POLITECNICO DI TORINO
Repository ISTITUZIONALE

Real Time Monitoring for Model Based Design of Power Converters

Original
Real Time Monitoring for Model Based Design of Power Converters / Gregoretti, F., Usmonov, M.. - ELETTRONICO. - (2021), pp. 1-6. (2021 IEEE 22nd Workshop on Control and Modelling of Power Electronics (COMPEL) Cartagena, Colombia 2-5 Nov. 2021) [10.1109/COMPEL52922.2021.9645977].

Availability:
This version is available at: 11583/2948331 since: 2022-01-20T09:57:05Z

Publisher:
IEEE

Published
DOI:10.1109/COMPEL52922.2021.9645977

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

©2021 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

Real Time Monitoring for Model Based Design of Power Converters

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

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

***Abstract*—Controllers for power converters have moved to the digital domain and the operating frequency has greatly increased making the process of debugging, tuning and experimenting with complex features more difficult. More powerful testbenches based on easily accessible System On Chip (SOC) devices can be used. This paper presents and compares two approaches to the problem. The first is a dedicated approach in which all the components of the system have been developed separately, the second one is based on the use of an integrated commercial enviroment such as Matlab & Simulink with ad hoc adaptation for real time monitoring and controller data acquisition.**

***Index Terms*—Digital control of power converters, Monitoring and testing framework, FPGA SoCs, Integrated Design Environment, rapid prototyping, model based design, Matlab**

I. Introduction

Digital controllers in switch-mode power converters offer many advantages such as the capability to easily implement advanced control laws through software or programmable hardware enhancing the reconfigurability and allowing to use a single implementation for meeting a wide range of product requirements [1]–[8].

The general structure of a digitally controlled power converter to which we will refer in the following (Figure 1) comprises a switch-mode power converter circuit, sensors for electrical quantities and the associated analog-to-digital converters (ADC) and finally a computation hardware which must execute the tasks of control algorithms, that may involve digital signal processing, filtering, and sometimes system identification [9], in discrete time domain. Converter switches are ultimately driven by the digital pulse width modulated (DPWM) signals whose duty cycle is determined by the computation hardware to achieve desired control behaviour. Depending on the architecture and performance requirements, different families of digital devices [10]–[17], such as microcontrollers (MCU), digital signal processors (DSP), application specific integrated circuits (ASIC), complex programmable logic devices (CPLD) or field-programmable gate arrays (FPGA), can used for the implementation.

High frequency commercial converters are usually implemented as custom chips and in the prototyping or parameter tuning phases the logic circuits may be substituted by field-programmable gate arrays (FPGA). Tuning of a controller or its characterization over a range of external and internal parameters requires a repetitive sequence of hardware changes and measurements, while debugging the values of internal parameters requires the real time acquisition of a large number of internal signals. System-On-Chip (SoC) devices, which integrate into a single chip a large FPGA and a multi-core Hard Processor Systems (HPS), offer the capability of rapid prototyping and faster debugging. These two parts of SoC devices are tightly connected to each other and able to communicate at high speed.

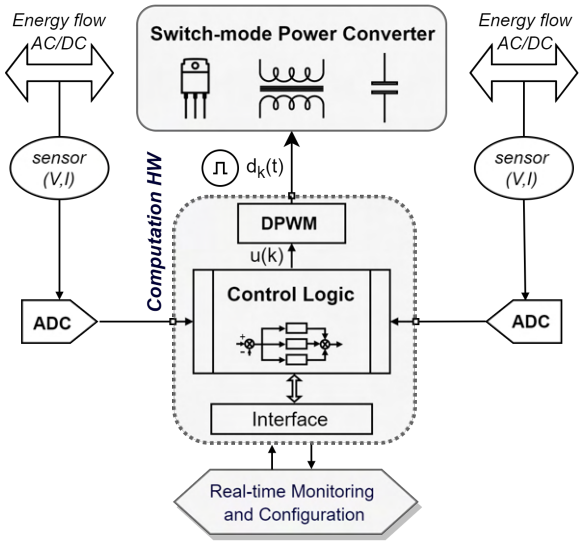


Fig. 1. Digital Control of power converter

In programming and monitoring phases, HPS may implement the task of collecting data, configuring the FPGA, and providing high level interface to the user, where the FPGA fabric executes the main controller logic, as shown in Fig. 2.

