POLITECNICO DI TORINO
Repository ISTITUZIONALE

A Framework for the Development and Monitoring of Digital Control in Power Converters

Original
A Framework for the Development and Monitoring of Digital Control in Power Converters / Crovetti, P., Gregoretti, F., Musolino, F., Usmonov, M.. - ELETTRONICO. - (2020), pp. 76-81. (7th International Conference on Electrical and Electronics Engineering, ICEEE 2020 tur 2020) [10.1109/ICEEE49618.2020.9102523].

Availability:
This version is available at: 11583/2840840 since: 2020-07-21T10:31:51Z

Publisher:
Institute of Electrical and Electronics Engineers Inc.

Published
DOI:10.1109/ICEEE49618.2020.9102523

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

18 November 2026

A Framework for the Development and Monitoring of Digital Control in Power Converters

Paolo Crovetti
Department of Electronics and Telecommunications
Politecnico di Torino
Torino, Italy
paolo.crovetti@polito.it

Francesco Gregoretti
Department of Electronics and Telecommunications
Politecnico di Torino
Torino, Italy
francesco.gregoretti@polito.it

Francesco Musolino
Department of Electronics and Telecommunications
Politecnico di Torino
Torino, Italy
francesco.musolino@polito.it

Maksudjon Usmonov
Department of Electronics and Telecommunications
Politecnico di Torino
Torino, Italy
maksudjon.usmonov@polito.it

***Abstract*—The design of advanced digital controllers for power converters requires dedicated hardware which may result in increased complexity and development time. As the number of control variables increases, operations such as monitoring, debugging, testing and optimization become more difficult. The paper presents a methodology and a prototype of a system that have facilitated non-invasive monitoring of the controller of power converter in real-time and performing more effective automated tests. This has been achieved by leveraging the tremendous reconfigurability of modern System-on-Chip (SoC) devices that integrate both Field Programmable Gate Arrays (FPGA) and high-performance standard Hard Processor System (HPS) in one package with built-in, programmable interconnects. In this proposed architecture, FPGA section implements digital controller and HPS handles monitoring and interfacing for remote supervision by re-using standard Linux system libraries. The monitoring system sets the controller parameters, collects all internal controller signals and organizes them. It provides an interface through ethernet for external data processing environments such as MATLAB© or Python, which in turn allows to integrate other instrumentation sources in order to execute automated tasks. This paper will also present the application of the proposed methodology to the investigation of the quantization-induced limit cycles in converters.**

Keywords-component, formatting, style, styling, insert

I. Introduction

Compared to analog systems, digital controllers in switch-mode power converters offer serveral advantages such as lower sensitivity to parameter variations and better versatility in terms of cost and design re-use [1]–[4]. The ability to easily implement advanced control laws through software enhances the flexibility a nd r econfigurability, th us, al lowing to us e a single hardware implementation for meeting a wide range of product requirements [5]–[11]. Moreover, standard and well-established automated design tools can be applied to greatly reduce development time and design effort, whereas complex control strategies with math-intensive and highly parallel algorithms can be described and implemented by an Hardware Description Language (HDL) [12]–[14].

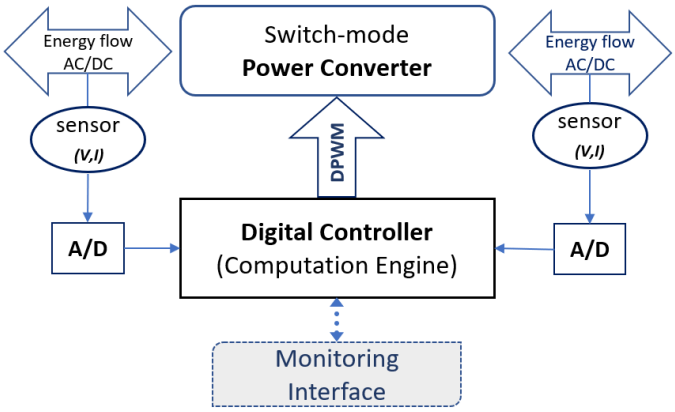


Figure 1. Digital control of power converter - Generic block diagram

Modern power converters with digital controllers, as summarized in Fig.1, mainly consist of these functional units:

- A switch-mode power converter circuit;
- Sensors for measuring electrical quantities at different nodes;
- Analog-to-digital converters for sensor data acquisition;
- A computation engine, necessary for implementing digital control laws, filtering, identification and etc.;
- A monitoring interface, mostly optional, but useful for testing and supervision of controller operation;

Current computational engines used for digital control algorithms in power converters include microcontrollers, digital signal processors (DSP), application specific integrated circuits (ASIC), complex programmable logic devices (CPLD) and field-programmable gate arrays (FPGA) [15]–[19].

