A New Platform and Methodology for System-Level Design of Next-Generation FPGA-based Digital SMPS

Brian MacCleery[1], Olivier Trescases[2], Muris Mujagic[1], Damon M. Bohls[1], Oleg Stepanov[1], Garret Fick[1]
[1]National Instruments, Austin, Texas, USA
[2]University of Toronto, 10 King's College Road, Toronto, ON, M5S 3G4, Canada
E-mail:brian.maccleery@ni.com

Abstract— **Increasing adoption of FPGAs in digital switched-mode power supply (SMPS) control is driving significant interest in improved platforms for development and commercial deployment. A new methodology for high-level system design of SMPS controllers is proposed and developed commercially. The feasibility, validity and accuracy of the approach is evaluated through the design and testing of a single-phase and three-phase DC-to-AC inverter. Simulation and experimental data are compared to demonstrate the high accuracy of the novel continuous time co-simulation interface. The user defined FPGA software, developed without requiring any knowledge of HDL languages, is then transferred to a COTS FPGA-chip-on-board SMPS control system for high-volume commercial deployment. The same graphical programming tools are used to develop an FPGA-based real-time SMPS hardware-in-the-loop (HIL) simulator for exhaustive validation testing of the control system hardware and software.**

I. Introduction

The increasing adoption of Field-Programmable-Gate-Array (FPGA) based controllers for high-frequency, digitally-controlled Switched Mode Power Supplies (SMPS) [1]–[4] is raising considerable interest in finding the most efficient and appropriate development tools. The FPGA embedded systems development represents a significant paradigm shift compared to that for microprocessors, micro-controllers and Digital Signal Processors (DSP), that are more conventionally used in SMPS. A typical digitally-controlled SMPS and suitable controller targets based on unit cost for different power levels are shown in Figs.1(a) and (b), respectively.

FPGAs provide an attractive architecture for power electronics control systems, due to their ability to produce custom high-frequency, low-latency (ns) gating signals, ability to place digital pulse-width modulators (DPWM) and dead-time circuits in dedicated hardware, high speed true-parallel digital signal processing capabilities, and silicon-gate-level (SGL) user configurability. Due to the fast changing requirements of the smart grid and renewable energy markets, the inherent field re-configurability of FPGAs is also attractive from the perspective of long-term support, maintenance and interoperability with evolving standards and communication protocols. As shown in Fig. 1(b), the lowest cost FPGAs on the market have now reached the $5 range in high-volume, which makes them attractive not only for proof-of-concept research prototypes but also for mass-produced converters and motor drives in the several hundred watt range and above. Below this range, full-custom Application Specific ICs (ASICs) [5] and specialized low-cost micro-controllers with dedicated high-resolution, high-frequency DPWM blocks [6] dominate the market today. Modern FPGAs are actually hybrid DSP devices utilizing hard-core DSP processing elements, integrated within the reconfigurable computing fabric to yield 1^9 to 1^{12} fixed-point multiply-accumulate operations per second (MACS), often at a lower cost per MACS and lower power dissipation per MACS compared to processors and DSPs [7]. A comparison between a Xilinx Spartan-6 LX45T FPGA and TI C6000 Series TMS320C6203B DSP with respect to performance per chip, per dollar and per watt, as measured in million multiply-accumulate operations per second (MMACS), is shown in Fig. 1(c). Whereas the C6000 series DSP contains two parallel multiplier units operating at 300 MHz, the Spartan-6 FPGA contains 58 parallel multipliers operating at 250 MHz. Based on retail low volume Digikey pricing, the massively parallel hybrid FPGA outperforms the DSP by a factor of 24×, 10× and 40×, with regards to performance per chip, per watt and per dollar respectively.

While FPGAs have been the dominant control platform for research in high-frequency, low-power SMPS in the last decade, industry development teams have been challenged to incorporate FPGAs as the main processing unit for a number of reasons. First, FPGA development remains outside the reach of many skilled power electronics designers, since it requires register-transfer-level (RTL) programming methods, such as Verilog and VHDL. The languages have a significantly steeper learning curve compared to the higher level programming tools available for DSPs and processors. Debugging, verification and long term support of register level code is also significantly more challenging than that for higher level code. The digital and analog design requirements to produce a high reliability custom printed-circuit-board (PCB) containing modern fine-pitch ball-grid-array (BGA) FPGA devices and high-speed data converters is also significantly more complex and expensive than the design of a board containing a low density chip. The adaptation of system-level graphical dataflow programming languages such as LabVIEW to FPGAs, as

