High-Level Synthesis for Accelerating the FPGA Implementation of Computationally Demanding Control Algorithms for Power Converters

Denis Navarro, Óscar Lucía, *Member, IEEE*, Luis A. Barragán, Isidro Urriza, and Óscar Jiménez, *Student Member, IEEE*

***Abstract*—Recent advances in power electronic converters highly rely on the development of new control algorithms. These implementations often require complex control architectures featuring microprocessors, digital signal processors, and field-programmable gate arrays (FPGAs). Whereas software implementations are feasible for most power electronics practitioners, FPGA implementations with ad-hoc digital hardware are often a challenging design task. This paper deals with the design and development of control systems for power converters using high-level synthesis tools. In particular, the Xilinx Vivado HLS tool is evaluated for the design of a computationally demanding application, the real-time load estimation for resonant power converters using parametric identification methods. The proposed methodology allows the designer to use a high-level description language, e.g., C, to describe the identification algorithm functionality, and the tool automatically generates the hardware floating-point data-path and the control unit. Besides, it allows a fast design-space exploration through synthesis directives, and pipelining and parallelization are automatically performed to meet timing constraints. The evaluation performed in the study-case control architecture shows a significant design complexity reduction. As a consequence, high-level synthesis tools should be considered as a new paradigm in accelerating digital design for power conversion systems.**

***Index Terms*—Digital control, field-programmable gate arrays (FPGAs), high-level synthesis, power conversion.**

NOMENCLATURE

$R_{\rm eq}$	Equivalent resistance of the induction system.
$L_{\rm eq}$	Equivalent inductance of the induction system.
$C_{\rm res}$	Resonant capacitor.
$v_{\rm bus}$	Bus voltage.
v_o	Inverter output voltage.
i_L	Inverter output current.
$f_{\rm sw}$	Switching frequency.
f_s	Sampling frequency.
T_s	Sampling period.
T_c	Computing period.
P_o	Output power.

Manuscript received March 30, 2012; revised July 25, 2012, September 12, 2012, November 16, 2012; accepted January 07, 2013. Date of publication January 11, 2013; date of current version August 16, 2013. This work was supported in part by the Spanish MICINN under FPU Grant AP2010-5267, Project TEC2010-19207, Project CSD2009-00046, and Project IPT-2011-1158-920000, by DGA-FSE, and by the Bosch and Siemens Home Appliances Group. Paper no. TII-12-0263.

The authors are with the Department of Electronic Engineering and Communications. University of Zaragoza, 500018 Zaragoza, Spain (e-mail: olucia@unizar.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2013.2239302

I. INTRODUCTION

POWER electronics has experienced in recent years a significant technological breakthrough due to the development of enabling technologies that allow to implement power systems with improved performance, higher efficiency, and reduced cost. Among these technologies, digital control has been a keystone in the development of such systems [1], replacing analog control to obtain more reliable, efficient, and versatile power converters.

When developing digital controller, two approaches are mainly possible [2], [3]. First, microprocessors (μPs) and digital signal processors (DSPs) offer a fully software solution that can match the requirements of most applications with a complete set of development tools. Second, field-programmable gate arrays (FPGAs) [4]–[9] can be used to implement ad hoc digital hardware that can perform specific tasks not suitable to be carried out by μPs or DSPs.

Nowadays, advances in silicon technology have brought to the market FPGA technology with enough resources to implement computationally demanding control algorithms [10]–[13]. This makes FPGA technology an interesting choice for power conversion systems from the point of view of performance and cost. In spite of this, most of the industrial designs are still performed using software approaches, i.e., DSPs or μPs. One of the main reasons for this is the additional skill set required to work with hardware description languages (HDLs) to perform a register transfer level (RTL) description of the control system, which may also increase the design-cycle time and production cost. Whereas implementing a complete system using a HDL [14] is a discouraging task for regular DSP or μP user, the use of high-level programming languages is a well-known and easy-to-use tool. For this reason, high-level synthesis tools (HLSTs) [15]–[18] arise as an alternative to HDLs when using FPGAs.