However, as the switching speed of modern converters increases and control intervals shorten, very high-performance processing units are required to implement some algorithms in fully digital mode. Moreover, real-time monitoring, debugging

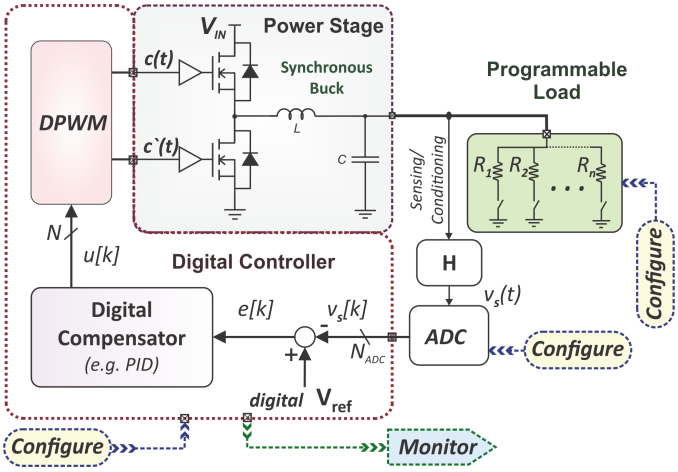


Figure 2. Outline of digital control loop for synchronous buck converter with digital configuration and monitoring interface. Output is connected to programmable load module. Both "configure" and "monitor" represent digital interfaces. $c(t)$ and $c'(t)$ are complementary pulse-width-modulated (PWM) signals that drives the converter switches at f_{sw} frequency.

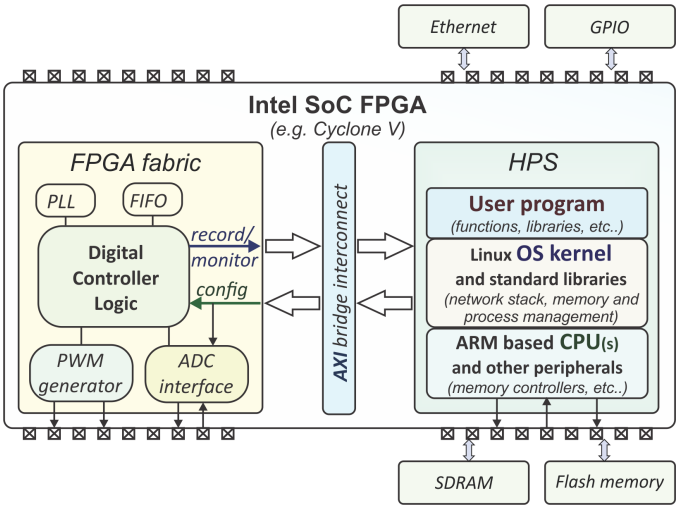


Figure 3. Basic internal modules of Intel SoC package (e.g. Cyclone V) used to build a digital controller. FPGA and HPS fabric communicate through built-in AXI (Advanced eXtensible Interface) bridge. FPGA part implements digital controller, while HPS performs configuration and monitoring/ recording (see also Fig.2). ADC interface logic written in HDL interacts with external ADC chip

and verification of largely parallel, high-speed operations or auto-tuning of critical parameters in digital feedback loops require special setup. When experimenting different control behaviors in a real system, one needs to perform extensive tests and validations for many possible scenarios, while monitoring all external and internal states of the digital control loop for correctness. So, it becomes very time consuming and laborious task if not done in a programmatic manner. With the advent of system on chip (SoC) devices, and their increasing affordability, above-mentioned challenges of computation and monitoring tasks are eased. SoCs that feature both FPGA and Hard CPU units in one package, with built-in, very efficient, high speed interfaces allow to partition the tasks accordingly. Specifically, monitoring of FPGA parallel operations can easily be done through an embedded user program running inside CPU unit, which allows to analyze and test innovative control laws for power converters efficiently and with great details. This also enables to verify novel control methods for different operating conditions, to look for possible glitches or non-linearities, intrinsic to digital control, and to the mitigate of such issues.