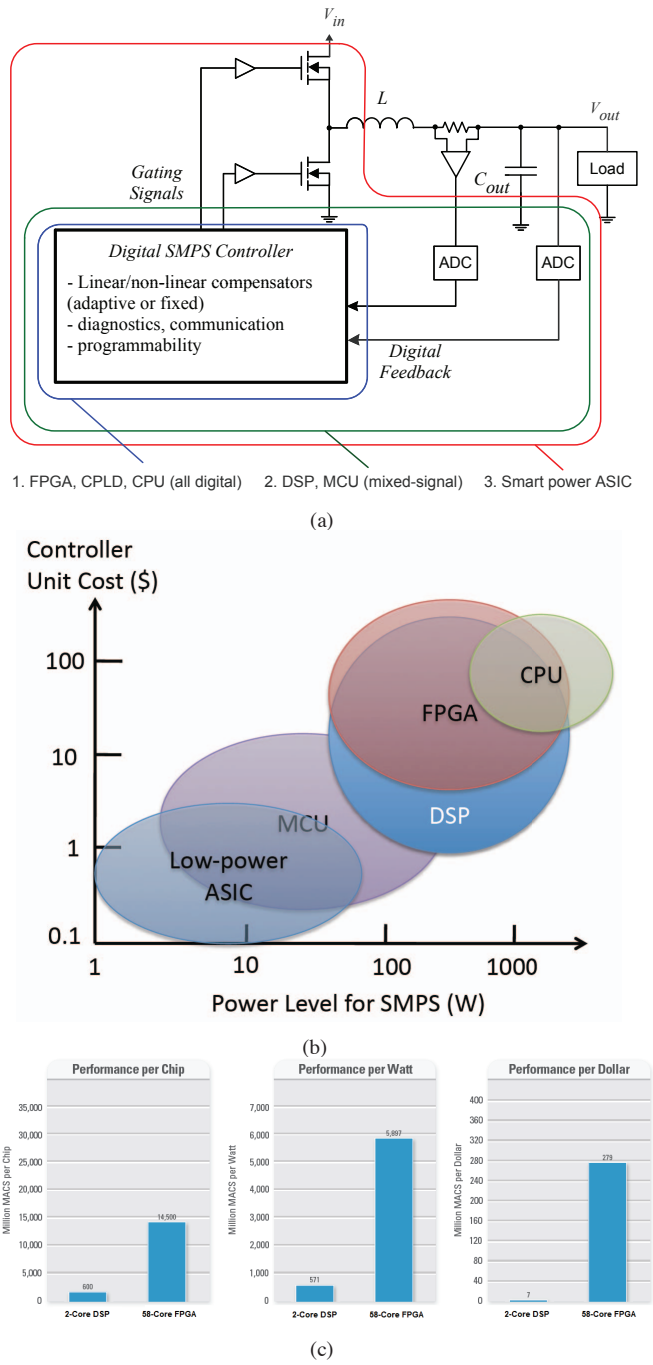


Fig. 1. (a) Generic digitally-controlled SMPS. (b) Spectrum of suitable control targets for mass production. (c) Performance comparison between Spartan-6 FPGA containing 58 DSP cores and conventional dual-core DSP.

described in this work, and the emergence of commercial off-the-shelf (COTS) hardware targets optimized for high-volume commercial deployment provides an attractive alternative to fully custom hardware development. COTS control boards containing user programmable FPGAs and power electronics specific I/O on a pre-validated PCB are intended to eliminate significant cost and risk associated with full custom FPGA board design for high-volume commercial deployment.

II. Requirements for System-Level Methodology and Hardware Platform

This section examines the application-specific needs of SMPS designers and researchers in regards to system-level FPGA programming tools and introduces a new platform developed through significant commercial R&D effort in consultation with researchers and commercial design engineers, with the goal of significantly reducing the cost and risk of power electronics design. **Key Requirements:**

1) The system-level FPGA design tool must integrate with a full-featured power electronics circuit simulator in a way that provides behaviorally and temporally accurate co-simulation to capture the high-speed, coupled dynamic interaction between the FPGA and the SMPS. The simulation platform must also have the capability to model power device parasitic elements, thermal effects and other non-idealities as required to discover design flaws before target deployment.
2) The FPGA code used in the final target must be exactly identical to the code used for design validation simulations. While this seems obvious, most designs today use different sets of code in the system-level simulation and the final hardware target, thereby invaliding simulation results.
3) The software development flow should be bi-directional rather than unidirectional. Changes made to FPGA code at any stage from prototype to post-production should automatically be reflected wherever the synthesized code is referenced in the tool-chain, thereby enabling automated testing.
4) The FPGA resource utilization efficiency must be comparable to hand written register transfer level (RTL) code, thereby eliminating the need to hand tweak and thereby "contaminated" the generated code.
5) The tool must include fixed-point math blocks and power electronics IP libraries that enable efficient development of fixed-point control, signal processing and power analysis algorithms.
6) The tool must target pre-validated commercial off the shelf (COTS) control boards that meet the specific control, I/O and performance and cost needs of modern high-volume commercial SMPS products.
7) The tool should also be suitable for developing fast, real-time hardware-in-the-loop (HIL) SMPS simulators for the purpose of enabling comprehensive validation of the production control.

The primary research focus of this paper is the solution and validation via simulation and experiment of the proposed design methodology and platform described above for single and three-phase inverters for grid-tied renewable energy systems.

III. Comparison of Proposed Design Flow to Conventional "Model Based Design"

A principal goal in developing the new platform and method is to reduce the cost, risk and development time necessary to bring commercial grid tied SMPS products to market in the clean energy, smart grid and electric transportation industries through improvements in platform-based design methods. Platform-based design tools and techniques have been an area of considerable research focus in recent years [8]–[12] due to the increasing complexity of embedded computing devices, driven by Moore's Law effects, and business pressures to accelerate time to market while simultaneously improving the reliability, energy efficiency and performance of the design.

