

Implementation of a Digitally-Controlled, Isolated DC/DC Converter with PMBus

Don Alfano, Ka Leung
Silicon Laboratories Inc.

Abstract - This paper describes a digitally-controlled, isolated 35 W half-bridge dc-dc converter based on a novel monolithic integrated circuit controller. The digital controller uses a hardware digital signal processor (DSP) for control execution and is managed by a microcontroller (MCU). This combination provides fast response, in-system programmability, and cost-effectiveness making it useable in a wide range of applications. It also enables power delivery and power management control functions to be implemented in a single chip. The dc-dc converter design uses this controller on the secondary side of the power supply and uses a second MCU and a solid-state isolator to digitize and transmit primary-side voltages to the secondary-side, forming a seamless high-performance, fully-digital control architecture. An overview of the secondary-side controller and discussion of the entire system implementation are presented in this whitepaper.

Introduction

Digital control promises significant system performance gains resulting from complex control algorithms that are difficult to implement in analog. This enables improved performance in traditional power architectures, as well as new innovation with the power stages, that was previously impractical due to the lack of a sophisticated control mechanism. System reliability is improved with the elimination of external analog components, which are prone to value changes with age and temperature. The addition of in-system programmability enables the manufacturer to revise system design and generate custom product versions primarily through software modifications.

Unlike analog, digital control introduces latency due to feedback parameter quantization and calculation times [1]. To minimize this effect, digital blocks in the control path must be designed for high data throughput and low latency, the most critical of which are the loop ADC, filter and digital modulator. While various implementations have been reported, the most common forms are based on a programmable DSP, custom hardware or some combination of a processor and custom hardware. The DSP executes discrete time calculation of control variable values in real time [3]. This approach is typically limited by size, power, cost and throughput making it impractical for many system applications. The dedicated hardware-based approach uses fixed-architecture state machines to execute the control algorithm. Hardware can be optimized for cost and performance making this a potentially lower-cost and more efficient approach than the DSP. However, this approach lacks flexibility because the control hardware cannot be significantly changed once fabricated and must therefore be designed for a specific end application. The controller used in this dc-dc converter combines a programmable processor and optimized hardware blocks in a way that extracts the

maximum benefit from each. The first part of this paper provides an overview of the digital control chip. The second part describes the power system hardware and software implementation.

Chip Architecture

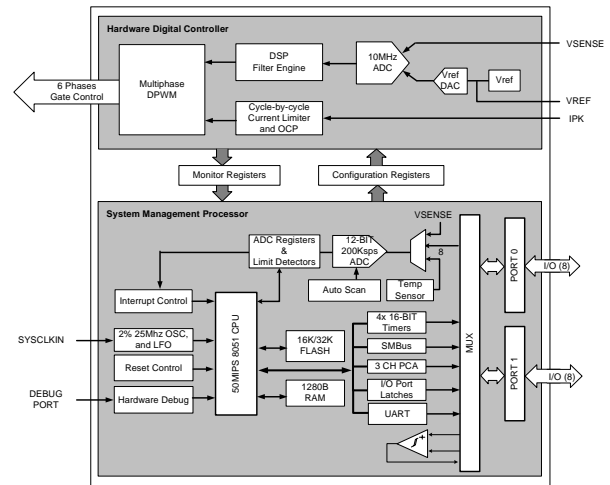


Fig. 1. Digital Power Supply Controller Block Diagram

Figure 1 shows a block diagram of the digital controller integrated circuit. It is partitioned into two main sections: a power supply-specific hardware DSP that implements a high-bandwidth, fully-independent digital control loop, and a software-programmable system management processor for system functionality [5]. The hardware control path includes a high-speed differential ADC with voltage-reference DAC, a programmable infinite-impulse response (IIR) compensating filter engine and a six-phase DPWM finite state machine. The reference DAC, ADC and compensator work together to generate a duty cycle-modulated control signal $[u(n)]$ that controls six independent switching phases. Integrated hardware protection circuits provide cycle-by-cycle current limiting and OCP fault detection.

The system management processor consists of an 8-bit 8051 CPU with a throughput of 50 MIPS, an 8-channel self-sequencing ADC for system parametric monitoring, four 16-bit timers and an SMBus port. This processor provides system initialization, low-bandwidth control loop optimization, fault recovery and housekeeping. It also provides PMBus interface to facilitate communication between the controller and other system intelligence.