The research activity presented in this paper describes a methodology and a prototype of a cost-effective, laboratory testbench implemented with FPGA-SoC board (Fig.3), which allows to make real-time, non-invasive monitoring of digital power converters. The implemented system has enabled to perform automated tests with varying parameters. In particular, it provides a MATLAB interface for remote supervision and data analysis and testing of complex conditions can be performed on a given converter. The ability to capture and investigate internal values of critical digital variables in real-time facilitates debugging of a control algorithm with any complexity. In the proposed architecture, a digital controller has been implemented in the FPGA section using HDL, whereas HPS handles configuration of parameters and interfacing for remote supervision. The "DE1-SoC" evaluation board [20], that incorporates *Cyclone V SoC* from Intel, has been used for the testbench.

The paper is organized as follows: the overall system behavior and the structure of each module are described in Section II providing a few examples of the design procedures. Section III describes the prototype which implemented a digital PID (proportional-integral-derivative) controller for a synchronous buck converter. As a case study, quantization-induced limit cycle oscillations (LCO) has been investigated and a recently proposed LCO suppression method, based on Dyadic digital pulse modulation (DDPM) [21], is examined to demonstrate the system capabilities. Finally, in Section IV some concluding remarks are given, and further future works are discussed.

II. Methodology Description

We use, as an example, the implementation and the characterization of a digital control loop for a synchronous buck converter. Fig. 2 illustrates the general outline of a converter under experiment, whose output voltage is regulated by a digital controller, which in turn may be configured and monitored through digital interfaces. In order to observe the behavior of the controller in different load conditions, both in steady-state and transient mode, the converter output is connected to a programmable load module that can also be configured through a digital interface. For testing and verification, one may also need to assess the behavior of the control with different Analog-to-Digital Converter (ADC) configurations, thus the configurable interface for ADC has been included. All digital components and their interconnections are implemented inside *Cyclone V SoC* from Intel that integrates FPGA and HPS units in one package. For this particular experiment, "DE1-SoC" evaluation board is used. It provides also a number

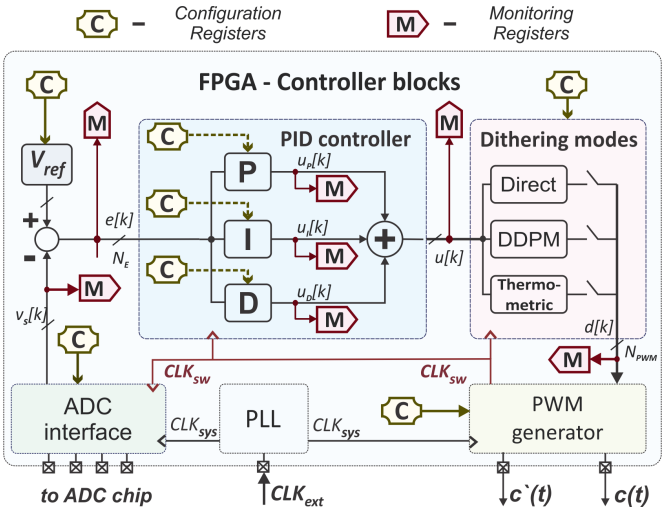


Figure 4. Overview of controller blocks inside FPGA. Each block has its own set of "Configuration" and "Monitoring/Recording" registers whose values are exchanged with HPS. Registers are set of dedicated memory signals that store controller data.

of external peripherals, such as ADC, extrenal memory,video and ethernet interfaces. The functions of the digital control loop are partitioned as depicted in Fig.3.

The user program running on top of the Linux OS in the HPS side configures a nd m onitors t he c ontroller l ogic which is deployed in FPGA side. More detailed explanations about the operation of each module and sub-module are given in the subsequent sections.

A. FPGA Blocks

PID-based compensator has been adopted in the digital controller for its simplicity in terms of design and description. However, any other kinds of control techniques with greater complexity can be deployed instead of PID, the only limitation being the number of available logic elements on a given SoC package. The FPGA fabric inside Cyclone V features 85k logic elements, of which only less than 10% has been occupied by the whole system with all its components. Fig. 4 and 5 illustrate the outline of the controller and communication blocks that are implemented in the FPGA section using HDL.