Fig. 2 clarifies the differences in design flow between the traditional methodology and the proposed methodology, known as "graphical system design". The salient feature of the proposed approach is that high-level graphical implementation code for the FPGA, rather than a high-level software model, is developed in the simulation context and then automatically synthesized to FPGA hardware. By automatically synthesizing the implementation code inclusive of the interfaces to analog and digital I/O, the need for any knowledge of register-level FPGA programming languages, such as VHDL and Verilog, is eliminated. The approach must also enable bidirectional, team oriented development by ensuring that the graphical implementation code running in the deployment context on the FPGA is always exactly identical to that in simulation. Bi-directionality implies that the same implementation code is used in both the simulation and deployment contexts, such that changes made in either context are automatically and necessarily reflected in both. In this way, a single instance of the implementation code can be edited and evolved continuously during development. Furthermore, testing of the implementation code in the simulation context can be fully automated to enable comprehensive validation and verification against formal design requirements.

By contrast, in the traditional methodology, sometimes called "model based design" the simulation environment is used to develop a model of both the SMPS plant and the control software. Then a multi-step process is used to convert the control model from the continuous time domain to discrete time, from floating point math to fixed point math, and from system-level code to automatically generated register level code. This development methodology is unidirectional, since changes made to the code in the deployment context are not automatically reflected in the simulation context. Care must be taken to ensure that the register level code is not modified in the deployment context after code generation, otherwise the software model in the simulation context will not match the register level code in the deployment context. Changes should be made only to the software model in the simulation context, otherwise closed loop simulation results are invalided. However, the integration of input/output (I/O) signals from ADCs/DACs and digital I/O chip-sets connected to the FPGA typically must be implemented using register level programming. In the case of a traditional full-custom FPGA circuit design, the majority of FPGA software engineering cost (typically 70 percent according to Xilinx) is devoted to the creation of I/O driver interfaces between the FPGA and the external devices [13].

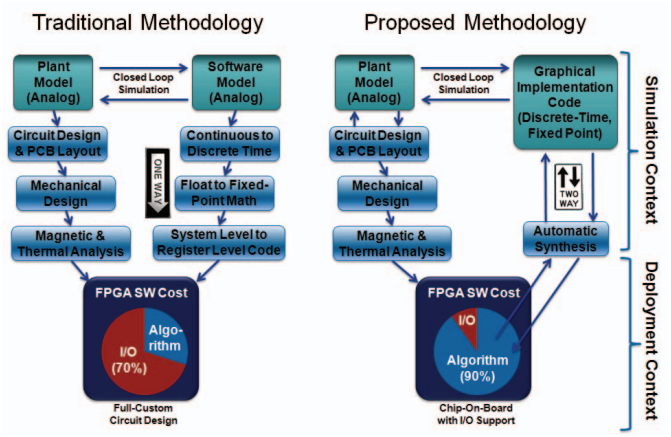


Fig. 2. Comparison of traditional system-level design methodology for FPGA software design, involving the conversion of software algorithm models to register-level code and hand coding of I/O interfaces, versus the proposed methodology in which the graphical implementation code including the I/O interface is automatically synthesized and no knowledge of register level programming is required.

To eliminate the bulk of the non-recurring engineering (NRE) expense associated with developing and testing the I/O interface logic, the proposed methodology utilizes integrated chip-on-board embedded systems suitable for high-volume deployment with pre-validated I/O and support libraries. The motivation for this is two-fold. First, pre-validated chip-on-board designs with I/O support enable the bulk of the software development expense, typically around 90%, to be focused on high value custom algorithm design and validation rather than low value I/O support. Furthermore, the I/O interface code, along with its behavior and timing characteristics, can be modeled in the simulation context, in order to accurately capture the important dynamics at the boundaries between the digital and analog domains.

The second motivation for a chip-on-board approach is that the cost and risk of printed circuit board (PCB) design and layout has been steadily increasing over the years due to Moores Law effects. Specifically, the number of I/O pins and signal speed on modern computing devices is increasing while the mechanical pitch is decreasing. While in the past, large scale chip packages such as dual in-line package (DIP) and quad flat package (QFP) were available, most modern devices are offered in fine-pitch ball grid array (BGA) packages with less than 1 mm between pins. Each pin is surrounded by up to eight adjacent pins so six or more PCB layers are typically required for successful routing of the high density, high speed signals. At the same time, the decreasing size of computing devices means that thermal heat dissipation is more locally concentrated within the package, therefore careful

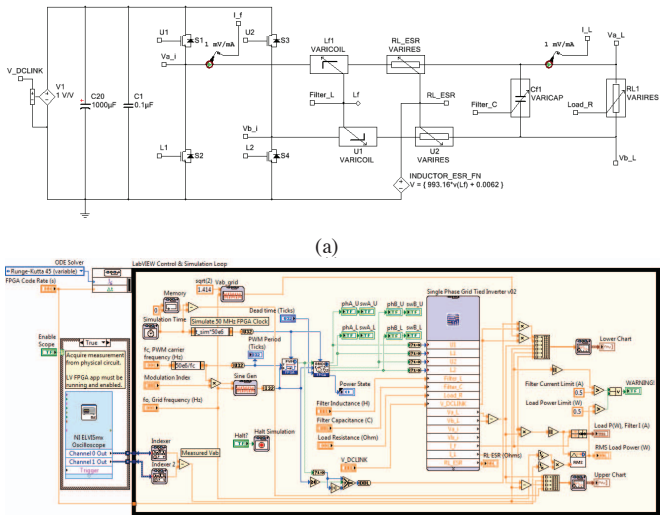


Fig. 3. Single-phase inverter (a) co-simulation schematic and (b) digital controller developed in the graphical FPGA programming language. Each square node in the schematic represents a continuous time interface between the SMPS and FPGA software.

consideration of heat transfer and cooling is required.

