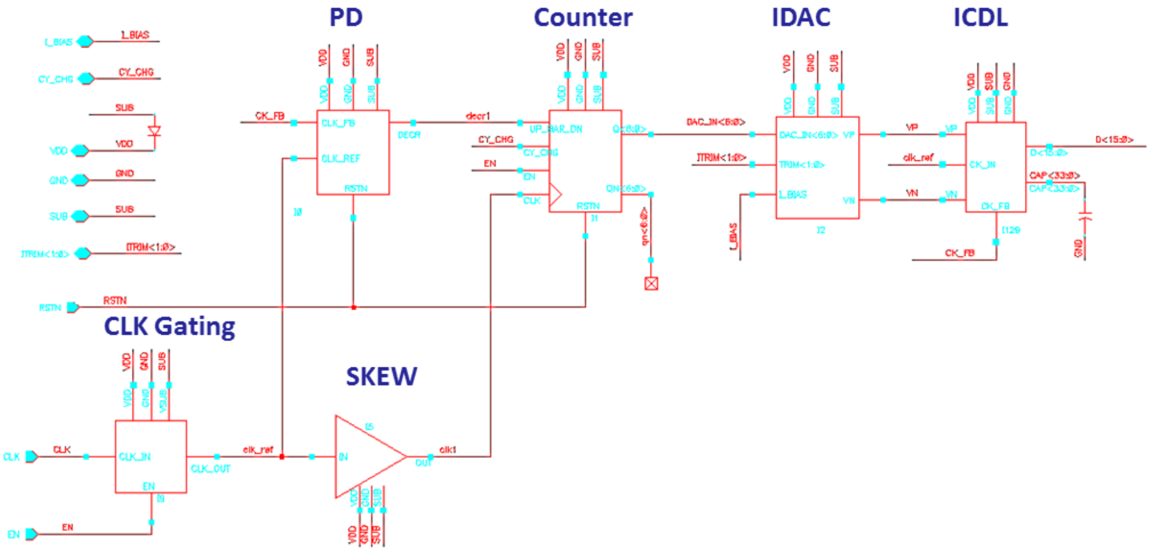

Figure 29: Schematic of the delay locked loop

The DLL consists of 6 sub-blocks:

1. Phase detector
2. Up/Down Counter
3. Current DAC (IDAC)
4. Current controlled delay arrays (ICDL)
5. Clock Gating, and
6. Skew block

The details of each block of the DLL are explained in the sections below.

4.2.1 Phase Detector

The design uses a simple D–Flip flop to implement a digital phase detector. Figure 30 shows the schematic of the phase detector used in the DLL. To compare the time period of the reference clock (CK_{REF}) with the delay generated by 16 delay cells, the output of the delay cells (CK_{FB}) is sampled with CK_{REF} using a D flip-flop.

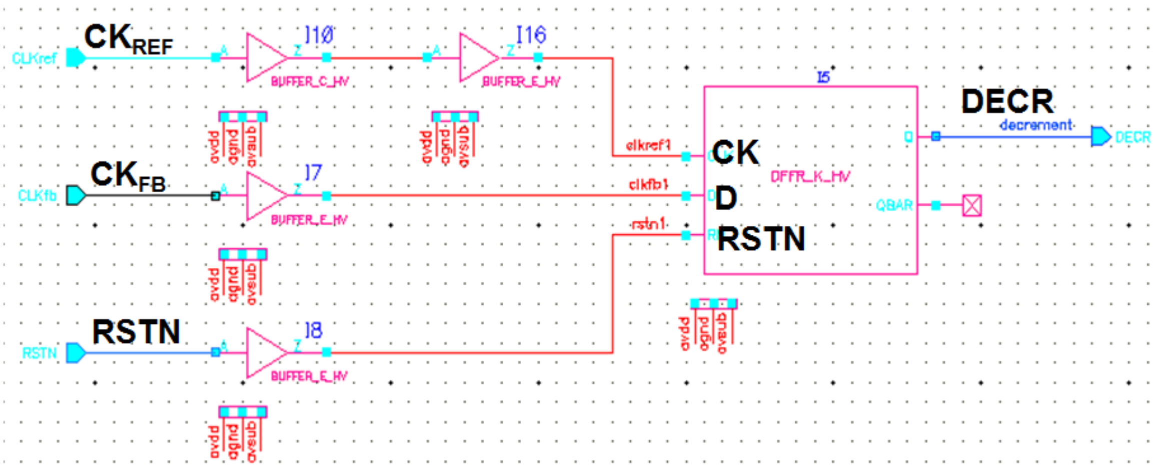

Figure 30: Schematic of the Simplified phase detector used in the design

The operation of the phase detector is explained in Figure 31. When delay provided by the delay cells is larger than targeted delay (31.25ns) D flip-flop (DFF) in PD samples a 'low' and request to increase $I_{CONTROL.}$ Since, delay in the delay array is inversely proportional to $I_{CONTROL}$ (Equation 14) with the increase in $I_{CONTROL}$ the delay of the delay-cells reduces and moves towards the target delay.