The external fixed clock signal CLK_{ext} is provided to the built-in PLL (phase-locked-loop) block, that will generate the system clock CLK_{sys} with the desired frequency for all other FPGA blocks. The digital *PWM generator* block takes CLK_{sys} and the duty cycle value $d[k]$ as inputs, and generates two complementary PWM signals ($c(t)$ and $c'(t)$) at the output, which are separated by a dead-time interval. The frequency of generated PWM signals and their resolution are configured through dedicated registers. Moreover, in order to synchronize with other blocks, it produces another clock signal CLK_{sw} at the switching frequency of the converter.

Each module incorporates an individual set of memory blocks called registers. They are divided into two categories: *configuration* and *monitoring* registers. Modules obtain their settings from their own set of configuration registers, while they write all the acquired and computed data to the respective monitoring registers. At the HDL level, those registers are just a set of internal variable *signals* with the specific length.

For example, *ADC interface* performs an acquisition of converter's output voltage $v_s[k]$ by sending proper digital commands to the external ADC chip. This block controls the sampling rate and the resolution of the ADC according to the given configuration. Digital reference voltage value $V_{ref}[k]$ is provided through another register. By comparing $v_s[k]$ and $V_{ref}[k]$, error signal $e[k]$ is calculated at the same time instance, since all calculations are done in parallel. Both acquired $v_s[k]$ and calculated error value $e[k]$ are written to their monitoring registers, as depicted in Fig.4 where configuration registers are shown as **C** blocks andmonitoring ones as **M** blocks.

Internal blocks of PID controller are also designed in the similar manner, in which coefficient values (K_p, K_i, K_d) are configurable and the results of each section (u_p, u_i, u_d) can be monitored independently.

A dithering module is added between PID and PWM generator for testing different types of digital dithering methods in order to observe their effect on the converter output voltage. Applying various dithering methods on the discrete signal $u[k]$ coming out from PID compensator allows to minimize the non-linear LCO of output voltage. Dithered duty cycle values $d[k]$ at each switching cycle can be monitored as well. Extensive analysis related to LCO-free operation of the DC-DC converter were studied and detailed experimentation results were given in [22]. In the current testbench, two different techniques, namely *thermometric* and *dyadic* (DDPM) based ditherings, are implemented inside the module. The type of the selected dithering technique may be changed through the module's configuration register set.

Furthermore, the digital controller may be configured for open loop operation as well. This may be very useful to isolate PWM and dithering blocks, bypassing the error generator and PID compensator modules.

All *configuration* and *monitoring* registers are connected to the HPS through the built-in bridge interconnect from the standard AXI bus whose purpose is to enable the HPS to access the FPGA signals like its own peripherals with unique assigned base addresses. The connection is based on a *master-slave* relation. Both FPGA and HPS can be configured either master or slave devices, depending on which section is likely to initiate the process. In the current configuration, all FPGA registers are seen as slave peripherals, whereas HPS acts as a master device.

Figure 5 describes the implementation overview of the AXI bus interface from the FPGA side. Many specific steps of assigning base addresses and making FPGA to AXI connections are automated by *Quartus - Platform Designer* tool [24]. This tool provides numerous pre-designed blocks that are easily added to the design using graphical user interface. Programmable Input-Output (PIO) registers and AXI Master/Slave Logic are the examples of those pre-designed blocks. Once the modules inside *Platform Designer* are ready, the

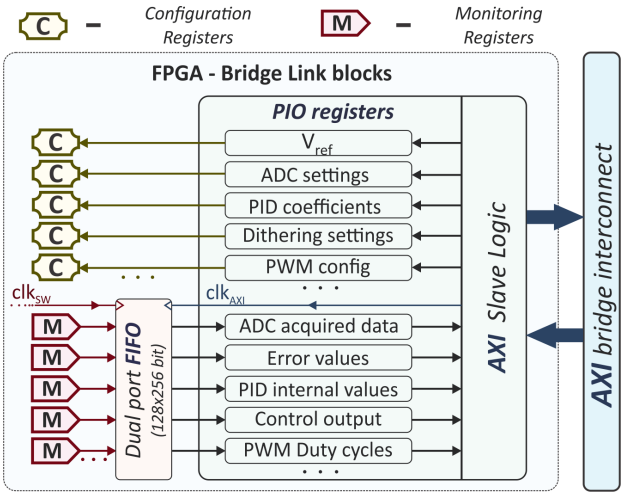


Figure 5. Section of AXI interface blocks inside FPGA. Set of Programmable Input-Output (PIO) registers are used to connect FPGA blocks with built-in AXI bridge interconnect. PIO registers are seen as slave peripherals from HPS side. "Monitoring" data are fed to the HPS through Dual-port FIFO for synchronization.