ADC and Reference DAC

The control loop front-end shown in Figure 2 contains a differential input, 6-bit 10 MHz Flash ADC with a programmable LSB size of 2 mV to 24 mV. This range allows the ADC to dynamically modify control loop gain and prevent limit cycle oscillation. A reference DAC provides the output voltage reference level. This DAC includes an internal band-gap voltage reference with two percent accuracy over a temperature range of -40 to 125 °C. An external voltage reference may be connected to the VREF pin to attain better accuracy, and the internal reference is disabled in software. The resulting quantized error voltage is connected to the compensator input multiplexer and to a transient detector.

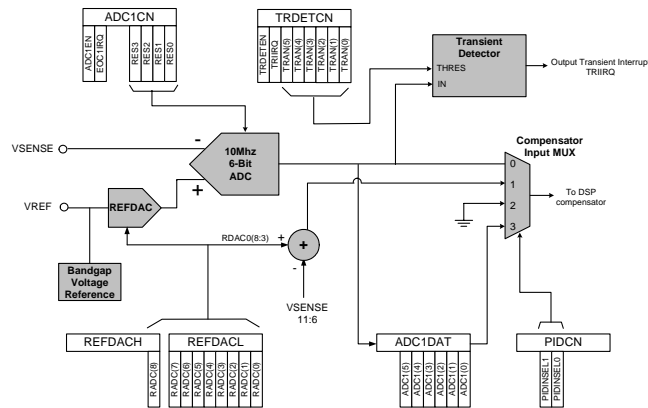


Fig. 2. Control Loop Front-End

In normal operation, the ADC output should change by ± 1 LSB. Movement beyond this range occurs only during a transient. The programmable transient detector monitors the ADC output every $0.1 \mu\text{s}$ and interrupts the CPU when the absolute value of the ADC output exceeds its programmed limits. In response to a transient detector interrupt, the CPU may be programmed to increase the loop gain, reconfigure the pulse waveform or change control modes.

The compensator input multiplexer allows one of four possible front-end control sources to be routed to the digital controller section. These sources are: the 6-bit windowed ADC output for normal operation; a 6-bit ground-referenced difference between the REFDAC level and Vsense for soft-start; ground for system debug and CPU data for custom low bandwidth control applications, such as PFC. In each case, the sampling rate of the DSP is automatically adjusted to the sampling rate of the data source to perform high-speed control loop operation or low-speed start-up operation.

DSP Filters and Dithering

The front-end control term referenced above is processed by the DSP filter providing the necessary compensation to maintain adequate control loop phase margin. The DSP filter consists of a cascade of a proportional-integral-derivative (PID) filter and low-pass filter as shown in Figure 3. All coefficients are dynamically-programmable,

enabling the system management processor section to optimize control loop response in situations such as changes in loading conditions and input voltage.

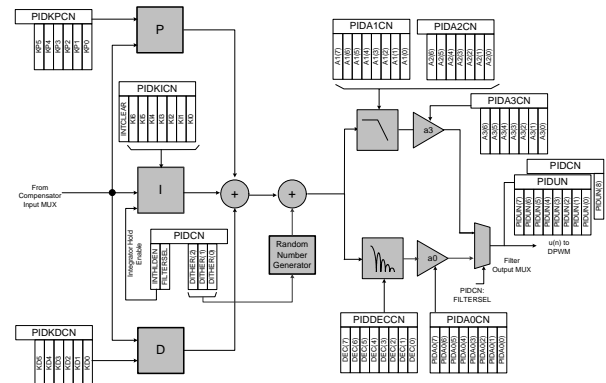


Fig. 3. DSP Compensator Block Diagram

One of two low-pass filters can be selected through the configuration register for use by the PID filter. Both low-pass filters are useful in removing high-frequency power supply noise caused by the zeros of the PID filter.

The PID filter output is a sum of the proportional gain (K_p), the integration gain (K_i) and the derivative gain (K_d) terms derived from the error signal of the input source. Should the integrator input not achieve a zero value, integration will continue until the integrator output is saturated at a maximum or a minimum (integrator wind-out). This event adversely affects control loop response because the integrator requires additional recovery time to return to its normal operating range as the loop attempts recovery. The most common cause -of integrator wind-out is cycle-by-cycle current limiting, which truncates the PWM duty cycle during a current limit cycle. Built-in anti-wind-out hardware protects against integrator wind-out by inhibiting integration action during current limit cycles.