IV. DEMONSTRATION OF SYSTEM-LEVEL DESIGN AND SIMULATION PLATFORM

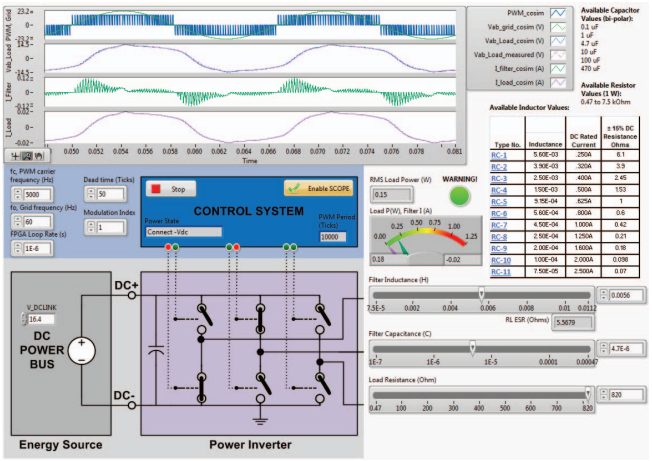
The feasibility, validity and accuracy of the new design methodology containing the desirable characteristics described in Sections II and III is evaluated through the design and testing of a single-phase and three-phase DC-to-AC inverter. An experimental inverter was assembled using a low-power off-the-shelf three-phase IGBT intelligent power module (IPM), and the graphical FPGA implementation code was validated using co-simulation techniques before downloading to the COTS FPGA board.

The power electronics circuit schematic for the single-phase H-bridge inverter is shown Fig. 3(a). Note that each labeled square node represents a variable time step (continuous-time) interface between the SMPS circuit simulation environment and the graphical dataflow FPGA programming tool. A novel patent-pending co-variable time-step co-simulation solver mechanism is used to create a continuous-time interface between the control and simulation environments. The tools automatically negotiate a mutually agreed time-step to satisfy the required error tolerances. By automatically adjusting the time-step on both sides of the co-simulation interface, the patent pending variable time-step environment creates a continuous-time co-simulation interface, able to accurately capture coupled dynamics interactions that span between the environments. Take the coupled differential equations for a brushed DC motor as an example. With a continuous-time co-simulation interface, accurate results can be achieved with the mechanical equation solved on one side while the electrical equation solved on the other, by automatically adjusting the time-step as necessary. By contrast, a fixed time-step solver may give incorrect results during fast transient events such as a reversal of the H-bridge during full speed operation of the motor, unless a sufficiently small time-step is used for the entire simulation run. The graphical programming software associated with the FPGA-based inverter control application is shown in Fig. 3(b). The precise timing and cycle-by-cycle execution behavior of the FPGA blocks is accurately modeled together with the transient interaction with the power electronics circuit dynamics.

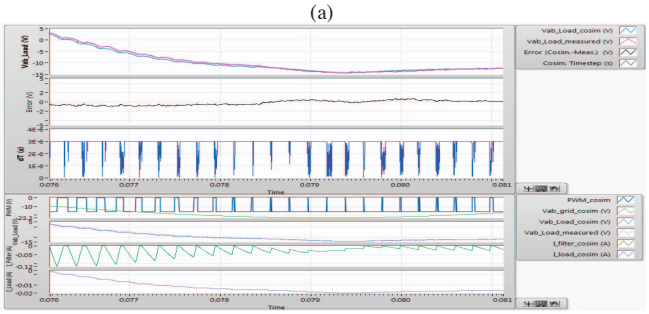
The software interface shown in Fig. 4(a) shows the results of a co-simulation run in which experimental voltage measurements are also acquired from an oscilloscope. The line-to-line voltage, Vab_{Load}, is charted on the same plot as the inverter simulation results. Due to the fidelity of the novel continuous-time co-simulation interface, the signals match closely. The variable co-simulation time-step, $dT(s)$, is automatically adjusted by the solvers to satisfy accuracy constraints, as shown explicitly in the simulation result.

Note that the exact same graphical FPGA control blocks (PWM and BRIDGE Controller) of Fig. 3(b) are used in both the desktop co-simulation and physical deployment target, Fig. 5(a). On the graphical front panel of the running FPGA application, Fig. 4(a), the loop rate is set such that the execution timing (and behavior) of the control algorithm in the physical FPGA is an exact match to the behavior of the same FPGA logic executing in the Windows co-simulation application. The experimental setup, Fig. 6(b), uses a ST Micro STGIPS10K60A 10 A, 3-phase inverter Intelligent Power Module (IPM).