HDL code of those blocks are generated automatically.

The *configuration* registers are then connected directly to their corresponding PIO block signals. Meanwhile, *monitoring* registers are linked via dual-port FirstIn First Out (FIFO) modules. The purpose of the FIFOs is to synchronize data transfer from FPGA to HPS, since they work at different clock rates with completely separate clock sources. The values inside *monitoring* registers are updated at every switching cycle of the converter, hence FIFOs prevent missing any data. For the current experiment, FIFO depth was chosen very conservatively to be 128 words. However, depending on the project requirements, it can be adjusted to have different size.

B. HPS Program Fow

The HPS portion of *Cyclone V* SoC features a dual-core CPU based on ARM Cortex-A9, as well as many other peripherals. There are two ways of developing a software for the HPS section: bare metal and Linux OS based. In bare metal application, a developer needs to write each functionality and has a great control over each peripheral, memory and timing. However the codebase becomes tedious to develop and maintain. Meanwhile, in Linux based development, many functionalities related to file handling, communication protocols and user interface are provided by the system.

The main role of the program running inside the HPS is to link external data processing and visualization tools with the FPGA, as shown in Fig.6. In other words, HPS provides an interface to higher-level development environments such as MATLAB. High speed link can be established to the host computer running MATLAB, through standard ethernet interface that is built-in peripheral integrated in the HPS section. And the ability to use standard Linux OS with its standard services and libraries allows a fast and easy development of HPS software. The present work uses the

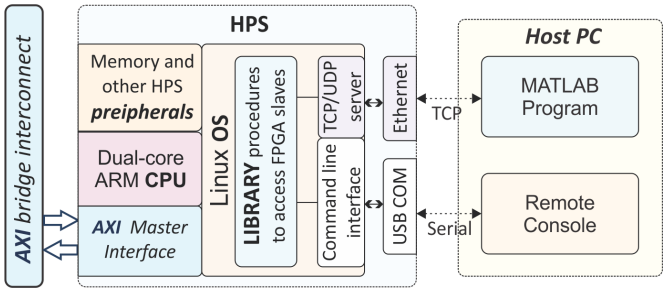


Figure 6. Block diagram of the HPS program. HPS communicates with the Host PC through dedicated ethernet port, by using TCP-IP protocol.

MATLAB environment and its tools to communicate with the SoC board. The program running on the HPS first initiates the TCP server. The MATLAB program then connects to the running server using the ethernet connection, then sends the converter configuration data which are then written to the *configuration* registers inside FPGA. After this, HPS program starts recording the values of the required *monitoring* registers from the FPGA and forwards them to MATLAB.

The ability to use the standard Linux operating system on a board extensively assists the process of establishing communication to the host computer.

Furthermore, the board features a USB-to-COM link, for a simple serial connection to the Host PC, that allows to open a remote console and execute command line tools of the Linux environment running inside HPS.

The TCP server and command line programs are both written in C programming language. They use the custom library procedures for accessing FPGA registers. For example, each register base address is mapped in Linux kernel memory space with the procedure named as *initializeFPGA()*.

C. Interfacing with MATLAB

MATLAB provides a user friendly environment for data analysis and visualization. Many electronics laboratory instruments, such as programmable power supplies, digital oscilloscopes or spectrum analyzers, have a toolboxes that can easily be integrated into MATLAB. So, it is possible to generate completely automated testing results within MATLAB.

In Fig.7, the simplified flowchart blocks of both HPS and MATLAB programs are illustrated with some of the function names that are developed.

In the following chapter, few examples of the testing results generated from MATLAB, connected to the SoC board and programmable power supply, are given.

III. Prototyping and Testing Results

The block diagram of the experimental setup is shown in Fig.8. As it can be seen, the DE1-SoC evaluation board featuring *Cyclone V* chip is handling the control and monitoring of the synchronous buck converter and programmable load, while host computer is running MATLAB program which collects and analyzes all the data coming from the converter and other instruments.