The PID transfer function provides one pole and two zeros. The output of this filter is passed to one of two selectable low-pass filters. The first low-pass filter has two programmable poles and one zero at one-half of the sampling frequency ($f_s/2$). This filter's high sampling rate (typically 10 MHz) updates the DPWM multiple times in a given switching cycle for fast transient response. This filter also has a non-unity dc gain to allow the CPU to temporarily boost control loop gain during a transient for faster recovery. The second (decimation SINC) filter generates multiple zero, which can be placed at the DPWM switching frequency and its harmonics to minimize loop-switching noise.

Dithering controls low-frequency oscillation tones by breaking up limit cycle sequence and improves effective resolution of the DPWM without increasing clock frequency beyond practical levels. The dither has a Gaussian noise distribution property and is generated digitally by a linear feedback shift register.

DPWM

The DPWM generates up to six timing phases that can be modulated by hardware or software in real time. The DPWM is designed for flexibility and supports PWM and phase modulation schemes. Phase-to-phase timing can be programmed for a fixed dead time; or the system management processor can dynamically vary the dead time during converter operation to account for temperature, loading and input voltage variation. The DPWM can be clocked at 25 MHz, 50 MHz or 200 MHz resulting in a practical PWM frequency range of 48 KHz to 1.2 MHz.

As shown in Figure 4, the main input of the DPWM is the compensated duty ratio $u(n)$ from the compensator. The DPWM input multiplexer selects either $u(n)$ or data system management processor data as the input modulation source. The multiplexer output connects to a pair of symmetry locking registers that are useful in applications where the duty cycle of the DPWM is slaved to a selected master pulse. The symmetry lock output is connected to four $u(n)$ limiter circuits allowing the system management processor to adjust the offset and limits of $u(n)$. This results in up to four individual corrected $u(n)$ functions.

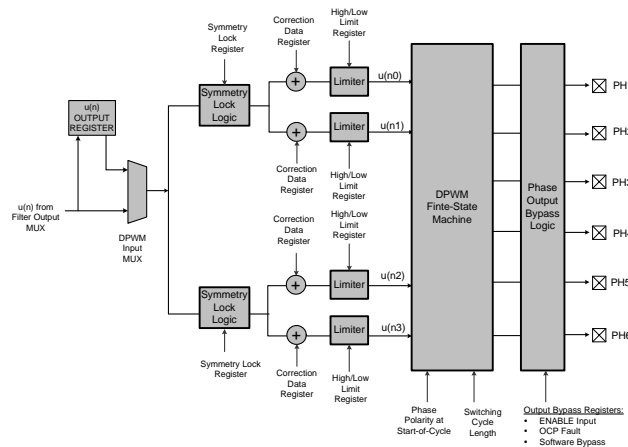


Fig. 4. DPWM block diagram

The heart of the DPWM is a finite-state-machine (FSM) based timing generator where each edge of each DPWM phase is implemented in a separate FSM [5]. Each edge of each DPWM phase output, h , can be programmed to have a timing dependency to any of the other DPWM phases or to itself. The dependency is set up by data contained within individual configuration registers for each FSM. Once initialized, the input to each FSM is hardware-modulated to select from one of the four corrected $u(n)$ functions, and each $u(n)$ can be mapped to any FSM. Positive, negative or MCU-controlled phases, overlapping or non-overlapping, can be implemented with this architecture. As shown in Figure 5, each edge of each phase is controlled by one of either a $u(n)$ -modulated, an absolute or a relative command to construct the output pulse of each phase. By providing this FSM-based architecture, each phase edge can be defined separately so as to generate the associated edge virtually independent of the other edges. With such an architecture based in hardware, each FSM decision requires

only one or two clock cycles as compared to that required by an instruction-based engine.