HLSTs allow the user to describe the control system using a high-level language, most commonly C, in order to generate automatically the RTL description for either FPGA or ASIC implementation. These tools simplify and therefore accelerate the

design process for complex algorithms and ease the migration of some designs from DSP to FPGAs [19]. Up to now, HLSTs have been used mainly for signal processing developments [15], [20]. However, there is a big potential for using these tools for power converter control algorithm development, especially for high demanding FPGA-based controllers [11]. Among these applications, hardware-in-the-loop (HIL) implementations, advanced controls based on estimators, or real-time identification systems are examples of complex control systems with a data-path that encourages the use of HLSTs [13], [21]. Moreover, HLSTs allow a straightforward migration of floating-point algorithms to FPGA implementations.

Compared with other alternatives, HLSTs present several advantages. DSP Builder and System Generator are DSP design tools from Altera and Xilinx, respectively, that enable the use of Simulink for FPGA design [14]. They both allow implementing hardware floating-point data-paths, but the designer has to design the control logic, which is one of the most time-consuming tasks, and they are limited to the available library elements. Moreover, the simulation time is significantly higher than the behavioral simulation using C or MATLAB. Handle-C is also a C-like language that allows to implement a design into an FPGA, but it does not allow floating-point implementations and the designer has to extract the parallelism of the algorithm using specific instructions, i.e., "par" statement [22]. Another alternative is to use the VHDL-2008 fixed-point package fixed_pkg, and the VHDL-2008 floating-point package float_pkg [23]. These packages are included in the IEEE Standard VHDL LRM 1076-2008. However, these libraries do not allow to pipeline the arithmetic operators to increase speed, and the control logic must be synthesized by the designer.

Therefore, the aim of this paper is to analyze the design and development of part of a digital controller used in a resonant power converter. Vivado HLS tool (former AutoESL) is evaluated for the design of a parametric impedance identification system for a resonant power converter applied to domestic induction heating. A general procedure for digital controller design using an HLST will be proposed, and the main conclusions extracted from the design-case example will be discussed.

The remainder of this paper is organized as follows. Section II summarizes the main characteristics of HLSTs and proposes a design procedure for power converter control algorithm. Section III explains the design case used to analyze the performance of the selected HLST, detailing implementation results. Finally, the main conclusions of this paper are drawn in Section IV.

II. HIGH-LEVEL SYNTHESIS TOOLS

A. HLSTs and Digital Control of Power Converters

HLSTs and digital control of power converters have separately evolved during the last 20 years, achieving nowadays a remarkable grade of maturity.

On the one hand, HLSTs have significantly evolved since its origins in the 1980s. According to [15], three generations can be distinguished. The first generation was developed during the 1980s and was mainly a research-oriented one with no or little impact in industry [24]. The lack of a real need, obscure input languages, and a problematic performance limited the adoption of these tools. The second generation started in the mid-1990s and was fostered by the major computer-aided design (CAD) companies present in the market, i.e., Cadence, Synopsys, and Mentor Graphics. Several commercial tools were deployed, but they still lacked of good performance and reduced learning curve, making it not interesting for current designers. Finally, the third generation of HLSTs started at the beginning of the 21st century. This generation of tools currently offers an improved performance and user interface, leading to a significant reduction in the design times. In addition, most of them have adopted high-level languages commonly used by design engineers, such as MATLAB or C, reducing the learning time. Currently, most HLSTs are focused and being used for data-processing applications [15]. Among the many alternatives, Vivado HLS from Xilinx, Synphony C Compiler from Synopsys, Catapult from Mentor Graphics, and CtoS from Cadence are relevant examples of HLSTs.

First applications of digital control started in the late 1980s [25], [26], and many studies on digital control of power converters were published during the 1980s and 1990s. However, by that time, the limited digital resources failed to provide improved performance at a reasonable cost, and the application to commercial products was insignificant. Since the late 1990s, advances in digital hardware, i.e., μPs, DSPs, and FPGAs, allowed to implement high-performance and cost-effective digital controllers, contributing to a major breakthrough in power conversion technology. After a few years of discussion about the benefits and drawbacks of digital control versus analog control [27], digital control has led to higher performance controllers, with reduced cost and increased reliability. In the last years, uncountable examples of advanced digital controllers can be found in the literature [11], [28]–[32]. Many of these controllers have been taking advantage of the processing capabilities of FPGAs. This case is precisely where HLSTs can help the designer to reduce the design time by avoiding complex HDLs. The Berkeley Design Technology, Inc. (BDTi) has also reported several independent studies supporting the viability of these tools.