However, the development of SoC based control and monitoring system requires knowledge from multiple discipline areas, such as digital control, VHDL, C/C++ programming languages and etc. [22] (Fig. 4), which leads to more time being spent on implementation and verification phases, rather than the design itself.

Alternatively, model based approach using Matlab & Simulink can achieve the same objective with automatic code

978-1-6654-3635-9/21/$31.00 ©2021 European Union

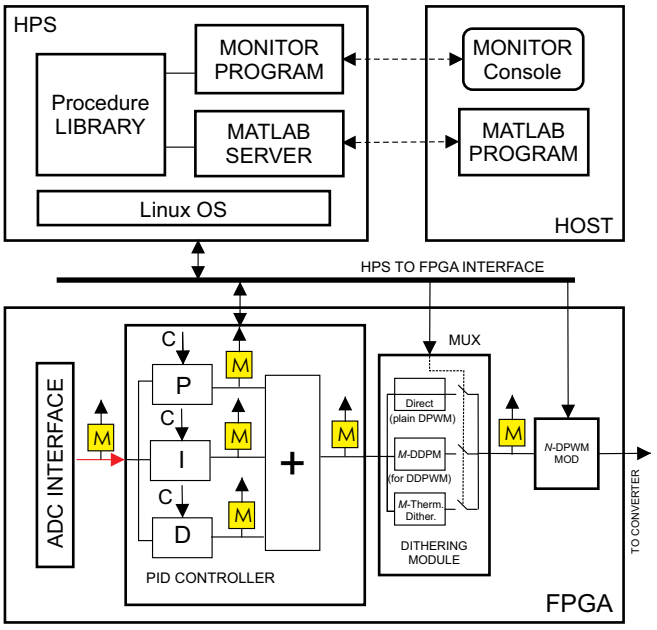


Fig. 2. Basic internal modules of SoC-FPGA pakcage (e.g. Intel Cyclone V) used to implement digital controller and real-time monitoring. FPGA and HPS fabric communicate through built-in AXI (Advanced eXtensible Interface) bridge. FPGA section runs digital controller logic, while HPS performs real-time configuration and monitoring/recording. ADC interface logic controls external ADC chip.

generation for SoC-FPGA devices, enabling faster prototyping and easier verification [16] at higher abstraction level.

This paper presents our experience in building experimental environments for the operation of switching converters as an extension to the experiment carried out in the previous work [22]. First, we will synthetically present a project dedicated to the analysis of different modulation types of the PWM signal driving the converter, which has been developed with generic development workflow of SoC devices for digital controller design and real-time monitoring of power converters. Secondly we will present the with fully based on the integration of commercially available MATLAB tools developed to overcome the main limits of the first approach and with a wider scope.

II. Dedicated Environment

The experiment had a target of analysing the influence of PWM modulation techniques on the converter output noise due to limit cycle [21]. The monitoring environment was developed to reduce the effort of collecting and organizing the large amout of data generated by varying input, output and internal controller parameters. We have used a ”DE1-SOC” protyping board from Intel which contains an ADC module, set of high level interfaces for connecting to a host computer system, and a ”Cyclone V” SoC chip with a large FPGA array and a dual core ARM-based HPS. The converter is a synchronous NMOS buck converter driven by the PWM signal from the controller operating in the range of 100 kHz to 200 kHz switching frequencies.

Figure 2 shows the main components, that are the SOC board and the host processor, and the functional architecture of

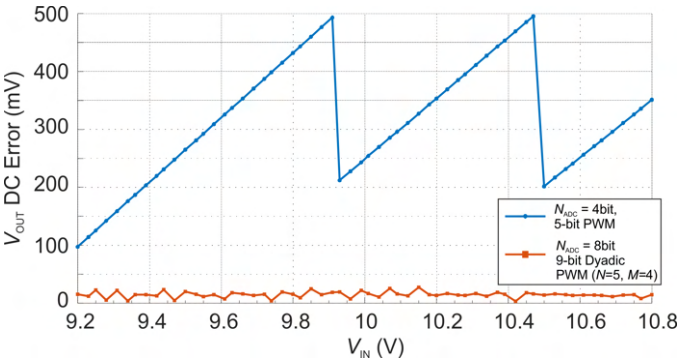


Fig. 3. Comparison of output voltage DC errors at different Vin voltage levels, tested with direct PWM and dithered PWM, and experimentally verified using the framework described in [22]. Vin=10V, fsw=100kHz, Vout=5V, digitally controlled synchronous NMOS buck converter