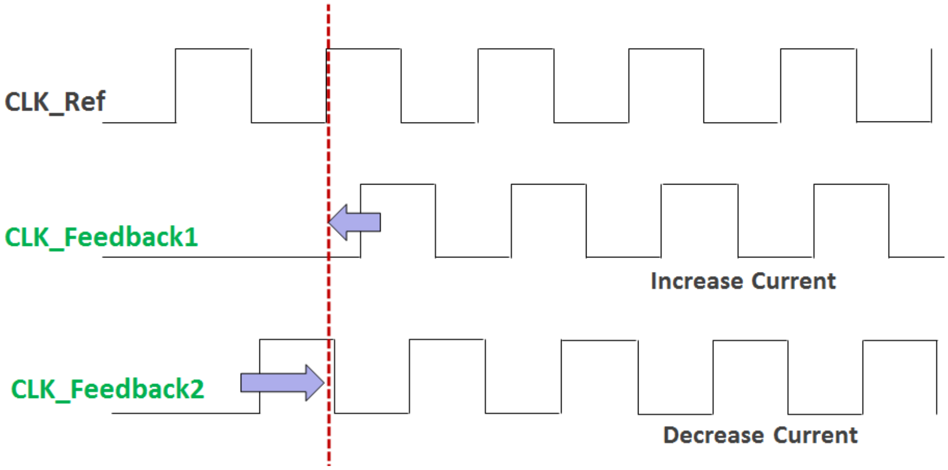

Figure 31: The functionality of the Simplified Phase detector

4.2.1.1 Verifying PD robustness against the error-condition

Figure 32 discusses a scenario when the logic for the DLL control could possibly fail. This arises when the initial delay provided by the delay array (T_{DELAY}) is smaller than $T_{REF}/2$. When $T_{DELAY} < T_{REF}/2$, PD outputs a 'low' and the control loop understands that the delay provided by the delay cells are more than the reference delay, which is not true. In fact, the delay provided by the delay cell here is much smaller than the reference delay ($T_{DELAY} < T_{REF}/2$). So, the design should ensure that across PVT corners, the delay provided by the delay cells never reach below $T_{REF}/2$. This issue is discussed further in the section 4.2.3.

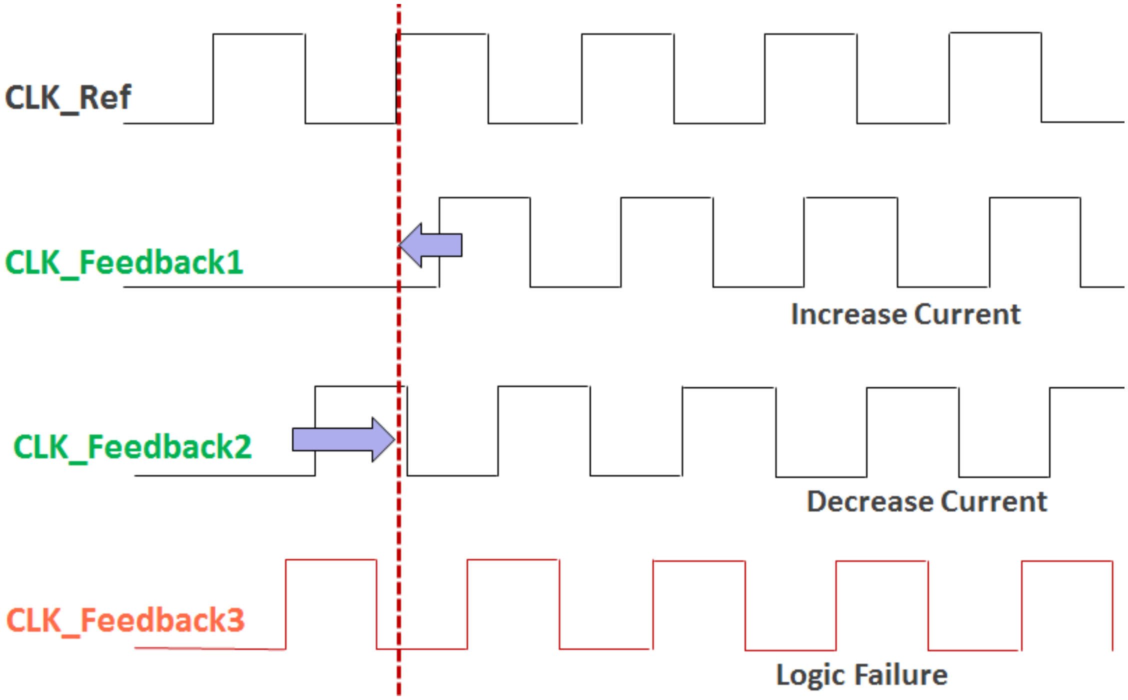

Figure 32: The limits of the Simplified Phase Detector

4.2.2 Counter

Counter in the DLL is used to track the context of the DLL. To support 7 bit IDAC the design uses a 7 bit synchronous counter. Each time the phase detector detects a 'low' (or 'high') the output of the counter is incremented (or decremented) as a result IDAC changes the output current required to change the delay in the delay locked loop. During the PFM mode, the counter output is preserved to hold the state of the DLL in the locked condition and minimize the locking time when converter transitions back to PWM mode.

4.2.2.1 Roll-over Protection

The design includes detection of an under-run/over-run condition. An over-run condition can be defined as a state when the counter output in DLL has reached to its limit and loop wants to move it further, e.g. counter has reached 127 and the PD requests to increase it further. This can result into roll-over and counter output can then become 0 rather than 128 (as requested). The protection circuit in the counter ignores the request to increase output after reaching to 127 and vice-versa to avoid count roll-over. This is a fail-safe feature which takes care of extreme variation of PVT on delay in the DLL.