As a conclusion, both digital control techniques and HLSTs have evolved and are well present in the current technical scenario. Up to now, digital control algorithms have been implemented using either high-level programming languages, for μP and DSP implementations, or HDLs, for FPGA implementations. In the same manner, HLSTs have been mainly applied to data-processing tasks. Considering advances in both fields, the design of digital control algorithms for power converters by using HLSTs arises as a new paradigm for the design of digitally controlled power converters.

B. Design of Power Conversion Systems Using HLSTs

The design procedure of a power conversion system, composed of the power converter and the control architecture, is summarized in Fig. 1. In a first step, the static power converter design and the FPGA-based control architecture design can be considered as independent optimization tasks. Both of them are optimized to fulfill certain design conditions, i.e., desired specifications, by optimizing the design variables according a certain set of design constraints.

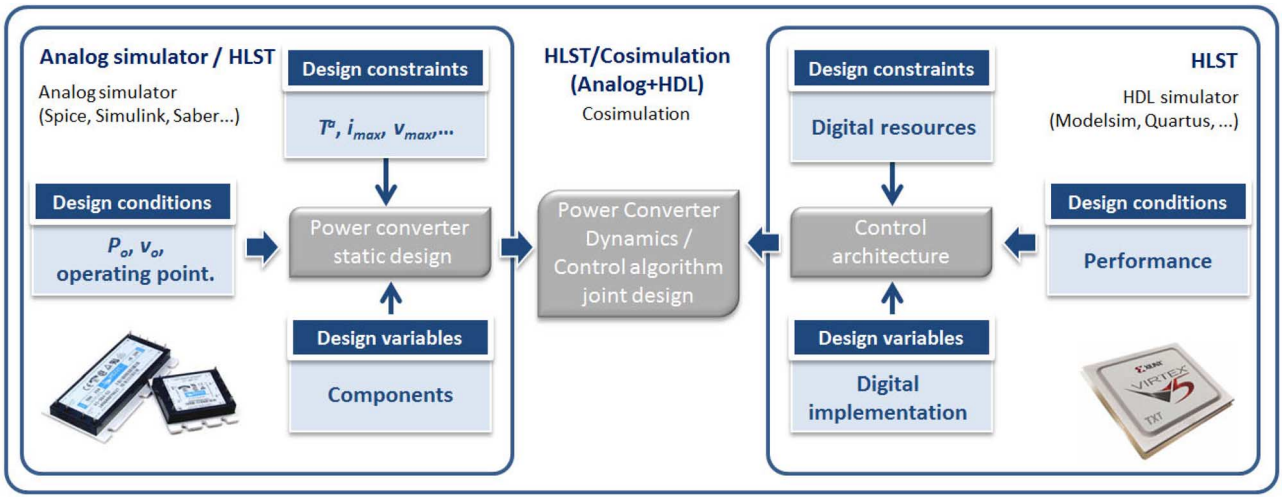

Fig. 1. Design procedure for FPGA-controlled power converters using CAD and simulation tools: proposed design procedure (blue bold) and classical approach.

First, the optimized design for the power converter [33] results from combining the design conditions, i.e., nominal operating conditions, with the design constraints, e.g., maximum temperature and voltage and/or current, and the design variables, i.e., components values. The control architecture is optimized in the same manner to meet the required performance, i.e., controller bandwidth (BW), margin, and phase gains (MG, PG), according to the constraints imposed by the available digital resources.

In a classical design procedure, the power converter is optimized using analog simulators, i.e., Spice or Simulink, whereas the control architecture is described using a HDL, i.e., VHDL or Verilog, and simulated in an HDL simulation tool as Modelsim or NCsim. After that, a cosimulation [34] or mixed-signal simulation [35] is required to perform the dynamic analysis of the power converter.

The proposed HLST-based design procedure eliminates the need for describing the digital hardware using a low-level HDL and substitutes it by a high-level programming language, such as SystemC, C++ or C. The HLST automatically generates the RTL description that will be later translated into the bitstream to program the FPGA. The directives and constraints defined by the user allow to optimize the design according to performance and/or area criteria. This allows to obtain and compare different implementations in a few minutes. As a consequence, the design space exploration is significantly simplified and further optimization can be achieved. Moreover, the high-level control architecture description also simplifies the joint power converter—digital control design and simulation, since both blocks can be simulated and modeled using the same high-level language.