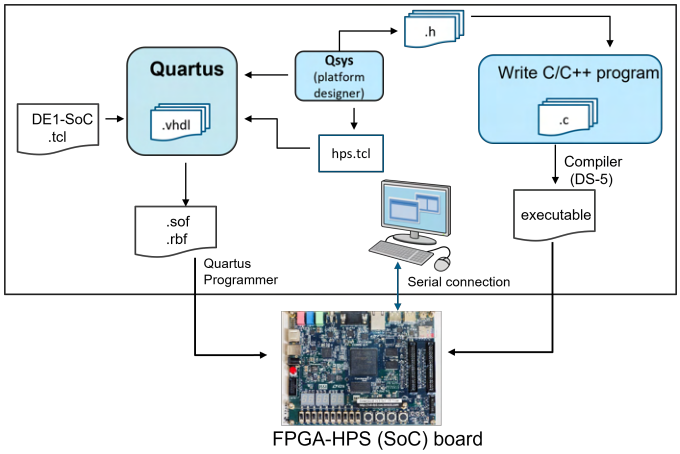


Fig. 4. Generic design flow for SoC board

the system. External components were a programmable input power supply and a programmable load.

The FPGA section implements the controller which has been manually designed using parametric blocks. Parameters which may programmed are shown in figure 2 by the letter ”C”. The significant variables of the controller are sampled and stored into memories indicated by ”M”. Control and memory registers may be written or read by the HPS section through an internal high speed interconnect. The HPS, using a library of procedures called by a MATLAB® server process running under the Linux operating system, receives commands from a MATLAB script program running on the host processor which serves also as display and monitoring unit. First the controller parameters are set, data from the converter in operation are collected and finally the results are computed and displayed. The script may also set and acquire data from components outside the converter such as the power supply and the programmable load . As an example, figure 3 represents the combined result that has been obtained by many testing sequences executed automatically by a Matlab script, and shows the output voltage error as a function of the input voltage using either a direct PWM or a more complex dyadic [21] modulation. The approach, extensively described

in [22], has proven to be very effective in performing complex analyses on a buck converter investigating non linear, limit cycle related noise and transient behaviour.

Limits arise from the fact that the complete system is based on the use of three different and independent design flows, namely the HDL tool for the design of the controller, the programming of the MATLAB server and its library and finally the experiment script itself. For each parameter to be set or variable to be monitored, it is necessary to define and instantiate it in the VHDL code, define it at the FPGA/HPS interface and at the variable level in the server program and finally to define it and use at the MATLAB script level. These definitions are related to three different tools, must be consistent and have to be manually checked, requiring multiple code changes for each modification in the controller structure, which in turn is a time consuming and error prone effort (Fig. 4).

III. Using an integrated environment

A second framework was developed using an existing integrated environments, namely a MATLAB/Simulink HDL coder toolbox automating the hardware-software co-design workflow for SoC based systems. The same DE1-SoC board was used and an high-speed GaN based buck converter (shown in Fig. 8) was developed in order to push the testing to higher frequencies. Some limitations of the Simulink toolbox due to high-speed operation of the controller were overcome by an extension of the toolbox.

A. HDL coder: features and limitations

The HDL coder package contains the blocks that can be inserted inside Simulink to design hardware compatible models. The main advantages of this tool are that the simulation and hardware models are compatible and that the automatic VHDL code generation for FPGAs is more reliable with respect to manual coding. In addition, it allows to implement some hardware algorithms directly in MATLAB script and maps it precisely to the VHDL code. and supports the SoC based hardware with automatic configuration of HPS-FPGA interconnect bridges (e.g. AXI4 master-slave). Finally it generates the SoC software interface model/script for communicating with HPS section in order to observe FPGA signals of interest in real-time.

The general workflow of SoC based hardware design using the integrated environment with the HDL coder toolbox is shown in figure 5. The required hardware model (Fig. 6) is designed in Simulink, using the HDL coder blocks. In our case, the target hardware model is the controller for the GaN based buck converter. The model can be simulated to verify the operation and may need additional simulink blocks to simulate external signals such as signals from the ADCs.