So, the protection circuit holds the last state of the counter output (used to control IDAC/Delay) when:

1. The present D-Word is 127 and Phase Detector (PD) requests for an increment, or
2. The present D-Word is 0 and Phase Detector (PD) requests for a decrement.

4.2.3 Current - Digital to Analog Converter (IDAC)

The delay locked loop (DLL) uses a current DAC as a current summer to generate control, $I_{CONTROL}$, for the current controlled delay lines. Figure 33 shows schematic of the IDAC used in the DLL. For example, say the delay provided by the delay array is T_{DELAY} which is less than target delay T_{CLK}. So, current in the delay cell is more than required and it has to be reduced which is achieved by reducing the counter output. Additional trim bits ITRIM [1:0] ensure robustness of the design across extreme PVT corners. The resolution of the IDAC determines the switching jitter in the DLL. Higher is the resolution of IDAC, we can finely tune the delay by tuning the control current, and so smaller will be switching jitter. The IDAC uses 7 bits to control the current. This resolution is sufficient to limit the switching noise below 30ps in the typical corner.

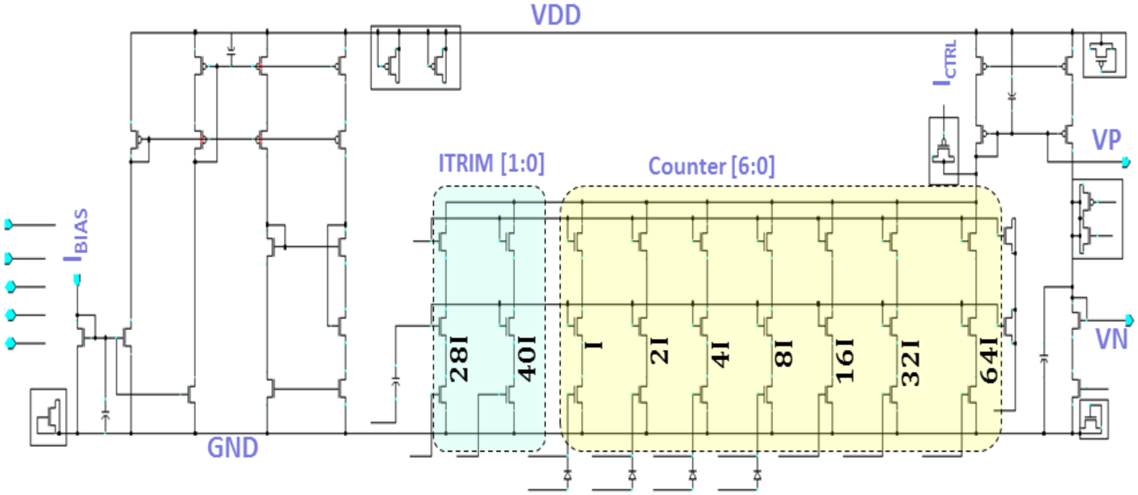

Figure 33: The schematic of the Current DAC (IDAC) used in the design

4.2.3.1 Linearity of the IDAC

Figure 34 shows input code vs. total output current plot of the IDAC across PVT corners. It shows that the IDAC is linear.

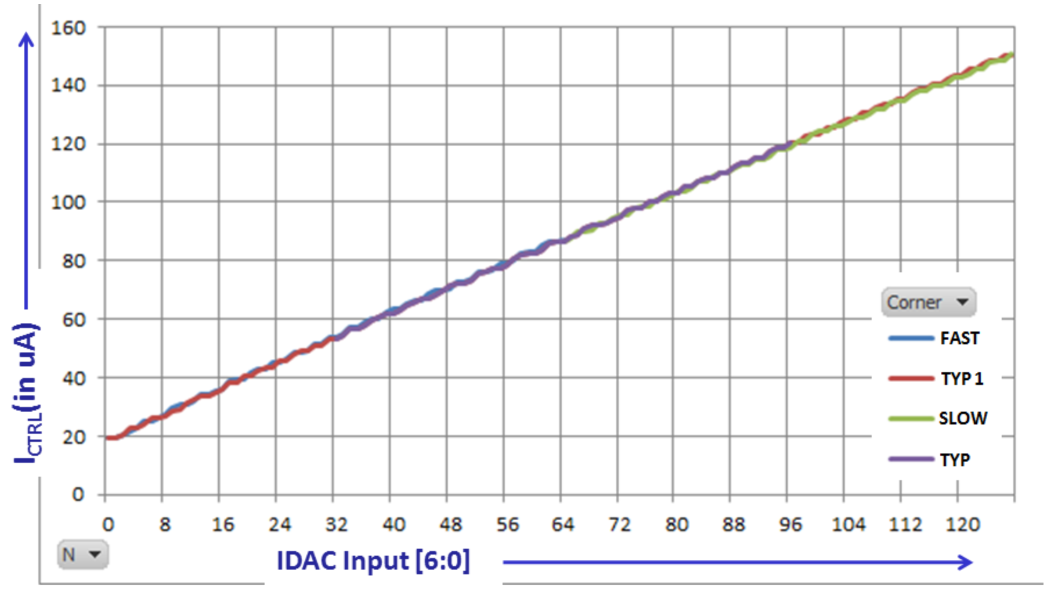

Figure 34: Input vs I_{CTRL} of the IDAC across PVT

4.2.3.2 Monotonicity of the control voltage

The control voltage output (VP, VN) of the IDAC, change monotonously (~linearly) with the IDAC input as shown in the Figure 35. VP and VN are used copy the IDAC current $I_{CONTROL}$ in the delay cells.