The C-based input can include a test bench. If required, the same C test bench can be also used to verify the output RTL, therefore improving designer productivity by removing the need to create RTL test benches for RTL verification. The HLST also creates the scripts required to verify the generated RTL through cosimulation with the original test bench and a variety of RTL simulators.

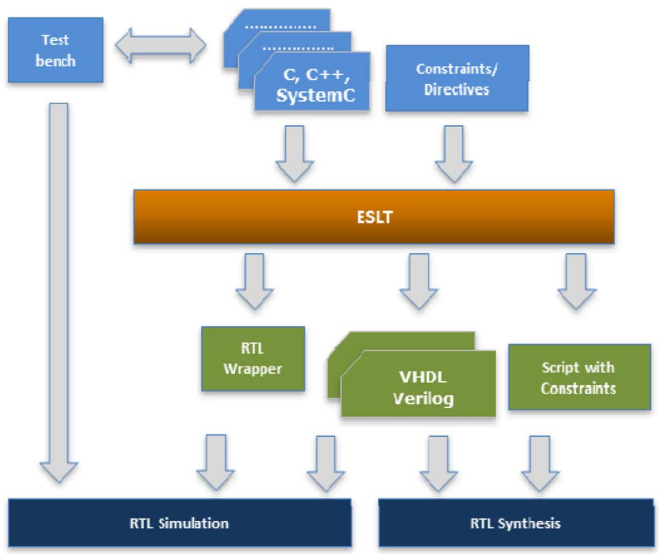

Fig. 2. HLST-based implementation procedure.

Fig. 2 shows the standard HLST-based implementation procedure. This approach enables us to further explore the design space, optimizing hardware–software codesigns and allowing to easily compare different final hardware implementations of the same algorithm in terms of performance, digital resources, and estimated power.

As a consequence, this approach significantly simplifies both the design and simulation tasks. It reduces the designer-required skills and increases the productivity, reducing the design times and improving the reliability. Moreover, it enables to explore the design space to achieve the most efficient implementation in terms of performance or digital resources.

In Section III, we present a design example of a resonant power converter applied to induction heating designed by using the Xilinx Vivado HLS tool. Taking advantage of this tool, the design space is explored to determine the solution that fulfills the constraints with the minimum area.

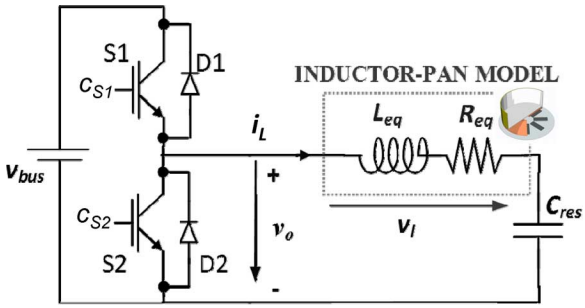

Fig. 3. Half-bridge series resonant inverter for induction heating applications.

III. Study Case: Impedance Identification Algorithm for Resonant Converters

The proposed HLST-based design procedure for digitally controlled power converters has been applied to the design of part of a resonant power converter system.

The considered converter is shown in Fig. 3, where the power converter schematic is detailed. The series resonant half-bridge inverter [36], [37] is composed of two switching devices, S1, S2, typically IGBTs, with two antiparallel diodes D1, D2. The resonant tank is composed of the electrical equivalent of the inductor-pot system, i.e., $R_{\rm eq}-L_{\rm eq}$ [38], and the resonant capacitor $C_{\rm res}$.

The control architecture block diagram is outlined in Fig. 4. It is a full-FPGA hardware–software codesign [7], [39]. The digital hardware platform is classically described in VHDL whereas the software platform is written in C. The inverter output voltage v_o and output current i_L are digitized using $\Sigma\Delta$ analog to digital converters (ADC). This information is used to compute the output power and to perform the resonant tank impedance identification. Besides, the operation checking block (OC) ensures that certain operation constraints are met such as maximum output power and current, and the zero voltage switching (ZVS) to ensure high efficiency. The software platform receives the user commands through the user interface (UI) and computes the modulation pattern for the digital pulse width modulator (DPWM) block using the information received from the HW platform. The input of the system is therefore the desired output power. In addition, it performs several system supervision tasks.