Then the HDL workflow advisor sets the options of the Simulink hardware model such as the choice of the target hardware platform supported by the toolbox and some auxiliary configurations. The HDL workflow advisor will take care of creating SoC project, generating VHDL codes for FPGA part,

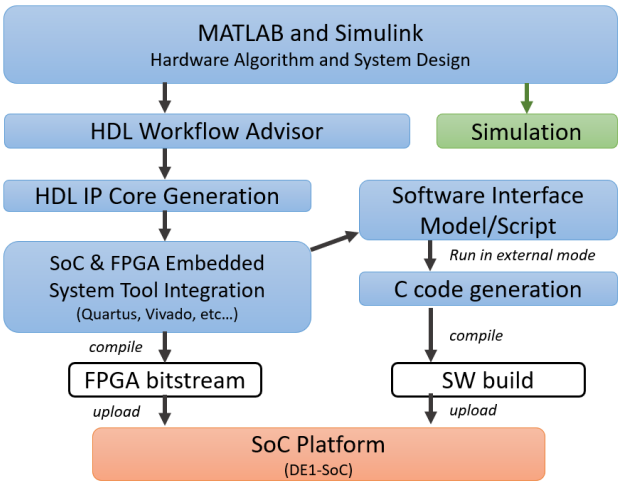


Fig. 5. Standard SoC HDL workflow of MATLAB & Simulink

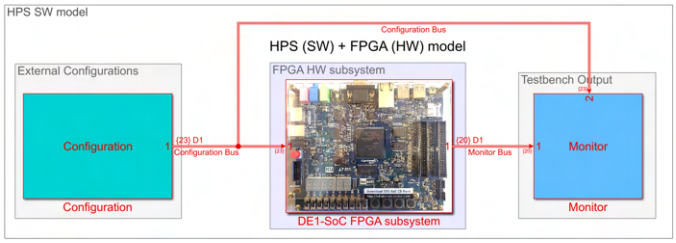


Fig. 6. Model based design of digital controller in Simulink, compatible for SoC-FPGA code generation

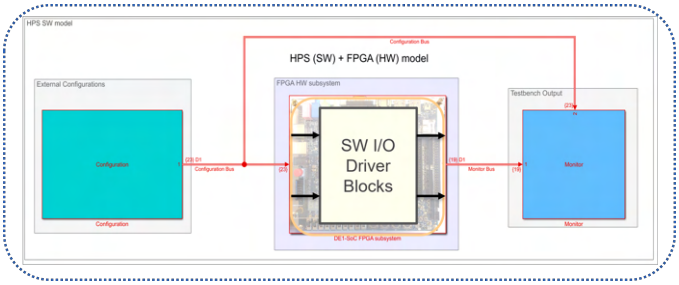


Fig. 7. Generated Software Interface Model to be used by Embedded Coder toolbox and run in external mode by Simulink

the software interface model/script (Fig. 7) for HPS section, the interconnect bridge configuration, and finally compiling and uploading the FPGA bitstream, in one uninterrupted automatic flow.

Designing the controller in Simulink enables to verify its behaviour by simulating before actually deploying it into the hardware. After the hardware deployment, it also allows to monitor the FPGA signals in real-time by running the software interface model/script generated by the HDL coder toolbox. In that process, Simulink first generates the HPS compatible C code, then compiles, uploads and runs it on HPS part of the board. All the outgoing and incoming data between the interface model running in Simulink and the SoC board are exchanged by the defined communication link that are usually Ethernet and/or serial ports.

A major limitation has been encountered when the auto-

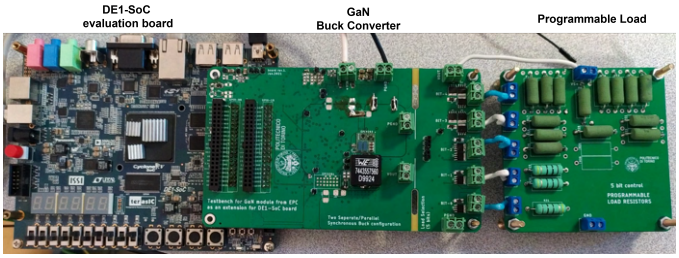


Fig. 8. Testbench setup for testing digital controller loop of GaN-based buck converter using model based approach

generated software interface model (Fig. 7) is run in Simulink for real-time monitoring of FPGA signals. Since the control of switch-mode power converter operates at high-speed, corresponding FPGA signals are updated at least every converter switching cycle. This fast updated FPGA signals can not be fully captured by the interface model running in simulink, due to different time-steps and speeds of the system components. In this experiment, the converter board operates at 300kHz - 800kHz. But, the most reliable Simulink time-step for the interface model is found to be only 1ms (or 1kHz). Therefore when observing the signals with this time-step, most samples are lost. To overcome this limit, we have designed an extension to the framework to maintain real time monitoring and at the same time taking the full advantage of HDL coder toolbox.