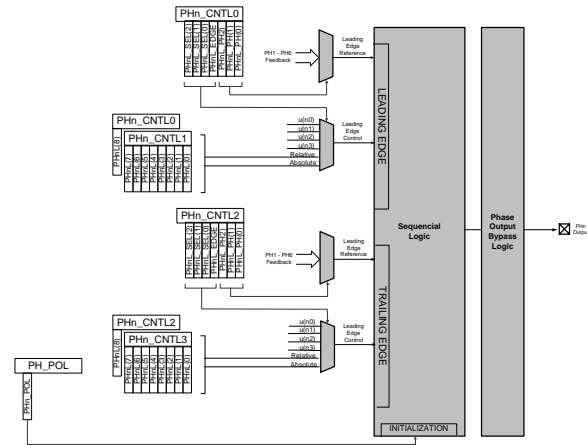


Fig. 5. DPWM Finite-State Machine Block Diagram ($n = 1, 2, 3, 4, 5$ or 6)

Output bypass logic provides safe stop states for all phase outputs in the event of a predetermined condition. When this predetermined condition occurs, the bypass logic overrides the DPWM output by forcing each phase output into user-defined states during power supply shutdown, thus placing the power supply in a known safe state. The bypass logic can be programmed to occur automatically during overcurrent protection or when an external pin is enabled. The bypass operation can also be initiated by the system management processor section in software. Each of these bypass conditions has an associated programmable stop pattern.

System Management Processor

The system management processor, shown in Figure 1, implements a standard 8051 organization and peripherals. The MCU core employs a pipelined architecture that executes most of its instructions in one or two system clock cycles, and the MCU is capable of running at 50 MHz with a peak throughput of 50 MIPS.

The analog front-end of the MCU consists of a 12-bit, 200 kbps ADC and associated auto sequencing logic, limit registers and temperature sensor. The ADC has eight input channels, and each channel has a corresponding output register and limit detector. The limit detectors compare the converted output to user-programmed limits and generate an MCU interrupt when these limits are exceeded. The ADC is also equipped with auto sequencing logic, which does not require MCU supervision during data conversion. The auto-sequencing feature automates the analog data acquisition process and enables system protection functions, such as input over-voltage protection, input under-voltage lockout, output voltage monitoring and over-temperature protections, to be implemented in firmware. The MCU has an internal temperature sensor, which monitors chip temperature from -40 to 125 °C. This temperature monitoring is also useful in providing

necessary compensation to optimize power efficiency of switchers and gate drivers.

The system management processor section also features four counter/timers for use with device peripherals or for general-purpose use. Other system functions include a high-precision oscillator, a phase-locked-loop based clock multiplier, a software-coded PMBus, a UART and two GPIO ports.

Isolated dc-dc System Design

The half bridge converter of Figure 5 operates at a PWM frequency of 400 kHz and uses secondary-side control for optimum transient response [1,2,5]. The PH3 and PH4 outputs control the synchronous rectifiers via a dual driver I.C. A small MCU on the primary side digitizes the input, capacitor node and average primary current signals and transmits the results to the secondary-side controller via single-chip digital isolator IC.

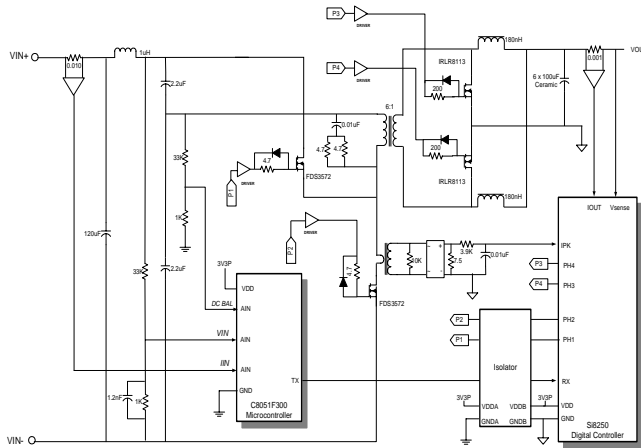


Fig. 6 – Half Bridge Circuit Diagram

The peak current sensing circuit uses a current transformer specifically designed to meet isolation voltage requirements and a full-wave rectifier and filter circuit. The output connects directly to the peak current detector input of the controller.

When power is applied, the controller executes an internal hardware reset followed by firmware initialization of all register and RAM-based parameters. The controller remains in a low-power state, monitoring digitized VIN data from the primary-side MCU. Controller firmware fully enables all peripherals and interrupts (including all system safeguards) and initiates soft-start when the value of VIN exceeds the UVLO threshold.

A general-purpose timer is used as a time base for soft-start. With each timer interrupt event, firmware increments the programmable VREF until the supply output voltage is within a specified range, at which time steady-state operation begins.

During steady-state operation (shown in Figure 7), the MCU operates in interrupt mode where hardware events divert program execution to specific routines in priority order.