Identifying the resonant tank impedance is a vital task in the proposed resonant inverter, since it determines not only the output power but also the performance and efficiency of the system [40]–[42]. Considering the wide range of different induction loads, each one leading to a different electrical equivalent, and the wide range of output powers required, this task becomes mandatory. This identification will allow to improve static operation, by identifying IH load that may lead to too high currents through the inverter [40], and dynamic operation, by correcting the gain of the digital controller according to the identified IH load [43].

In the past, several identification algorithms have been applied to obtain the harmonic impedance of the resonant load, including the lock-in and the discrete Fourier transform (DFT) algorithms [41]. In this paper, the recursive least-squares identification algorithm (RLS) will be used to prove the feasibility of the proposed HLST-based design procedure. RLS is an example of math-intensive algorithm where the use of floating-point operations may be necessary to obtain a sufficient level accuracy [13]. This block will be designed by using C language and the Xilinx Vivado HLS tool. The digital implementation will be optimized taking advantage of the improved HLST design process, and the performance and quality of the results will be discussed.

A. Least-Squares Identification Algorithm

Letting i_L be the inductor current and v_C be the capacitor voltage in the converter shown in Fig. 3, its continuous-time state-space equation is

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}\cdot\mathbf{x} + \mathbf{B}\cdot v_O \tag{1}$$

where $\mathbf{x}$ is the state vector, $\mathbf{A}$ is the system matrix, $\mathbf{B}$ the input vector, and v_O is the RLC-system input, in this case, the output of the inverter

$$\begin{aligned}\mathbf{x}(t) &= \begin{pmatrix} i_L \\ v_C \end{pmatrix} \\ \mathbf{A} &= \begin{pmatrix} -R_{\rm eq}/L_{\rm eq} & -1/L_{\rm eq} \\ 1/C_{\rm res} & 0 \end{pmatrix} \\ \mathbf{B} &= \begin{pmatrix} 1/L_{\rm eq} \\ 0 \end{pmatrix}.\end{aligned} \tag{2}$$

The discrete-time model equivalent to the continuous-time system with the state-space description (2) is given by

$$\mathbf{x}(k+1) = \mathbf{F}\cdot\mathbf{x}(k) + \mathbf{G}\cdot v_O(k). \tag{3}$$

Matrices $\mathbf{F}$ and $\mathbf{G}$ of the discrete time state-space description for the configuration depend on the numerical system and on the selected sampling period T_s. Using the trapezoidal integration method and assuming no change in v_o during consecutive samples, we have

$$\mathbf{F} = \left(\mathbf{I} - \frac{T_S}{2}\mathbf{A}\right)^{-1}\left(\mathbf{I} + \frac{T_S}{2}\mathbf{A}\right) \tag{4}$$

$$\mathbf{G} = \left(\mathbf{I} - \frac{T_S}{2}\mathbf{A}\right)^{-1} T_S\mathbf{B}. \tag{5}$$

Equations (3)–(5) lead to the following difference equation:

$$i_L(k+1) = \theta_1\cdot i_L(k) + \theta_2\cdot(v_O(k) - v_C(k)). \tag{6}$$

Neglecting second-order terms in T_S, the difference equation coefficients are

$$\theta_1 = \frac{1 - \frac{R_{\rm eq}}{L_{\rm eq}}\frac{T_S}{2}}{1 + \frac{R_{\rm eq}}{L_{\rm eq}}\frac{T_S}{2}} \qquad \theta_2 = \frac{\frac{1}{L_{\rm eq}}T_S}{1 + \frac{R_{\rm eq}}{L_{\rm eq}}\frac{T_S}{2}}. \tag{7}$$

Recording $N+1$ samples of i_L, and N samples of v_C and v_O, the following system of equations is obtained:

$$\begin{pmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{pmatrix} = \begin{pmatrix} i_L(1) & v_O(1) - v_C(1) \\ i_L(2) & v_O(2) - v_C(2) \\ \vdots & \vdots \\ i_L(N) & v_O(N) - v_C(N) \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \tag{8}$$