B. Ad-hoc extension as a replacement of software interface model

In order to avoid the speed related data loss, the standard HDL coder workflow has been modified. First, the hardware design must begin with a golden reference simulink model, that incorporates predefined FIFO blocks with dual ports to synchronize the HPS and FPGA in a flexible manner, as depicted in figure 9. The designer must insert a FIFO block for each monitored FPGA signal inside the simulink model. After this step the HDL workflow advisor operations remains the same.

After the HDL workflow advisor completes its job of creating, generating, compiling and uploading the model into the FPGA, a separate ad-hoc automation script must be run manually in the MATLAB working folder. The script takes the default software interface model/script as its input, and imitates the behaviour with more accuracy, by generating a C program for the HPS that implements interface between MATLAB and FPGA and a further MATLAB script as a template for monitoring the signals.

The general structure of this ad-hoc framework is shown in figure 10. The C code is automatically compiled and uploaded to the SoC board by the script. The main advantage of using this extension is to improve the numerical integrity of the monitoring process without sacrificing the integrated flow of the standard HDL coder toolbox. So, in the end, the designer can run the whole process without leaving the MATLAB/Simulink environment.

Digital control loop, similar to the one implemented in [22], has been redesigned in simulink model and adapted to the

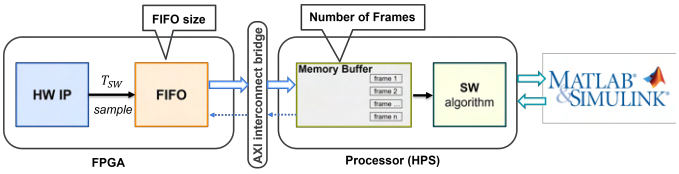


Fig. 9. FIFO queues in both FPGA and HPS section to overcome the problem of data loss due to speed differences

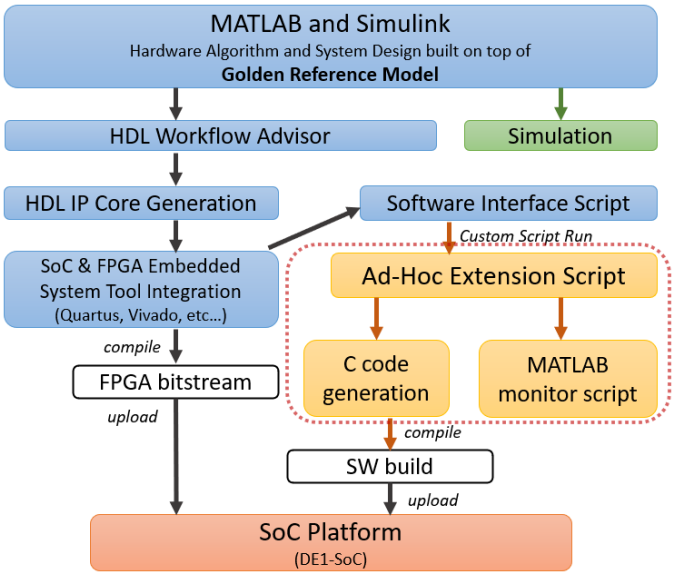


Fig. 10. Ad-Hoc framework for faster operation and better numerical integrity

GaN based board (fig. 8). Many automated experiments (fig. 11-16) have been carried out for testing the effectiveness of different PWM dithering techniques such as Dyadic and Thermometric in high-frequency GaN board, also demonstrating the integrated framework.

As shown in figures 11-14, output voltage errors will be different for the direct PWM signal with no dithering applied. Also, the direction of change in Vin-input voltage effects how DC error will behave. However, applying Dyadic based dithering, one can increase the effective PWM resolution without safrificing the system clock and stability, which can also eliminate the input voltage dependency for output voltage dc errors. This is verified both for small and large output current operations as well. Since this kind of experiments require significant amount of measurements and reconfigurations, automated testing script was applied to configure, acquire, and plot the data within MATLAB framework descibed above.