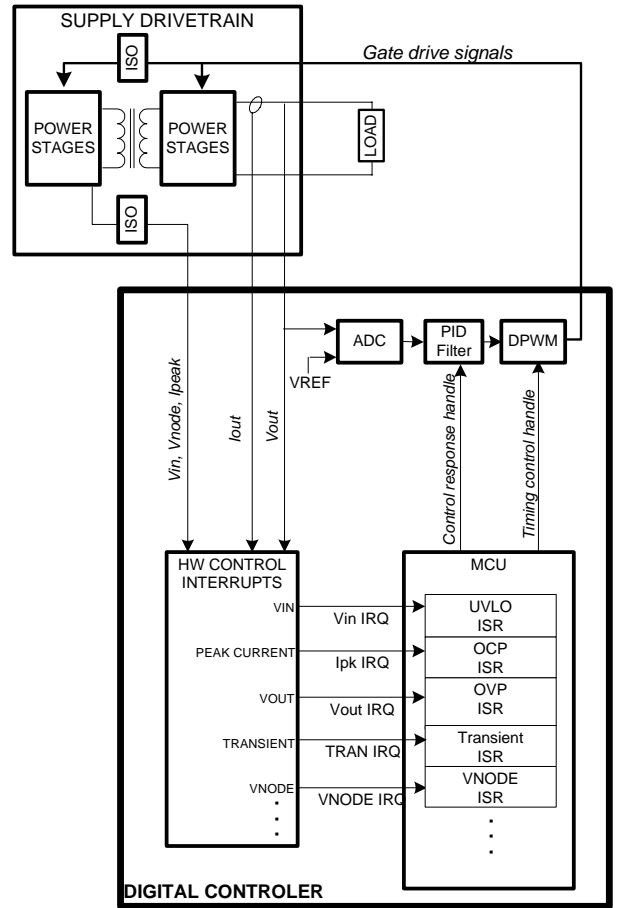


Fig. 7 Control Schematic

Transient-Triggered Nonlinear Control

The transient detector interrupt can trigger a change in loop compensation or simply increase the compensator gain. However, widening loop bandwidth can lead to instability [4]. A detection method shown in Figure 8 can be used to detect the actual onset of instability. The system management processing section monitors the pattern of the output voltage error signal after transient detection is executed. Upon detection of output instability, the gain of the programmable DSP compensator is reduced, and the system returns to a regulation with a slower dynamic response but adequate phase margin.

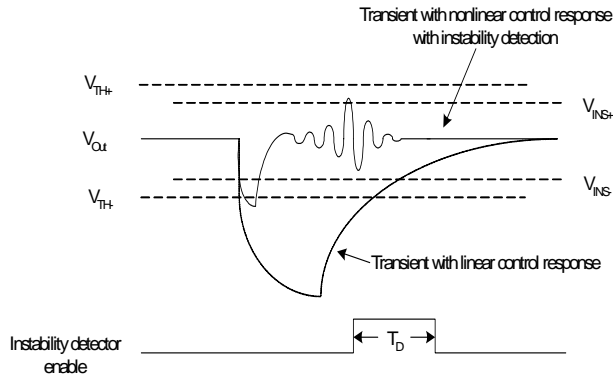


Fig. 8. Transient with Nonlinear Control

During steady-state operation, the MCU manages relatively lower bandwidth tasks on an interrupt basis, and hardware performs the following operations in parallel:

- Output voltage regulation – The programmable signal processors update duty cycle every 100 nS.
- Current limiting – The current detector provides cycle-by-cycle transformer current limiting.
- Analog-to-digital conversion – Input and output voltage, average current and local temperature are continuously scanned, digitized and stored by the self-scanned, 12-bit ADC.
- Parametric limit monitoring – Hardware limit detectors compare each converted parameter to prescribed limits and generate vectored interrupts when the parameter is out-of-range.
- Background task scheduling – A hardware timer (scheduler) generates periodic interrupts that vector the MCU to maintenance routines such as statistical calculations, variable normalization, communication and other system management tasks.

System safeguards (e.g. OVP) are performed in much the same way as the transient response example previously described. When a measured parameter is out of range, an interrupt diverts firmware execution to a specific ISR. This is true for OVP, OCP, OTP and other system-related fault conditions. Other system power management functions such as voltage positioning, sequencing, primary-side input filter bias correction, efficiency optimization and load sharing are implemented in software as well.

