A Programmable Digital Pulse Width Modulator Providing Versatile Pulse Patterns and Supporting Switching Frequencies Beyond 15 MHz

Eamon O’ Malley  
Dept. of Electronics & Computer Engineering (PEI)  
University of Limerick  
Limerick, Ireland  
eamonn.omalley@ul.ie

Dr. Karl Rinne  
Dept. of Electronics & Computer Engineering (PEI)  
University of Limerick  
Limerick, Ireland  
karl.rinne@ul.ie

Abstract—This paper describes the implementation of a novel and highly versatile digital pulse width modulator (DPWM) – probably the most critical component in any digitally controlled switching power converter. Its new architecture, composed of a delay-locked loop (DLL) and programmable DPWM module, allows the generation of high resolution, high switching frequency ($f_{sw}$) PWM signals. The DPWM IC was designed using a standard 0.35-µm CMOS process and supports switching frequencies beyond 15 MHz.

Keywords—digital control; digital pulse width modulation

I. INTRODUCTION

Digitally controlled switching power converters have received increasing research attention over the past number of years [1-6]. They have numerous advantages over converters controlled by analogue circuitry that can be grouped into three categories: performance (e.g. programmability, advanced control schemes), logistics/manufacturing (e.g. reduced component count, advanced calibration schemes) and design methodology (e.g. design re-usability, process independence, and improved IP protection).

Digital control translates into circuit blocks characterised by high data throughput and low latency. Various solutions to deal with and ease the processing speed requirements have been reported [3, 4], suitable analogue-to-digital converter (ADC) architectures (differential, delay-line) have been discussed [2, 4], controller implementations have been compared, and DPWM architectures have been recommended [1, 2, 4, 5, 7, 8].

A digitally controlled (shaded-area) power converter is shown in Figure 1. Power is taken from a source, processed by a switching converter, and delivered to a load.

A DPWM accepts a duty cycle command $d_c$, and translates it into one or more pulse-width-modulated driving waveforms $PWM_i(t)$ to control the active switches of the power converter.

II. DPWM CONSIDERATIONS FOR DIGITALLY CONTROLLED SMPCS

While previous work [1, 2, 4, 5, 7, 8] successfully designed DPWM modules for SMPC digital controller ICs, some of the crucial practical DPWM issues remain unresolved such as...
programmability of switching parameters (e.g. resolution, duty-cycle), versatility in the PWM output waveforms (e.g. dead-times) and switching frequencies beyond 1 MHz.

Digitally controlled switching converters (Figure 1) typically require the DPWM to interface to both a CPU (for programming parameters on start-up and status monitoring during normal operation) and a feedback controller (for cycle-by-cycle duty-cycle updates) as shown in Figure 2.

![DPWM I/O](image)

Figure 2 DPWM Input/Output requirements

Practical SMPC topologies typically require more than one PWM driving waveform with special phase relationships determined by dead/delay times as shown in Figure 3. Programmable dead/delay times (via the CPU) on initialization would eliminate the need for external phase shift/invert circuitry.

![PWM Pulse Patterns for various SMPC topologies](image)

Figure 3 PWM Pulse Patterns for various SMPC topologies

As discussed earlier, samples of the regulated signal and possibly extra signals must be taken periodically (sampling), and converted into digital signals (quantisation). Within a switching cycle the preferred sampling time point is determined mainly by three (partially conflicting) requirements:

- Samples should be taken as early as possible to ease ADC and discrete controller latency requirements
- Samples should be taken when synchronised noise in the power circuit is low
- Samples should be taken at the time point when the instantaneous value of the regulated signal (approximately) reflects the average value

Requirement 1 is self-explanatory. An updated duty cycle command \( d_k \) must be provided to the DPWM at the start of a new switching cycle. Requirement 2 indicates that sampling of the input signal(s) should not be performed during power stage commutation periods when high \( dv/dt \)'s or \( di/dt \)'s are to be expected. Sampling must be synchronised to power stage switching but should not coincide with opening or closing of power switches. Figure 4 (top) illustrates the challenge for the case of a typical synchronous buck converter (\( V_{in}=12 \) V, \( V_o=2.5 \) V, \( I_o=15 \) A). The instantaneous output voltage \( V_o \) exhibits severe synchronised noise spikes whenever the power mosfets are switched.

![Synchronised output voltage noise of a synchronous buck converter](image)

Figure 4 Synchronised output voltage noise of a synchronous buck converter (top: output voltage \( V_o \), bottom: sampling pulse for ADC)

Requirement 3 is based on the fact that in the presence of significant ripple it is desirable to control the average value of the regulated signal (rather than it’s peak or valley value). The time point when the output voltage equals its average value depends on parasitic (and therefore ill-defined) components in the power stage (e.g. the ESR of the output filter capacitor). As a first-order approximation requirement 3 can be approximately satisfied if samples are taken halfway through the on-time of the control FET (or, alternatively, half-way through the on-time of the complementary or freewheeling FET).