Another experiment (shown in figures 15-16) was carried out with open-loop system, in order to compare two PWM dithering techniques and their effects on the output voltage ripple of the converter, once again, carried out by scripted procedures. This latter figures approve the result obtained in [21], that shows the advantages of using Dyadic dithering modulation compared to other traditional methods, showing lower and consistent output voltage ripple both in high current and no load situations.

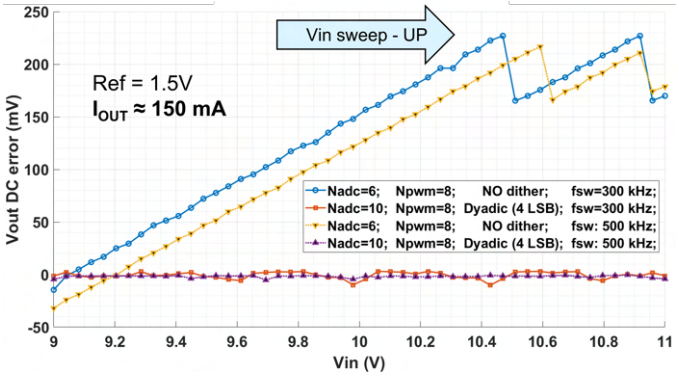


Fig. 11. DC error Vin sweep UP (150mA) - GaN based synchronous buck converter

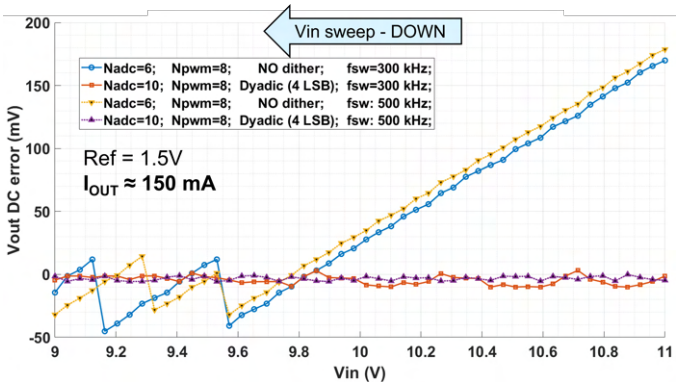


Fig. 12. DC error Vin sweep DOWN (150mA) - GaN based synchronous buck converter

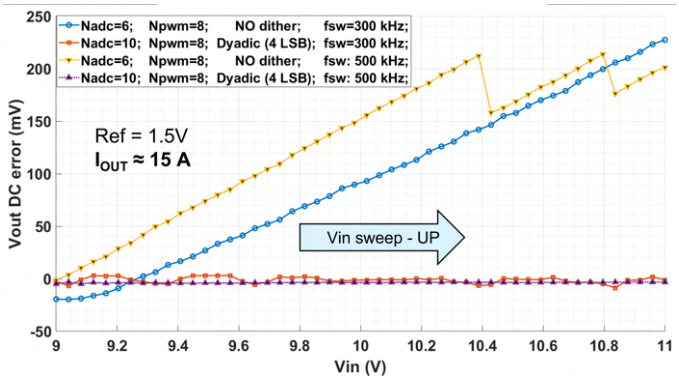


Fig. 13. DC error Vin sweep UP (15 A) - GaN based synchronous buck converter

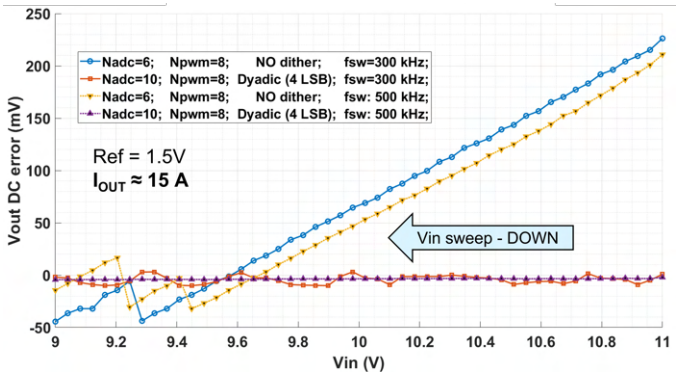


Fig. 14. DC error Vin sweep DOWN (15 A) - GaN based synchronous buck converter

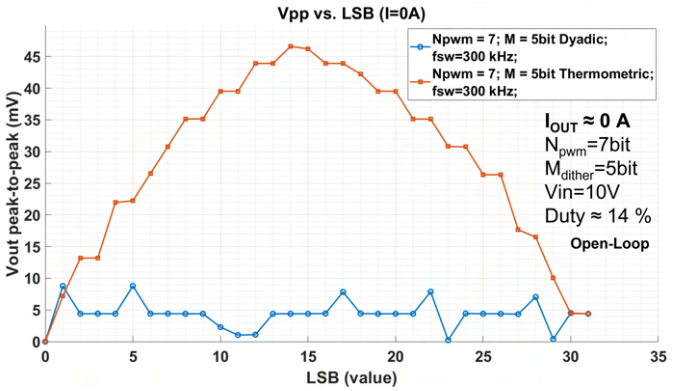


Fig. 15. Compare Dithering Methods (Dyadic vs. Thermometric; 0A - No load; Npwm=7bit; Dithering=5bit)

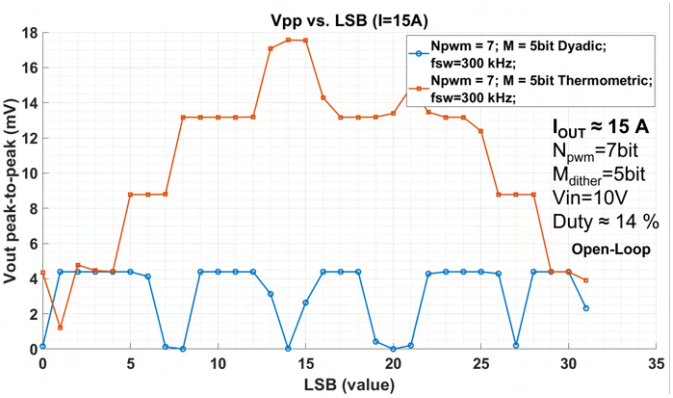


Fig. 16. Compare Dithering Methods (Dyadic vs. Thermometric; 15A load; Npwm=7bit; Dithering=5bit)

IV. Conclusions

The analysis of the paper has shown that the use of an integrated environment has the main advantage of presenting to the developer a uniform view of the system at the hardware and sotfware level from the architectural point of view and at the simulation and real time level from the operational viewpoint. In the general case the integrated approach provides a far larger flexibility and greatly reduces the development time of an experimental analysis for power converters and is a powerful tool for the designer.

However it requires the hardware platform which is used to be already integrated with the tool itself and this is true only for a limited set of them. If a new custom platform is developed, the effort of adapting it to an integrated environment may be large and may be worthwhile only if the platform itself is repetively reused. A tradeoff must be evaluated between the advantage of a general tool and the time to target for a single or limited number of applications.