For high-volume commercial deployment, the identical FPGA software, developed without requiring any knowledge of HDL languages, is then transferred to a pre-validated COTS General Purpose Inverter Control (GPIC) system, as shown in Fig. 5(a), containing Xilinx Spartan-6 LX45T FPGA. A block diagram of the GPIC hardware/software architecture is shown in Fig. 5(b). The GPIC I/O card provides a standard set of analog and digital I/O to meet the needs of a majority of grid-tied SMPS applications. This includes wind turbine converters, grid tied solar inverters, uninterruptible power supplies (UPS), industrial DC-to-DC converters, multilevel industrial motor drive/pump converters, commercial electric vehicles, electric-hybrid vehicles, and railway power bus systems. The GPIC I/O configuration is also designed for compatibility with the requirements of utility Flexible AC Transmission Systems (FACTS) used to control the power flow and enhance stability in transmission and distribution, including distributed energy storage systems (DESSs), high voltage DC (HVDC) transmission, unified power flow controllers (UPFCs), dynamic voltage restorers (DVR), static synchronous compensators (STATCOMs), Static VAR Compensator (SVC), solid state breakers (SSBs), solid-state transfer switches (SSTS), and more. The ability of a single general purpose control board to adapt to such a diverse range of application specific requirements can be attributed to the reconfigurable nature FPGA-based control systems, which provide silicon gate level (SGL) reconfigurability, along with the parallel digital signal

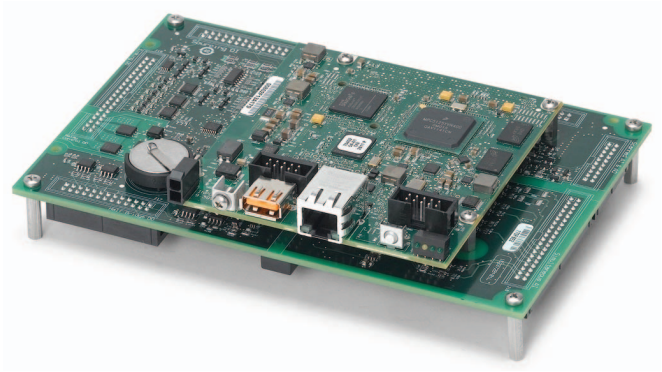


(a)

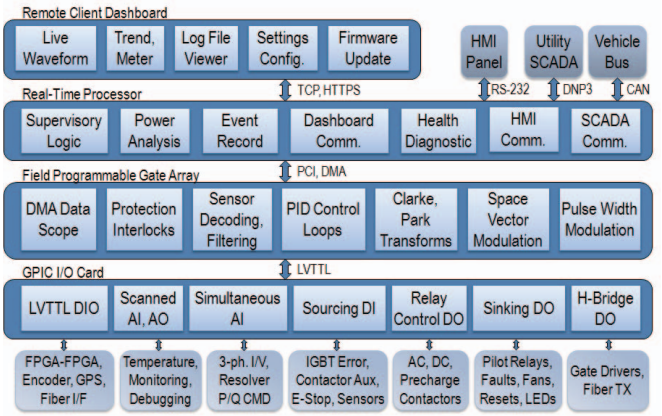


(b)

Fig. 4. (a) Co-simulation results plotted with experimental measurements overlaid on chart Vab_{Load} (V) and (b) 5 ms detail of Vab_{Load} (V) desktop co-simulation and overlaid experimental measurements for the single-phase inverter of Fig. 6(b). The co-simulation time-step, $dT(s)$, is automatically controlled by both solvers to accurately capture transient dynamics caused by switching events and disturbances.



(a)



(b)

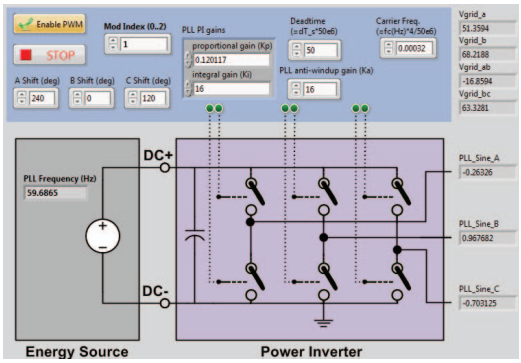
Fig. 5. (a) FPGA chip-on board general purpose inverter control deployment board and (b) block diagram of the GPIC hardware/software architecture.

processing performance of the 58 DSP cores integrated within the FPGA fabric and the selection of front end analog/digital I/O for SMPS applications.