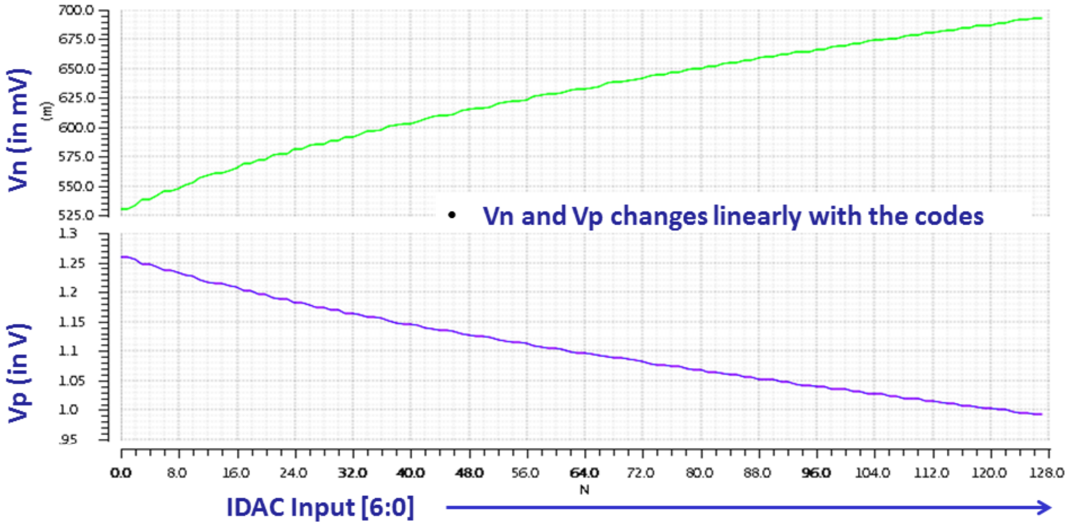

Figure 35: Variation of the control voltage output of the IDAC with the Input

4.2.4 Current Controlled Delay Line (ICDL)

A current starved buffer is used to generate current controlled delay line as shown in the Figure 36.

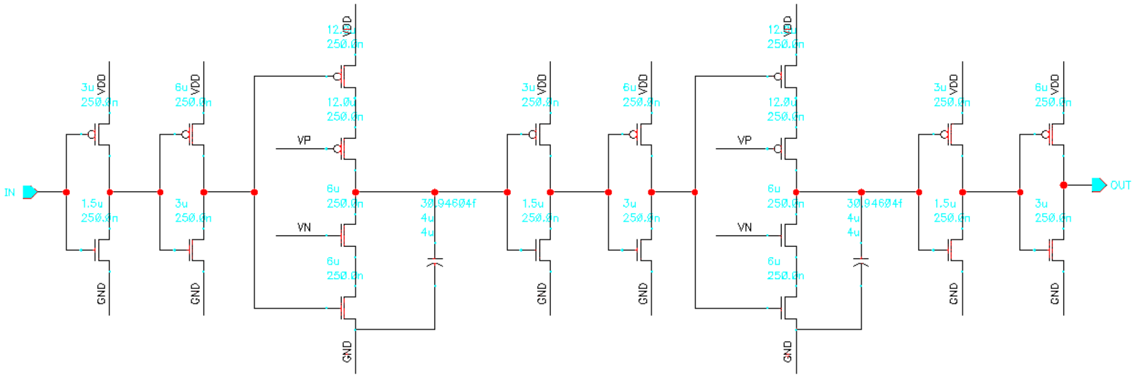

Figure 36: The schematic of the delay cell in the DLL

Pseudo-symmetric structure is selected for the delay lines. Pseudo-symmetric structure ensures that for each transition (rising or falling edge) delay line uses both

PMOS and NMOS of the current starved inverter to complete the transition at the output. The symmetric operation of delay cells is explained in Figure 37. For a rising edge (shown in yellow arrow), the delay cells first use NMOS followed by PMOS to complete the transition at the output. This ensures much lesser duty cycle variations due to mismatch of the NMOS and PMOS.

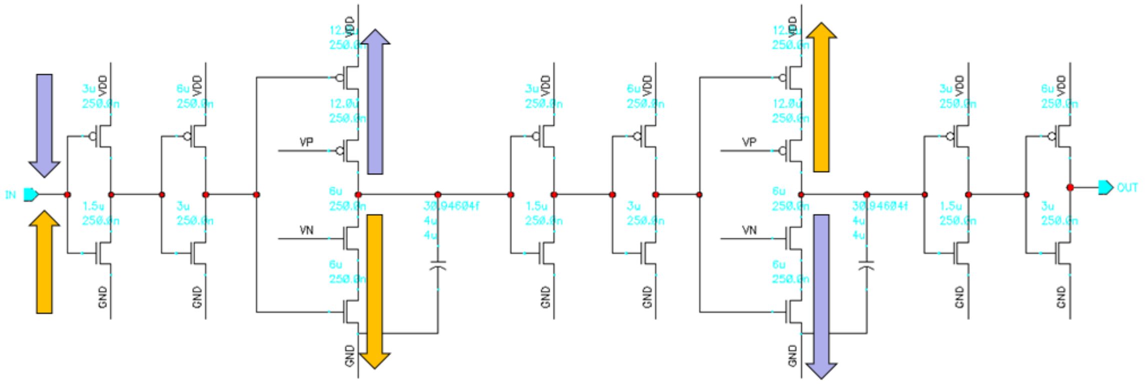

Figure 37: Pseudo-symmetric approach to reduce pulse width distortion

Table 3 compares the variation in the delay for rising and falling edge with and without (in bracket) pseudo-symmetric structures. Here,

t_R = VCDL stage delay when measured with the rising edge

t_F = VCDL stage delay when measured with the falling edge

Skew = Delay difference for rising and falling edge (~ duty cycle distortion)