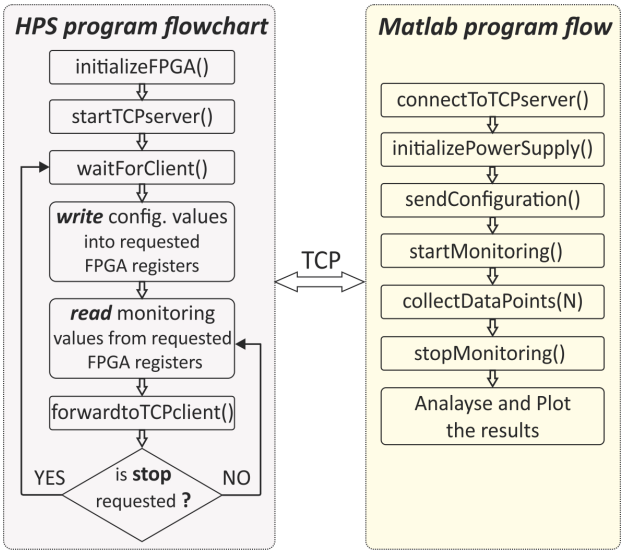


Figure 7. General Flowchart of HPS and MATLAB programs

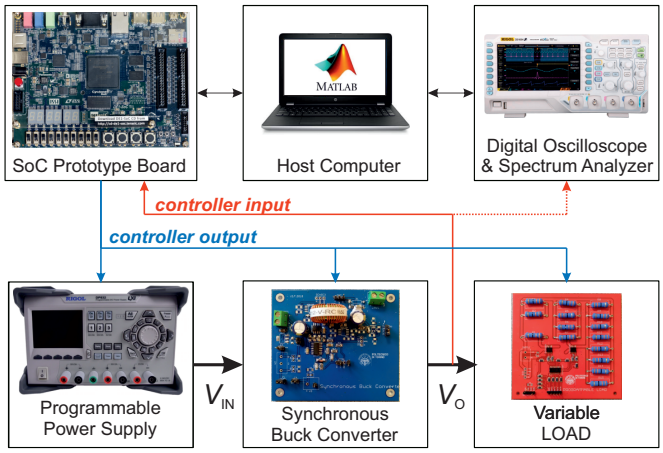


Figure 8. Block diagram of the experimental setup. [22]

Following parameters can varied in any combination and configured during run-time of the converter:

- ADC resolution: N_{adc} (from 1-12 bit resolution)
- ADC sampling rate: up to 250 ksps
- Digital voltage reference V_{ref}: range is 0 to 10 V
- PID coefficients: Kp, Ki, Kd
- Dithering modes: (Direct, Thermometric, Dyadic [21], [23])
- Dithering word size: (M_{dith} - number of bits used for dithering)
- Digital PWM resolution: N_{pwm} = [2-9 bits], maximum is limited by the system clock
- PWM switching frequency: f_{sw} (100 kHz or 200 kHz)
- Load resistor: (4 switches, each driving binary weighted resistance values. Overal 16 different load resistances that can be changed almost instantly)
- Supply Voltage: 0-30 Volts.

Many experiments on the converter control have been

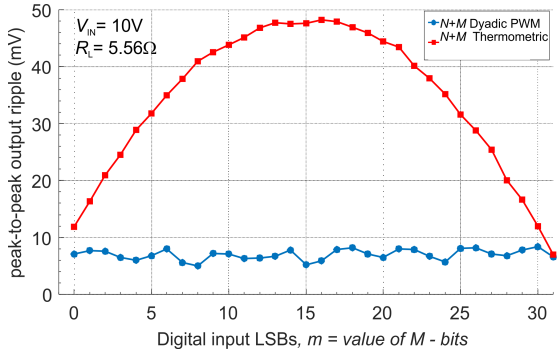


Figure 9. Measured dithering-induced ripple voltage (open loop configuration). Comparison between Dyadic and Thermometric ditherings under the same M_{dith} resolution enhancements. [22]

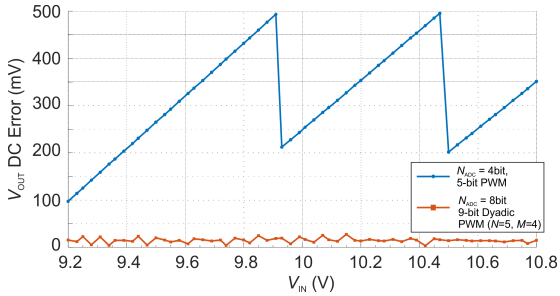


Figure 10. Static error in the output voltage under different input voltages (V_{in}). Comparison between when N_{adc} = 4bit and N_{adc} = 9bit [22]

performed using the testbench, with varying parameters. It has been used to study the effects of quantization on the output voltage when the power converter is controlled digitally. Obtained results have been published extensively in separate research papers [22], [23].