Experimental Results

Figure 9 shows the completed half-bridge evaluation board. (Note: this board has added functionality to facilitate test and evaluation and consequently more components than are required for the control function alone).

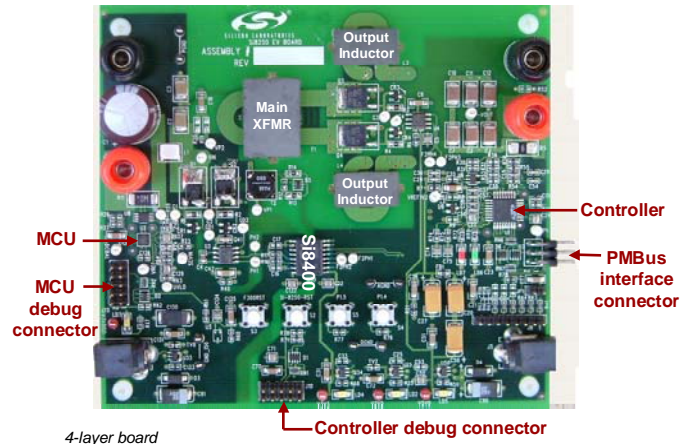


Fig. 9. Half-Bridge Evaluation Board

This board implements all circuitry shown in Figure 6 and has separate debug connectors for the primary-side MCU and secondary-side controller. A SMBus-to-USB adaptor is included to allow the user to connect the controller directly to his computer and use the included PMBus GUI to command action or request data.

The oscilloscope photos of Figure 10 demonstrate the transient response improvement brought by non-linear control.

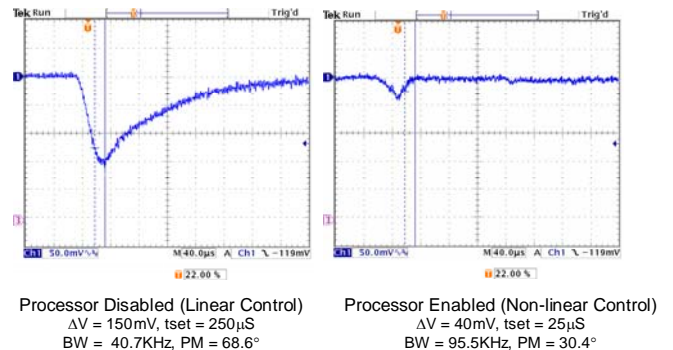


Fig. 10 Linear vs. Non-Linear Control Response

In each photo, a 57 percent load change was applied at a 1 A/ μs rate. The system management processor was disabled in the photo on the left, thereby providing a linear control response. The photo on the right was taken with the system management processor enabled, allowing the system management processor to perform nonlinear control. As shown, the nonlinear control case significantly reduced both the transient voltage magnitude and settling time.

PMBus

The SMBus port onboard the controller serves as the physical interface layer for PMBus. The PMBus protocol layer resides in the system management processor where PMBus exchanges are serviced by the CPU on an interrupt basis. PC-based GUI software allows a PC to act as the system master, facilitating system development and diagnostics.

Summary:

An isolated half-bridge dc-dc converter based on a unique digital controller architecture has been described in this whitepaper. The results demonstrate competitive performance with added functionality and flexibility. This is enabled by a novel controller architecture, which offers levels of performance found in ASIC solutions and flexibility typical of Flash-based processors. As such, this controller is useful in half-bridge and many other switchmode topologies.

References

[1] A. Peterchev, J. Xiao, S. Sanders, "Architecture and IC implementation of a digital VRM controller", IEEE Trans. on Power Electronics, Vol. 18, No. 1, January 2003 pp. 356-358.

[2] A. Prodic and D. Maksimovic, "Digital PWM controller and current estimator for a low-power switching converter", Proc. the 7th workshop on computers in power electronics, pp. 123-128, 2000.

[3] E. Dallago, M. Passoni and G. Sassone, "lossless current sensing in low voltage high current DC/DC modules supplies", IEEE Trans. Industrial electronics, vol. 47, pp 1249-1252, Dec. 2000.

[4] B. Miao, R. Zane, D. Maksimovic, "Detection of instability and adaptive compensation of digitally controlled switched-mode power supplies". IEEE APEC 2005, pp. 63-68

[5] K Leung "Optimizing System Operation Using a Flexible Digital PWM Controller," 2005.