Parameter	**Min**	**Avg.**	**Max**	**Unit**
Delay t_R	1.47 (1.61)	1.91 (2.0)	2.41 (2.46)	Ns
Delay t_F	1.48 (1.56)	1.92 (1.98)	2.42 (2.48)	Ns
Skew ($t_R - t_F$)	-8.0 (-145.1)	-11.7 (28.2)	-17.8 (160.2)	ps

Table 3: Improvement in duty-cycle distortion with pseudo-symmetric delay cell

Clearly, delay for rising edge and falling tracks each other closely in the pseudo-symmetric structure. So, we see a much smaller skew variation (min = -

18ps, max = -8ps) compared to the regular delay structure (min = -145ps, max = 160ps), leading to a much smaller duty-cycle-distortion.

4.2.4.1 DLL locking across PVT

Figure 38 explains the DLL locking operations across PVT corners. DLL starts with the default current in the IDAC corresponding to '64' at the counter output. For slow corner the delay corresponding to the default current is 41ns. And as the delay is inversely proportional to the controlling current the control loop starts increasing the current until the delay cells achieve targeted delay (=31.25 ns = T_{CLK}).

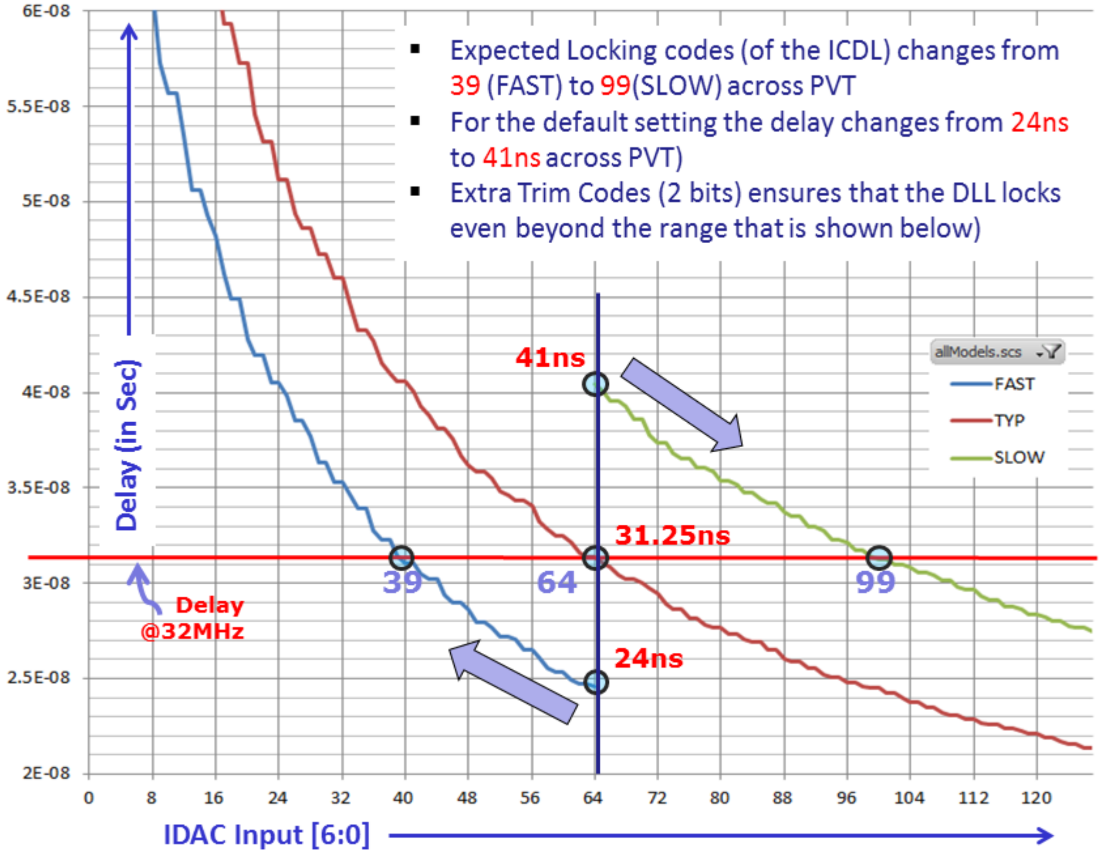

Figure 38: The variation of the delay with the counter output across PVT

Please note that in the worst case fast corner the delay is 24ns which is much higher than 15.6ns (= $T_{CLK}/2$). It ensures that the limit condition for the phase detector will not occur.

4.3 DPWM

4.3.1 Linearity

The linearity of the DPWM is observed by plotting DPWM output pulse-width with its input. Figure 39 presents the variation of the output pulse width with the input DWORD. The plot is linear. To accurately measure the linearity of the DPWM, one need to plot the deviation of the pulse width of the DPWM output from the ideal value for the range of input DWORD. In order to keep the design N – bit accurate, this deviation should be limited to half of the least significant bit (LSB). Equation 9 gives the expression for the LSB (expressed as ΔV_{DPWM}). In Figure 40, the deviation of the pulse-width of the DPWM output is plotted for the designed DPWM.