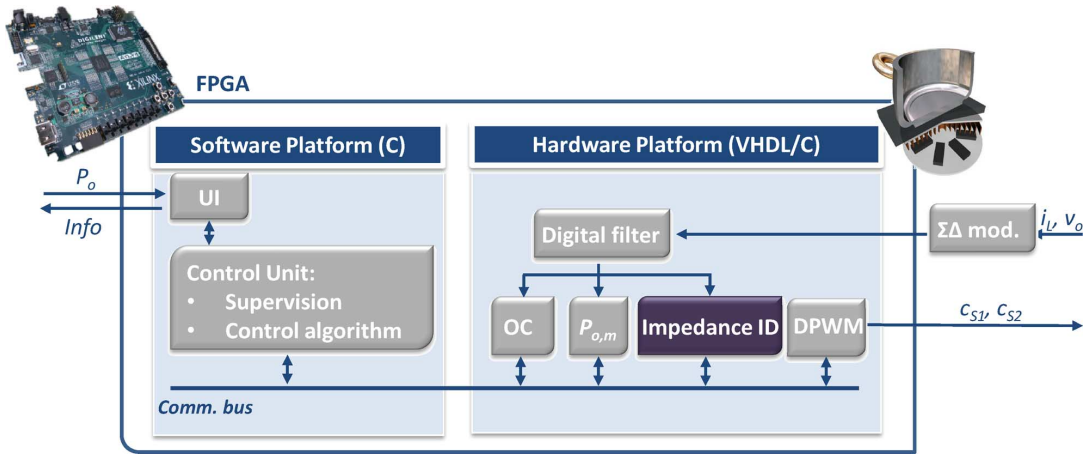

Fig. 4. Control architecture block diagram: Impedance ID block is designed using a HLST.

where $y(k) = i_L(k+1)$. This system of equations can be written as $\mathbf{y} = \Phi \cdot \boldsymbol{\theta}$. The LS estimation of the real coefficient vector is given by [44]

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{\Phi}^t \cdot \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^t \cdot \mathbf{y} \tag{9}$$

where $\boldsymbol{\Phi}^t$ is the transpose of $\boldsymbol{\Phi}$. In order to perform online identification the algorithm must run continuously as new measurement data are taken; thus, a recursive form of the LS algorithm must be implemented. The recursive least squares (RLS) estimation of the real coefficient vector is obtained as follows [44]:

$$\mathbf{L}(k) = \frac{\mathbf{P}(k-1) \cdot \mathbf{z}(k)}{1 + \mathbf{z}(k)^t \cdot \mathbf{P}(k-1) \cdot \mathbf{z}(k)} \tag{10}$$

$$\mathbf{P}(k) = (\mathbf{I}_2 - \mathbf{L}(k) \cdot \mathbf{z}(k)^t) \cdot \mathbf{P}(k-1) \tag{11}$$

$$\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \mathbf{L}(k)(\mathbf{y}(k) - \hat{\boldsymbol{\theta}}(k-1)^t \cdot \mathbf{z}(k)) \tag{12}$$

where $\mathbf{z}(k) = (\mathbf{i_L}(k) \quad \mathbf{v_O}(k) - \mathbf{v_C}(k))^t$ is the vector of measurements, and $\mathbf{I}_2$ is the 2×2 identity matrix. $\mathbf{P}(0)$ is initialized to a diagonal matrix with large positive diagonal elements, and $\boldsymbol{\theta}(0)$ is initialized to the zero vector.

Once the vector of coefficients is obtained, the load parameters can be estimated as

$$R_{\text{eq}} = \frac{1 - \hat{\theta}_1}{\hat{\theta}_2}, L_{\text{eq}} = \frac{1 + \hat{\theta}_1}{\hat{\theta}_2} \frac{T_S}{2}. \tag{13}$$

The above equation requires the measurements of i_L, v_C, and v_O in order to perform identification. To reduce the cost by avoiding the need of measuring v_C, the following difference equation can be used:

$$i_L(k+1) = \theta_1 \cdot i_L(k) + \theta_2 \cdot i_L(k-1) + \theta_3 \cdot (v_O(k) - v_O(k-1)). \tag{14}$$

In this case, neglecting second-order terms in T_S, the difference equation coefficients become

$$\theta_1 = \frac{2}{1 + \frac{R_{\text{eq}}}{2L_{\text{eq}}} T_S}$$

$$\theta_2 = \frac{-1}{1 + \frac{R_{\text{eq}}}{L_{\text{eq}}} T_S}$$

$$\theta_3 = \frac{\frac{1}{L_{\text{eq}}} T_S}{1 + \frac{R_{\text{eq}}}{2L_{\text{eq}}} T_S}. \tag{15}$$