A further point which requires further investigation and which has not been taken into account in this research is the effect of using an integrated environment for pushing the application to the limits of the hardware technology, that is whether it limits the optimizations which may be applied if a custom approach is taken. Typically a custom solution may better exploit technological or hardware advances, but on the

other hand the wider design exploration spectrum offered by an integrated environment may allow to find better performing archiectural alternatives.

ACKNOWLEDGMENT

The authors would like to thank the Intel© FPGA University Program for donating the DE1-SoC evaluation board and the MathWorks© Campus-Wide license Program for making available the MATLAB© and Simulink© tools.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[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] 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.

[9] C. Fernandez, M. Granda, P. Zumel, M. Sanz, A. Lazaro and A. Barrado, *"Identification system applied to the characterization of the input impedance of cascaded dc-dc converters,"* IEEE COMPEL 2018, Padua, Italy, 2018

[10] 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.

[11] 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.

[12] 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

[13] 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

[14] A. Prodic, D. Maksimovic, R.W. Erickson, *"Digital controller chip set for isolated DC power supplies"*, 18th Annual IEEE APEC, Miami Beach, 2003

[15] E. T. Mekonnen, J. Katcha and M. Parker, *"An FPGA-based digital control development method for power electronics"*, IECON 2012 - Montreal, QC, 2012.

[16] 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.

[17] 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

[18] 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.

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

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

[21] 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)

[22] P. S. Crovetti, M. Usmonov, F. Musolino and F. Gregoretti, *"A Framework for the Development and Monitoring of Digital Control in Power Converters"*, 7th International Conference on Electrical and Electronics Engineering, (ICEEE) 2020.