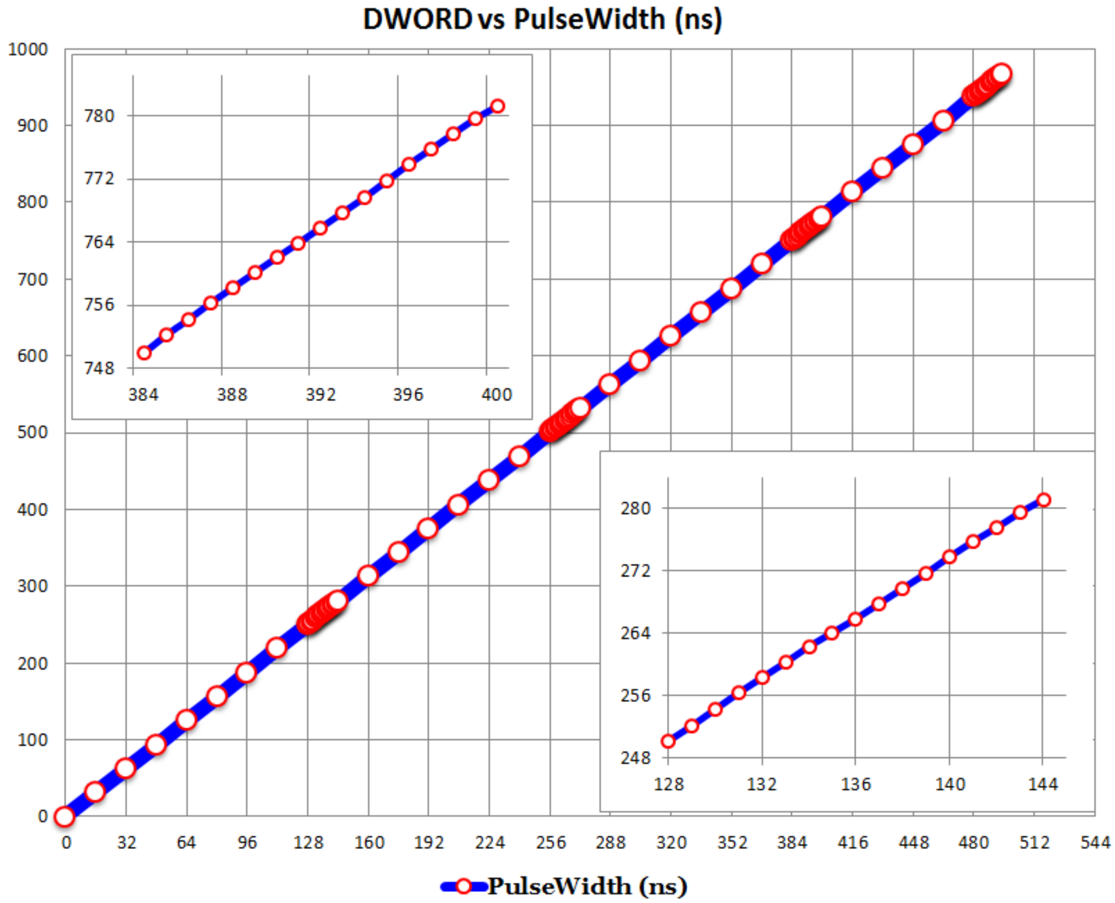

Figure 39: Variation of pulse-width of the DPWM with the input

As we can see the deviation is smaller than half the LSB (< 0.4 x LSB) which proves that the DPWM is 9 – bit accurate. The design is checked across PVT and found the deviation to be worst in slow corner (slow process, low supply voltage and high temperature).

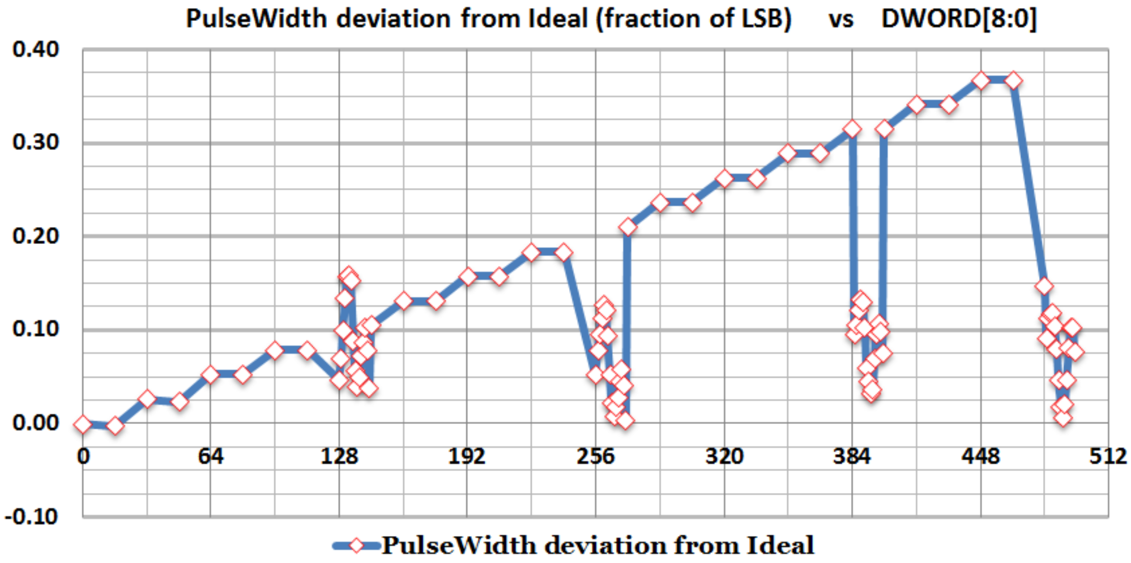

Figure 40: The Deviation of the DPWM output from ideal with the input DWORD

The deviation of the pulse width of the DPWM is caused by 3 main factors:

1. Delay of the 16-input Multiplexer following the DLL.
2. Clock to output delay of the reset generating D – flip-flop.
3. Time taken by the pulse width generating flip – flip to reset its output once it receives a reset signal.

We can see that the pulse width deviation ($T_{ON,\ ERR}$) varies between 0 to 0.4 times LSB (ΔT_{DPWM}). We can limit the $|T_{ON,\ ERR}|$ to 0.2 x ΔT_{DPWM} by delaying the input clock of the D flip-flop responsible for generating the pulse width by 0.2 x ΔT_{DPWM}.

4.3.2 DPWM **Power**

Table 4 lists power dissipation of the entire DPWM across PVT corners. The data is generated with the extracted layout simulation. The design consumes average power of 2.35mW in the worst case. Please note that, DPWM is responsible to provide 32MHz and 1MHz reference clocks to rest of the blocks in the designed buck converter. As a result, total power requirement of the DPMW include:

1. Power requirement to implement DPWM function, and
2. Power required by clock divider and buffers.

Since it is very difficult to separate the power lines in the layout, the power utilized by the DPWM function (P_{DPWM}) and clock dividers with buffers (P_{BUFFER}) observed during the schematic simulation in the typical corner. And the fractional utilization of the total power for the DPWM function is calculated.

Fractional utilization,

$$U_{DPWM} = \frac{P_{DPWM}}{P_{DPWM} + P_{BUFFER}} \qquad \text{Equation 21}$$

It was assumed that fractional utilization of the power by DPWM (i.e. U_{DPWM}) does not change between schematic and post-layout simulation while creating Table 4. Please note that the power dissipation of the DPWM design reduces by ~60% during PFM mode of converter.

Power (in mW)	**Average Power**		**RMS Power**	
	PWM Mode	**PFM Mode**	**PWM Mode**	**PFM Mode**
TYP	1.67	0.59	1.96	0.75
FAST	2.35	0.83	2.89	1.1
SLOW	1.65	0.63	1.89	0.67

Table 4: Power dissipation of DPWM in PWM and PFM modes

4.4 Layout

Figure 41 shows the complete layout of the DPWM. The layout was done using the IBM 0.18µm CMHV7SF process. It consumes an area of 0.209 mm^2 (618 µm x 338 µm).

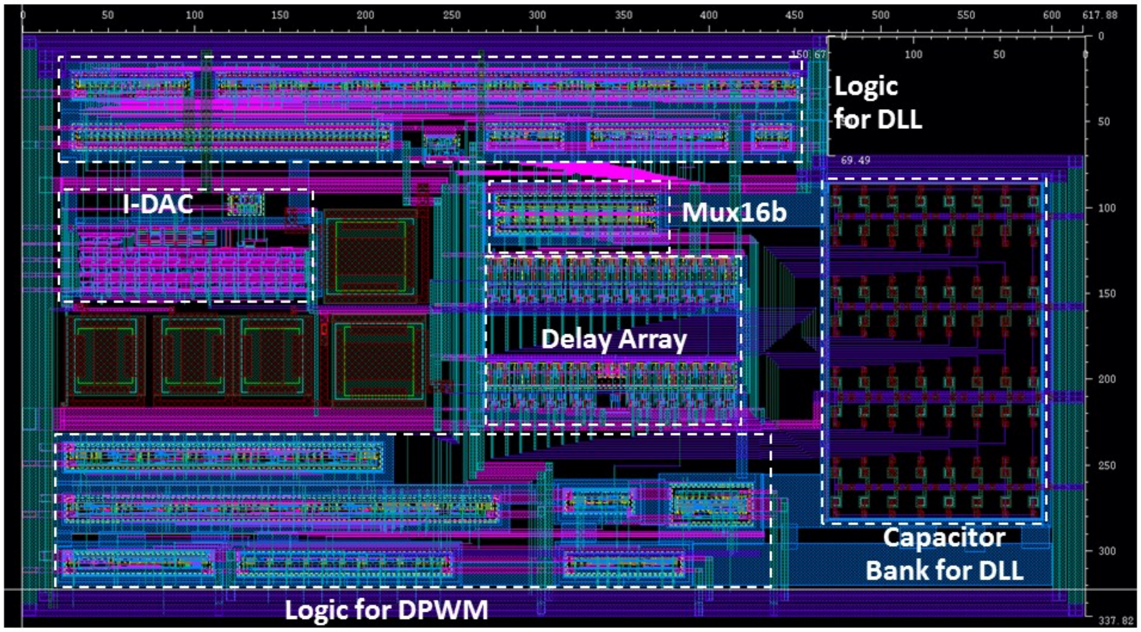

Figure 41: Layout of the DPWM

The major portion of the area consists of digital logic used in the design. Since the design is doesn't have strict matching requirements, every effort is taken to reduce the area of the design while keep the design time low by the use of the standard cells.