TABLE I
ARITHMETICAL OPERATIONS OF THE RLS

Number	RLS A	RLS B
Additions/sub.	15	36
Multiplications	25	54
Reciprocal	1	1

Once the coefficient vectors are obtained by the RLS algorithm, the load parameters can be estimated as

$$R_{\text{eq}} = \frac{-2}{\hat{\theta}_3} \left(1 + \frac{\hat{\theta}_1}{2\hat{\theta}_2}\right)$$

$$L_{\text{eq}} = \frac{\hat{\theta}_1}{2\hat{\theta}_3} T_S. \tag{16}$$

Taking into account that matrix $\mathbf{P}$ is symmetrical, the number of operations to perform in each iteration of the RLS algorithm for both implementations is shown in Table I. Note that a reciprocal operator is used instead of a divider. The recursive algorithm with the v_C measurement is denoted as A, and the recursive algorithm without v_C measurement is denoted as B. The RLS estimation of the real coefficient vector will be implemented in hardware as a MicroBlaze peripheral, while the load parameters will be computed in software by the microprocessor.

Second-order terms in T_S have been neglected in the derivation of the difference equation coefficients. That implies that the sampling time during one RLS algorithm iteration has to be small enough. That imposes a constraint in the sampling period of the ADC. T_S is usually equal to the maximum time available to perform computations T_C when working in real time. However, in order to increase T_C while keeping T_S sufficiently small to ensure algorithm convergence, the samples are organized as shown in Fig. 5. The sampling time is kept to a low value, and the algorithm is computed in T_C intervals. These values are optimized in simulation before implementing the algorithm.

Section III-B shows the implementation carried out by using an HLST. Taking advantage of the expedited design, several design alternatives will be compared.

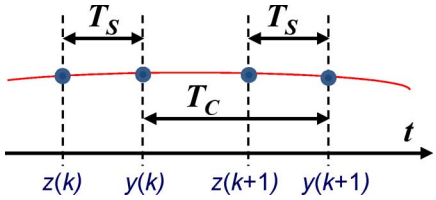

Fig. 5. Computing (T_c) and sampling (T_s) period organization.

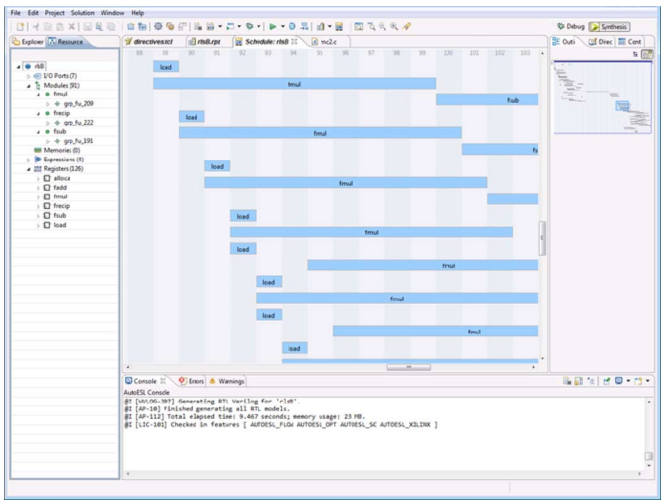

Fig. 6. Vivado HLS graphical interface (scheduling graph). Horizontal lines represent arithmetic operations and columns represent clock cycles.

B. Implementation and Results Comparison

The FPGA used to implement the proposed algorithm is a Spartan-6 LX45-2. Based on simulation results, the sampling time T_S is set to 0.1 μs, and the computing time T_C is set to 1 μs. The algorithm has been initially optimized taking into account matrix symmetry and factorizing common subexpressions. Simple-precision floating-point arithmetic has been used, although HLST allows defining any other numeric format to operate, for instance, with fixed-point arithmetic.