The programmable DPWM presented here as shown in Figure 5 resolves these practical considerations by providing four PWM signals (for switches in both the primary and secondary side) supporting various output modes (complementary or alternating) with programmable dead/delay times (to avoid external phase shift/invert circuitry) along with an ADC S&H synchronisation pulse delivered at a constant \( d_k/2 \cdot T_{sw} \) (where \( T_{sw} = 1/f_{sw} \)) as shown in Figure 3 and Figure 4 (bottom).
III. DIGITAL PULSE WIDTH MODULATOR ARCHITECTURE

Figure 5 shows a block diagram of the digital pulse width modulator (DPWM) IC. It is primarily composed of a programmable DPWM module and an analogue delay-locked loop (DLL). While previous DPWMs [1, 2, 4, 5, 7-9] have achieved high resolution switching frequencies ($f_{sw}$) at moderate clock speeds (e.g. 32 MHz) their architectures (usually containing asynchronous ripple counters, fixed length delay lines and large multiplexers) are inflexible for general purpose applications. The DPWM in Figure 5 provides PWM switching frequency waveforms in the range of 100 kHz to 15 MHz with resolutions of 6 to 12 bits by programming the required parameters via a CPU interface and adjusting the system clock (Clk in Figure 5) frequency $f_{clk}$ as required.

The analogue delay-locked loop architecture is based on [10] and contains a control circuit (composed of a phase detector, charge pump, first order loop filter and bias voltage buffer) along with a voltage-controlled delay line (VCDL) consisting of $M$ cascaded variable delay stages. For improved jitter performance and reduced on-chip area $M$ was chosen to be 8 for this design. The reference clock $clk$ passes through each delay stage. The final delay stage output $clk_d$, is compared by the phase detector to the input $clk$ to generate a phase alignment error signal. The phase error is integrated by the charge pump and loop filter to generate $vcp$. The bias voltage buffer then generates a replica of $vcp$ along with a bias voltage $vbias$. These voltages adjust the delay of each stage in the VCDL.

When the DLL locks the reference input clock to the delayed clock the total delay of the delay line should be equal to exactly one period of the reference clock resulting in 8 evenly delayed phases of the $clk$ at each buffer stage in the VCDL as shown in Figure 6. The DLL has a lock range of 35 MHz to 145 MHz.

The programmable DPWM module is a digital system operating synchronously to the DLL output clock $clk_d$. It supports a CPU interface allowing parameters such as duty-cycle, mode (complementary or alternating), dead-times, and number of bits resolution $N$ (6 to 12 bits) to be programmed on power up. The DSP interface allows updated duty-cycle values from a feedback controller to be programmed on a switching cycle-by-cycle basis during normal operation.

The DPWM module in tandem with the DLL generates the PWM waveforms based on the programmed parameters at clock frequencies given by $f_{clk} = \left(f_{sw} \times 2^N\right)/8$ (where $N$ is the number of bits resolution e.g. $f_{sw} = 4$ MHz @ 8-bit resolution requires $f_{clk} = 128$ MHz).

An asynchronous input $Stopn$ supports power supply protection features (such as high-speed over-current-protection). To protect the switches in the power stages activation of $Stopn$ resets all PWM outputs to zero within 1 ns.

Figure 6 DLL clock phases when locked to 1 period of the reference clock $clk$
IV. DIGITAL PULSE WIDTH MODULATOR DESIGN

A) Programmable DPWM Module

Figure 7 shows the architecture for the programmable DPWM module. The I/O Control block contains all the programmable registers (e.g. duty-cycle, dead/delay times, number of bits resolution) and controls all communication via the CPU or DSP to/from the DPWM.

The DPWM Controllers generate the individual PWM waveforms. Each DPWM Controller is primarily composed of a 2-state finite state machine (FSM). PWM3 and PWM1 duty-cycles are scaled versions of the PWM0 duty-cycle based on their programmed dead-times respectively (Figure 3). The controllers also calculate the required low time for each PWM output by calculating the 2’s complement of the high time (Figure 9). The ADC S&H logic uses the duty-cycle for PWM0 to generate the ADC_S&H_EN at \( \frac{d_k}{2} \cdot T_{sw} \) as discussed in section II.

![Figure 7 Block diagram of the Programmable DPWM Module](image)

In a digital counter/comparator type DPWM [7-9] the operating clock frequency is given by \( f_{clk} = (f_{sw} \cdot 2^N) \). To achieve high resolution high switching frequencies (e.g. \( f_{sw} = 1 \) MHz @ 10-bit resolution) requires \( f_{clk} = 1 \) GHz. As mentioned in Section III the DPWM module working in tandem with the DLL reduces the system clock frequency by a factor of 8 as follows.