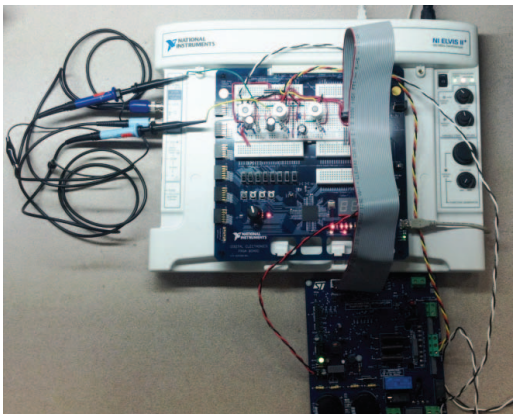
V. EMBEDDED SYSTEM VALIDATION USING FPGA-BASED REAL-TIME SIMULATION OF SMPS

To enable comprehensive and exhaustive validation using an FPGA-based hardware-in-the-loop (HIL) simulator, the same graphical system design environment is used to develop a high fidelity FPGA-based 9x9 state-space matrix solver for signal-level only or full power real-time simulation of SMPS. FPGAs are attractive for real-time simulation of SMPS since bus latency typically limits processor-based simulations to around 100 kHz, yielding approximately 5 percent error for an SMPS with a 2 kHz PWM switching frequency or 9 percent error for an 8 kHz switching frequency converter, unless special and solvers and hardware counters are used to capture inter-simulation time-step switching (ITS) events [14]. On the other hand, real-time simulation speeds in the multi-MHz range is readily achievable using FPGAs due to their true hardware parallelism and fast input-to-DSP-to-output speeds [15]–[19].

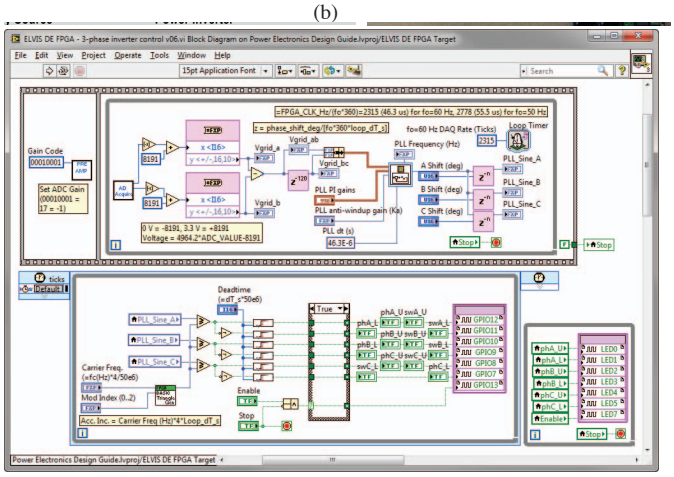
A downloadable reference design for the 3-phase inverter HIL system [20] shown in Fig. 8(a), uses 93 out of 180 of the DSP48A1 cores on a Xilinx Spartan-6 LX150 FPGA and executes at rates up to 3.57 MHz. Virtually any SMPS topology can be converted to an equivalent piecewise state-space model for real-time simulation, in which each variant of the piecewise model is associated with a particular switching state of the SMPS, depending on the conduction state of the transistors and diodes within the circuit. The DSP48A1 cores offer 18-bit resolution for multiplication and 48-bit resolution for accumulation. Since this provides sufficient accuracy for most simulations, a standard fixed-point word length can be used and the floating to fixed point conversion process is reduced to simple algebraic scaling. A per-unit representation is typically used to reduce the dynamic range of internal values, and the fixed-point simulation is compared to a circuit simulation of the same SMPS to validate the scaling choices before the simulation model is compiled to the FPGA, as shown in Fig. 8(b). The ability to simulate the SMPS in real-time under the control of the production control system enables risk free performance of tests that would otherwise be physically destructive or impossible due to bandwidth limitations of physical test systems.



(a)

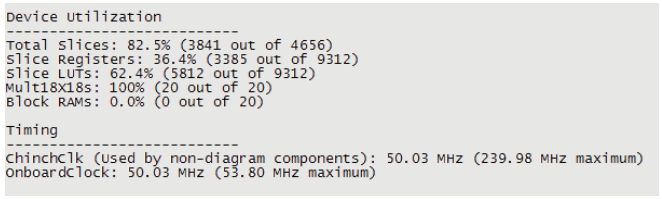


(b)

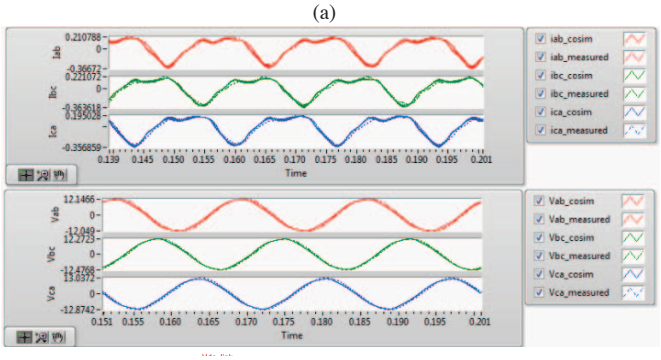


(c)

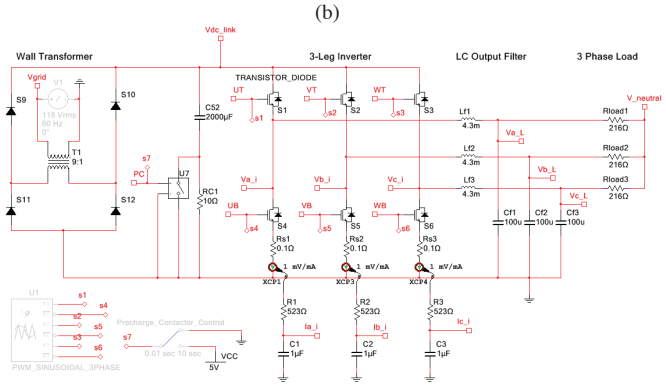
Fig. 6. (a) Three-phase inverter FPGA control application front panel GUI, (b) experimental platform for the inverter and (c) graphical FPGA programming software.