Figures 9-11 show some of these results obtained by running automated data collection tasks and analysis in MATLAB. Each of these MATLAB figures comes from the execution of one individual script which in turn generates a sequence of experiments in the controller with varying parameters and whose data are acquired from the monitoring units. For instance, in Fig.10, the static error in the output voltage is automatically acquired and plotted for 30 different input (Vin) voltage values for two different ADC resolution and dithering methods. All these configurations are set sequentially in the MATLAB script.

IV. Conclusions

The implementation of a FPGA-SoC based monitoring system for digitally controlled power converters has been presented in this paper. It has been shown that the proposed methodology and prototype have greatly facilitated non-invasive monitoring of the controller in real-time and resulting in more effective automated testing. This methodology can be applied to experiment other converter topologies with different control algorithms.

Automated programming of complex experiments and data analysis may be performed at high level of abstraction. How-

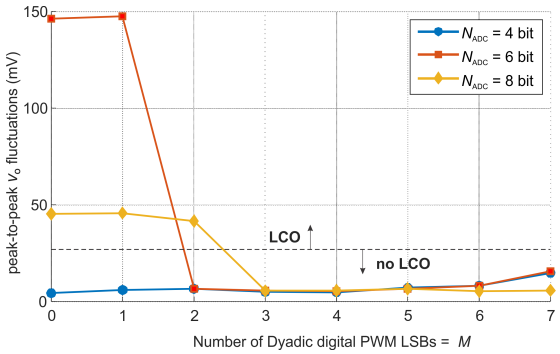


Figure 11. Amplitude of LCOs for N+M-Dyadic digital PWM with N_{pwm} = 5 under different values of M_{dith} for N_{adc} = 4, N_{adc} = 6 and N_{adc} = 8, I_O = 1A load current. [22], [23]

ever, currently, the design of the digital controllers of the converters are performed manually and the insertion of the monitoring and configuration i nterfaces a re a lso performed manually, although in a very systematic way. Future developments are along three main lines

- Automate the process inserting monitoring and configuration registers in the digital controller hardware
- Moving the design of the controller themselves to higher level languages and tools integrating them with the monitoring system.
- Implementing the experiment scripts in a non-proprietary environment and language such as Python, in order to expand the user-base of the methodology.

The final target is the development of a seamless integrated environment for the full design, characterization of power converters.

ACKNOWLEDGMENT

The authors would like to thank the Intel FPGA University Program for donating the DE1-SoC evaluation board.

REFERENCES

[1] D. Maksimovic, R. Zane and R. Erickson, *"Impact of digital control in power electronics"*, 2004 Proceedings of the 16th International Symposium on Power Semiconductor Devices and ICs, Kitakyushu, Japan, 2004, pp. 13-22.

[2] P. T. Krein, *"Digital Control Generations – Digital Controls for Power Electronics through the Third Generation"*, in 7th International Conference on Power Electronics and Drive Systems, Bangkok, 2007, pp. P-1-P-5.

[3] S. Buso, P. Mattavelli, *"Digital Control in Power Electronics"*, Lectures on power electronics, Morgan-Claypool, 2006.

[4] L. Corradini, D. Maksimovic, P. Mattavelli, R. Zane, *"Digital Control of High-Frequency Switched-Mode Power Converters"*, John Wiley and Sons, 2015.

[5] C. Buccella, C. Cecati, H. Latafat, *"Digital Control of Power Converters—A Survey"*, IEEE Transactions on Industrial Informatics, vol.8, no.3, pp.437–447, 2012.

[6] J. Chen, A. Prodic, R. W. Erickson, D. Maksimovic, *"Predictive digital current programmed control"*, IEEE Transactions on Power Electronics, vol.18, no.1, pp.411–419, 2003.

[7] H.-S. Youn, J.-S. Park, K.-B. Park, J.-I. Baek, G.-W. Moon, *"A Digital Predictive Peak Current Control for Power Factor Correction With Low-Input Current Distortion"*, IEEE Transactions on Power Electronics, vol.31, no.1, pp.900–912, 2016.