5 CONCLUSION

The design of a 9 bit hybrid DPWM for a switched mode buck converter is developed. The worst case average power dissipation of the DPWM is 2.35 mW. For power dissipation in the FAST corner (Fast MOSFETs and passive devices, 1.98V, 125ºC) is the worst case corner. The layout of the DPWM takes an area of 618 µm x 338 µm. The major sections of the design are implemented using digital logic to keep the design robust and the extensive use of the standard cells ensured faster implementation and ease portability to a lower technology node. Delay locked loop is optimized by adding several fail-safe features and making the design more robust to PVT variations. The jitter performance of the designed DLL is analyzed and explained. Pseudo symmetric approach is taken to ensure minimal duty cycle distortion by the DLL. Clock gating of the DLL is enabled to lower down the switching losses of the DPWM in the low load PFM operation of the converter and reduced the power dissipation by ~60%.

The extracted simulation of the DPWM showed the design to be 9 bit linear with the DNL to be less than 0.4*LSB. The DNL can be further improved (to +/- 0.2 LSB) by providing a skewed (by 0.2*LSB) clock to the pulse generator block.

REFERENCES

[1] Tao Liu ; Hyunsoo Yeom ; Vermeire, B. ; Adell, P. ; Bakkaloglu, B. "A digitally controlled DC-DC buck converter with lossless load-current sensing and BIST functionality," *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, pp. 388-390, 2011.

[2] Yu-Cherng Hung, "Delay-Line Sharing Based: A New CMOS Digital PWM Circuit", in *Computer, Consumer and Control (IS3C), 2012 International Symposium on*, pp. 305-308, 2012.

[3] Prodic, A. ; Maksimovic, D. ; Erickson, R.W., "Design and implementation of a digital PWM controller for a high-frequency switching DC-DC power converter", *Industrial Electronics Society, 2001. IECON '01. The 27th Annual Conference of the IEEE*, vol. 2, pp. 893-898, 2001.

[4] Jingquan Chen ; Ribeiro, M. ; Payseo, R. ; Dongsheng Zhou ; Smith, J.R. ; Kernahan, K., "DPWM time resolution requirements for digitally controlled DC-DC converters", Applied Power Electronics Conference and Exposition, 2006. APEC '06. Twenty-First Annual IEEE, 2006.

[5] Pui-Kei Leong ; Chun-Hung Yang ; Chi-Wai Leng ; Chien-Hung Tsai, "Design and implementation of sigma-delta DPWM controller for switching converter", Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on, pp. 3074-3077, 2009.

[6] Tao Liu, "Digitally Controlled Buck Converter with Lossless Current Sensing," *PhD Dissertation, Arizona State University*, Dec. 2011.

[7] Peterchev, A.V.; Sanders, S.R., "Quantization resolution and limit cycling in digitally controlled PWM converters", Power Electronics Specialists Conference, 2001. PESC. 2001 IEEE 32nd Annual, vol. 2, pp. 465-471, 2001.

[8] John Tucker, "Understanding output voltage limitations of DC/DC buck converters", Analog Application Journal, Texas Instruments, pp. 11-14, 2008.

[9] Syed, A. ; Ahmed, E. ; Maksimovic, D. ; Alarcon, E., "Digital pulse width modulator architectures", Power Electronics Specialists Conference, 2004. PESC 04. 2004 IEEE 35th Annual, vol. 6, pp. 4689-4695, 2004.

[10] Wikipedia, "DC-to-DC Converter", http://en.wikipedia.org/wiki/DC-to-DC_converter.

[11] Yanxia Gao ; Shaofeng Zhang ; Yanping Xu ; Shuibao Gao, "Analysis and comparison of three implementation methodologies for high-resolution DPWM", *Power Electronics Systems and Applications, 2009. PESA 2009. 3rd International Conference on*, pp. 1-7, 2009.

[12] Garlepp, B.W. ; Donnelly, K.S. ; Jun Kim ; Chau, P.S. ; Zerbe, J.L. ; Huang, C.; Tran, C.V. ; Portmann, L. ; Stark, D. ; Yiu-Fai Chan ; Lee, T.H. ; Horowitz, M.A., "A portable digital DLL for high-speed CMOS interface circuits", Solid-State Circuits, IEEE Journal of, Vol. 35, Issue. 5, pp. 632-644, 1999.

[13] Begur, Soumya Shivakumar, "A 45nm CMOS, low jitter, all-digital delayed locked loop with a circuit to dynamically vary phase to achieve fast lock", *MS Dissertation, Computer Science, Northeastern University*, 2011.

[14] Soliman, S.; Yuan, F.; Raahemifar, K., "An overview of design techniques for CMOS phase detectors", *Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on*, Vol. 5, pp. 457-460, 2002.

[15] C. Jia, "A delay-locked loop for multiple clock phases/delays generation", *PhD Dissertation, School of Electrical and Computer Engineering, Georgia Institute of Technology*, 2005.

[16] Rong-Jyi Yang ; Shen-Iuan Liu, "A 2.5 GHz All-Digital Delay-Locked Loop in 0.13 μm CMOS Technology", *Solid-State Circuits, IEEE Journal of*, Vol. 42, Issue 11, pp. 2338-2347, 2007.