(a)



(b)



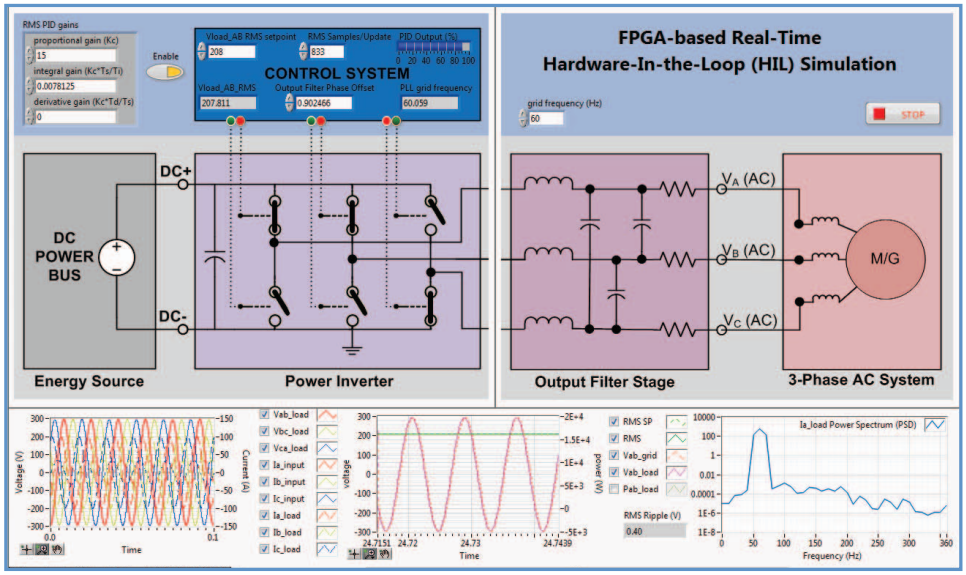
(c)

Fig. 7. (a) Three-phase inverter FPGA compile report showing efficient resource utilization, (b) three-phase experimental measurements (solid) and co-simulation results (dashed) overlaid on the same chart, and (c) Multisim simulation schematic used for co-simulation.

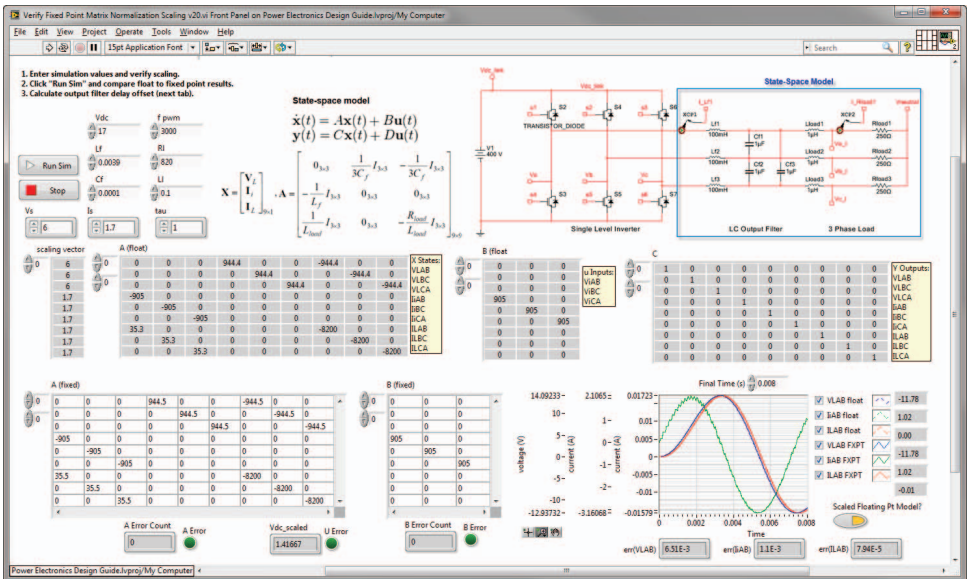
VI. Conclusions

We have demonstrated a new system-level design platform and methodology for FPGA-based digitally controlled SMPS design that satisfies the seven proposed requirements defined in Section II and the bi-directional FPGA co-simulation design flow defined in Section III. The salient features of the new methodology include: (1) A continuous time co-variable time-step co-simulation interface for accurate simulation of digitally controlled SMPS, (2) a bidirectional design flow for FPGA software simulation and deployment, which enables formal validation and verification of a single instance of the graphical FPGA implementation code, and (3) a standardized general purpose FPGA-chip-on-board system with full I/O driver support intended to reduce the cost and risk of developing grid tied SMPS control applications.