[8] P. Cortes, M.P. Kazmierkowski, R.M. Kennel, D.E. Quevedo, J. Rodriguez, *"Predictive Control in Power Electronics and Drives"*, IEEE Transactions on Industrial Electronics, vol.55, no.12, pp.4312–4324, 2008.

[9] S. Guo, X. Lin-Shi, B. Allard, Y. Gao, Y. Ruan, *"Digital Sliding-Mode Controller For High-Frequency DC/DC SMPS"*, IEEE Transactions on Power Electronics, vol.25, no.5, pp.1120–1123, 2010.

[10] B.J. Patella, A. Prodic, A. Zirger, D. Maksimovic, *"High-frequency digital PWM controller IC for DC-DC converters"*, IEEE Transactions on Power Electronics, vol.18, no.1, pp.438–446, 2003.

[11] L. Corradini, P. Mattavelli, E. Tedeschi, D. Trevisan, *"High-Bandwidth Multisampled Digitally Controlled DC–DC Converters Using Ripple Compensation"*, IEEE Transactions on Industrial Electronics, vol.55, no.4, pp.1501–1508, 2008.

[12] Siwakoti, P. Yam, E.T. Graham, *"Design of FPGA-controlled power electronics and drives using MATLAB Simulink"*, IEEE ECCE Asia Downunder, pp.571-577. IEEE, 2013.

[13] E. T. Mekonnen, J. Katcha and M. Parker, *"An FPGA-based digital control development method for power electronics"*, IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society, Montreal, QC, 2012, pp. 222-226.

[14] T. Gherman, D. Petreus and R. Teodorescu, *"A Method for Accelerating FPGA Based Digital Control of Switched Mode Power Supplies"*, 2019 International Aegean Conference on Electrical Machnies and Power Electronics (ACEMP) & 2019 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM), Istanbul, Turkey, 2019, pp. 322-328

[15] M. Hallworth, S.A. Shirsavar, *"Microcontroller-Based Peak Current Mode Control Using Digital Slope Compensation"*, IEEE Transactions on Power Electronics, vol.27, no.7, pp.3340–3351, 2012.

[16] D. Hamza, M. Pahlevaninezhad, P.K. Jain, *"Implementation of a Novel Digital Active EMI Technique in a DSP-Based DC–DC Digital Controller Used in Electric Vehicle (EV)"*, IEEE Transactions on Power Electronics, vol.28, no.7, pp.3126–3137, 2013.

[17] D. He, R.M. Nelms, *"Peak current-mode control for a boost converter using an 8-bit microcontroller"*, in IEEE 34th Annual Conference on Power Electronics Specialist (PESC), Acapulco, 2003, pp.938–943

[18] A. Prodic, D. Maksimovic, R.W. Erickson, *"Digital controller chip set for isolated DC power supplies"*, in 18th Annual IEEE Applied Power Electronics Conference and Exposition (APEC), Miami Beach, 2003, pp.866–872

[19] Z. Zhao, V. Smolyakov, A. Prodic, *"Continuous-Time Digital Signal Processing Based Controller for High-Frequency DC-DC Converters"*, in 22nd Annual IEEE Applied Power Electronics Conference and Exposition (APEC), Anaheim, 2007, pp.882–886

[20] Intel DE1-SoC Development and Education Board, Terasic Technologies Inc. [Online] Available:www.intel.com.

[21] P. S. Crovetti, *"All-Digital High Resolution D/A Conversion by Dyadic Digital Pulse Modulation"*, IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 64, no. 3, pp. 573-584, March 2017.

[22] M. Usmonov, P. S. Crovetti, F. Gregoretti, F. Musolino, *"Suppression of Quantization-Induced Limit Cycles in Digitally Controlled DC-DC Converters by Dyadic Digital Pulse Width Modulation"*, IEEE Energy Conversion Congress and Exposition (ECCE) 2019, pp. 2224-2231.

[23] P. S. Crovetti, M. Usmonov, F. Musolino and F. Gregoretti, *"Limit Cycle-Free Digitally Controlled DC-DC Converters based on Dyadic Digital PWM"* in IEEE Transactions on Power Electronics. (in press, available online doi: 10.1109/TPEL.2020.2978696)

[24] Intel Quartus Prime Software Suire. [Online] Available:www.intel.com