During the high/low time generation of the PWM waveform the count sequence for the 3 least significant bits (lsb’s) \( b_2-b_0 \) repeats every 8 cycles within the N-bit counter. This is illustrated in Figure 8 where rotating anti-clockwise around the circle binary one is added to the counter at each segment on a rising edge of \( f_{clk} \). This repetition can be avoided if instead binary 8 (b’1000) is added to bits \( b_2-b_0 \) of the N-bit counter (only add on each revolution of the circle not at each segment) hence reducing the \( f_{clk} \) by a factor of 8. Once \( b_2-b_1 \) of the N-bit counter equal \( b_2-b_1 \) of the programmed \( d_k \), \( b_2-b_0 \) of \( d_k \) select the delayed phase of \( f_{clk} \) from the DLL to reset the PWM output (CLK_SEL signals in Figure 7). The FSM data flow for generating the high/low times for each PWM signal based on this technique is shown in Figure 9.

![Figure 8 DPWM Reduced Operating Clock Frequency](image)

B) Delay-Locked Loop (DLL)

A block diagram of the DLL used to generate \( f_{clk} \) (the input clock (clk) delayed by exactly 1 period) is shown in Figure 11. It is based on the architecture in [10] and has a lock range of 35 MHz to 145 MHz. To achieve low jitter operation (low supply and substrate noise sensitivity), the buffer stages in the VCDL use differential clocked buffers.

![Figure 10 DLL Transient Response (reference clock (clk_ls) = 142MHz)](image)
The simulation shown in Figure 10 illustrates how the phase error increases the vcp voltage (increasing the VCDL delay) until the DLL reaches lock.

As discussed in Section IV (A) the DPWM module generates the select signals for choosing which DLL clock phase should reset the PWM signal from high to low during the switching period. To minimize the propagation delay for the DLL clock phase through the 8:1 mux (providing adequate clock setup time for the PWM flip-flop) the clock phases are separated from the decode logic within the mux and passed to the output via a transmission gate. Figure 12 shows one slice of the 8:1 mux illustrating how clk_d is fed directly to the T-Gate from the DLL.

On completion, the programmable DPWM module and DLL were simulated together using the Cadence mixed signal simulator SpectreVerilog. The mixed signal simulations were carried out using the synthesized gate-level netlist for the programmable DPWM module and transistor level netlist for the DLL.

Figure 13 shows a transient response for the DPWM programmed for 10-bit resolution with a switching frequency of 997 kHz. The ADC S&H pulse is generated at $dk/2 \times Tsw$ (or $Tsw/2$ when zero value duty-cycles are programmed). The PWM waveforms reflect the programmed duty-cycle values ($d<11:0> = d_k$) on a switching cycle-by-cycle basis. The DLL (vcp) shows a stable first order response throughout the simulation once the delayed clock (clk_d) locks to the reference clock (clk).

Figure 14 shows a zoomed in view of Figure 13 at the point where PWM2 goes low between markers A and B. As the duty-cycle value for this switching cycle is hex’2DC ($b_2-b_0 = 0$),

V. SIMULATION RESULTS

The DPWM digital and analogue sub-blocks were initially designed and simulated as standalone components using standard ASIC design practices. The programmable DPWM module was synthesized to a CMOS 0.35-µm standard cell library with the final netlist having a gate count of < 2.7 k gates (1 gate = 1 NAND gate (2 inputs)).
3'b100) phase four of clk d is used to reset PWM2. On the clock cycle when \( b_n - b_3 = 0 \) clk_d4 is selected by the DPWM module to reset PWM2 (all PWM clocks are in phase with clk_d except for the reset condition on each switching cycle).

Figure 15 shows a transient response for the DPWM programmed for 6-bit resolution with a switching frequency of 15 MHz (delta between markers A and B = 64 ns).

![Transient Response](image)

Figure 15 PWM and ADC_SH Outputs for \( f_{sw} = 15 \) MHz @ 6-bit resolution

VI. CONCLUSIONS

This paper describes a programmable DPWM IC architecture capable of producing flexible PWM pulse patterns supporting switching frequencies in the range of 100 kHz (@12 bits resolution) to 15 MHz (@ 6 bits resolution). In addition the DPWM generates an optimised synchronisation S&H pulse for the ADC. Versatility in the PWM patterns is achieved through programmable dead/delay times eliminating the requirement for external timing components.

The IC has been designed using a low cost well-established 0.35-µm CMOS process following standard ASIC design practices. Simulation results verify the new DPWM architecture and show it to be a more practical and advantageous alternative to previous DPWM ICs.

REFERENCES