This new methodology allows designers to seamlessly leverage the computational advantages of next generation FPGAs containing integrated DSP cores for digitally controlled power electronic converters without requiring any knowledge of HDL languages. With the ability to target COTS SMPS control boards, the novel controllers can be directly transferred from prototype to high-volume commercial products. In addition, the control system hardware and software can be exhaustively validated using real-time HIL simulation for comprehensive and formal verification.



(a)



(b)

Fig. 8. (a) Real-time FPGA-based 3-phase HIL inverter simulator, and (b) application used to perform and verify the float-point to fixed-point scaling of the state-space matrix coefficients.

REFERENCES

[1] E. Monmasson and M. Cirstea, "Fpga design methodology for industrial control systems;a review," *Industrial Electronics, IEEE Transactions on*, vol. 54, no. 4, pp. 1824 –1842, aug. 2007.

[2] L. Corradini, P. Mattavelli, E. Tedeschi, and D. Trevisan, "High-bandwidth multisampled digitally controlled dc;dc converters using ripple compensation," *Industrial Electronics, IEEE Transactions on*, vol. 55, no. 4, pp. 1501 –1508, april 2008.

[3] J. Morroni, R. Zane, and D. Maksimovic, "Design and implementation of an adaptive tuning system based on desired phase margin for digitally controlled dc;dc converters," *Power Electronics, IEEE Transactions on*, vol. 24, no. 2, pp. 559 –564, feb. 2009.

[4] D. Maksimovic, R. Zane, and R. Erickson, "Impact of digital control in power electronics," in *Power Semiconductor Devices and ICs, 2004. Proceedings. ISPSD '04. The 16th International Symposium on*, may 2004, pp. 13 – 22.

[5] O. Trescases, N. Rahman, A. Prodic, and W. T. Ng, "A 1v buck converter ic with hybrid current-mode control and a charge-pump dac," in *Power Electronics Specialists Conference, 2008. PESC 2008. IEEE*, june 2008, pp. 1122 –1128.

[6] "Intelligent Power Supply Design Center," Microchip Technology, 2012, available http://www.microchip.com/en-US/technology/intelligentpower/.

[7] "Spartan-6 FPGAs in Video Designs," Electronic Engineering Journal, 2012, available http://www.eejournal.com/archives/on-demand/2011083101-xilinx.

[8] A. Sangiovanni-Vincentelli and G. Martin, "Platform-based design and software design methodology for embedded systems," *Design Test of Computers, IEEE*, vol. 18, no. 6, pp. 23 –33, nov/dec 2001.

[9] K. Balasubramanian, A. Gokhale, G. Karsai, J. Sztipanovits, and S. Neema, "Developing applications using model-driven design environments," *Computer*, vol. 39, no. 2, pp. 33 – 40, feb. 2006.

[10] C. Brooks, C. Cheng, T. H. Feng, E. A. Lee, and R. von Hanxleden, "Model engineering using multimodeling," in *1st*

International Workshop on Model Co-Evolution and Consistency Management (MCCM '08), September 2008. [Online]. Available: http://chess.eecs.berkeley.edu/pubs/486.html

[11] G. Karsai, J. Sztipanovits, A. Ledeczi, and T. Bapty, "Model-integrated development of embedded software," *Proceedings of the IEEE*, vol. 91, no. 1, pp. 145 – 164, jan 2003.

[12] K. Keutzer, A. Newton, J. Rabaey, and A. Sangiovanni-Vincentelli, "System-level design: orthogonalization of concerns and platform-based design," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 19, no. 12, pp. 1523 –1543, dec 2000.

[13] B. Chown and M. Lang, "Modernizing system development: Requirements-based, model-driven design, implementation and test," in *ERTS Congress 2012, Embedded Real Time Software and Systems*, feb 2012.

[14] M. Matar and R. Iravani, "Fpga implementation of the power electronic converter model for real-time simulation of electromagnetic transients," *Power Delivery, IEEE Transactions on*, vol. 25, no. 2, pp. 852 –860, april 2010.

[15] A. Myaing and V. Dinavahi, "Fpga-based real-time emulation of power electronic systems with detailed representation of device characteristics," *Industrial Electronics, IEEE Transactions on*, vol. 58, no. 1, pp. 358 – 368, jan. 2011.

[16] M. Cirstea and A. Dinu, "A vhdl holistic modeling approach and fpga implementation of a digital sensorless induction motor control scheme," *Industrial Electronics, IEEE Transactions on*, vol. 54, no. 4, pp. 1853 –1864, aug. 2007.

[17] G. Parma and V. Dinavahi, "Real-time digital hardware simulation of power electronics and drives," *Power Delivery, IEEE Transactions on*, vol. 22, no. 2, pp. 1235 –1246, april 2007.

[18] H. Li, M. Steurer, K. Shi, S. Woodruff, and D. Zhang, "Development of a unified design, test, and research platform for wind energy systems based on hardware-in-the-loop real-time simulation," *Industrial Electronics, IEEE Transactions on*, vol. 53, no. 4, pp. 1144 –1151, june 2006.

[19] A. Saran, S. Palla, A. Srivastava, and N. Schulz, "Real time power system simulation using rtds and ni pxi," in *Power Symposium, 2008. NAPS '08. 40th North American*, sept. 2008, pp. 1 –6.

[20] "Power Electronics Development Center," National Instruments, 2012, available http://www.ni.com/powerdev.