The design constraints are the clock frequency and the throughput. Latency is not an issue in this application. They have been specified using the following HLST directives:

- create_clock−period10;
- set_directive_pipeline −II 100 "rlsA"

The first one sets a clock frequency of 100 MHz. The second one enables pipelining and sets a throughput of 100 clock cycles so that the ratio of throughput to frequency is lower than or equal to the T_C constraint. The following directives has been used to create a Xilinx MicroBlaze PLB peripheral:

- set_directive_resource-core PLB46S-metadata {−bus_bundle my_bus} [43] "rlsA" Theta1;
- set_directive_ resource-core PLB46S-metadata {−bus_bundle my_bus} "rlsA" Theta2

This tool automatically generates the data-path that meets the given constraints, selects the pipelining level of the arithmetic operator instances, and generates the control unit that controls the scheduling of the operations. Fig. 6 shows the scheduling graph as it is shown in the Vivado HLS graphical user interface. It should be noted that the design of the control unit of a deep pipelined data-path that controls the scheduling shown in Fig. 6 is a complex problem.

TABLE II
SIMULATION TIME COMPARISON

Method	Time A	Time B
Matlab	1 s	2 s
VHDL	7 min 5 s	15 min 8 s
C	1.5 ms	2 ms

TABLE III
DIGITAL IMPLEMENTATIONS FOR THE RLS A ALGORITHM

Digital Resources	HLST		
	Perf. 1	Perf. 2	Perf. 3
Add./Sub.	1(6)	1(5)	1(3)
Mult.	1(8)	1(7)	2(4)
Reciprocal.	1(24)	1(13)	1(6)
CLK Freq. (MHz)	150	100	50
Throughput. (cycles)	118	78	50
TC (μs)	0.79	0.78	1.0
DSP48A	15	15	20
Slices	663	567	675
Power (mW)	47	28	14

Algorithms A and B have been implemented using the Vivado HLS tool. For each algorithm, three clock frequencies have been considered aiming at obtaining the desired performance while minimizing the area.

Table II compares the simulation time during 10 ms for both algorithms using several design alternatives. It can be seen that the C approach, used in HLSTs, is significantly faster than both MATLAB and VHDL simulations. This, along with the fact that the same C testbench can be used to simulate the final RTL implementation, simplifies the design process.

Tables III and IV summarizes the results obtained after place-and-route for the A and B algorithms, respectively, implemented in order to meet performance specifications. Performance specifications have been obtained for three different clock frequencies, leading to different resource utilization and estimated power consumption. Implementations achieved by using Vivado HLS tool get satisfactory results for the proposed system. The HLST designs with $f_{\mathrm{CLK}} = 150$ MHz and $f_{\mathrm{CLK}} = 100$ MHz require the same number of operator instances, but the level of pipelining of these operators, shown in parentheses, is different. The designs with $f_{\mathrm{CLK}} = 50$ MHz require a higher number of slices, but the power consumption is

TABLE IV
DIGITAL IMPLEMENTATIONS FOR THE RLS B ALGORITHM

Digital Resources	HLST		
	Perf. 1	Perf. 2	Perf. 3
Add./Sub.	1(6)	1(5)	3(4)
Mult.	1(11)	1(7)	3(4)
Reciprocal .	1(25)	1(13)	1(8)
CLK Freq. (MHz)	150	100	50
Throughput (cycles)	137	100	50
TC (μs)	0.91	1.0	1.0
DSP48A	15	15	25
Slices	859	868	1677
Power (mW)	65	39	29

```
set_directive_resource -core PLB46S -metadata {-bus_bundle my_bus} "rlsA" Theta2
```

TABLE V
CODE COMPLEXITY COMPARISON: CODE LINE NUMBER

Algorithm	HLST	VHDL		
		Perf. 1	Perf. 2	Perf. 3
RLS A	35	812	747	646
RLS B	63	1112	1009	1275

reduced. When designing complex control systems, the design space exploration is also an extraordinary time-consuming task to be performed manually by a designer.

In order to establish an unbiased comparison that does not rely on the designer skills, the code complexity comparison, measured as the required line number, is shown in Table V. From this table, it is clear that the use of an HLST significantly simplifies the design process. Moreover, data dependencies are detected, and the operation schedule and control unit are automatically generated, as shown in Fig. 6, and floating-point implementation with pipelined operators is directly performed. These are usually identified as the most difficult and time-consuming tasks when implementing FPGA-based controllers, and they are significantly simplified by the use of HLSTs.

IV. CONCLUSION

In this paper, the design of FPGA-based control architecture for power converters taking advantage of high-level synthesis tools has been discussed. The design procedure using HLSTs has been highlighted, and it has been applied to the design of an impedance identification tool for a resonant power converter using the Xilinx Vivado HLS tool (former AutoESL). The results obtained prove the feasibility of the design process using HLSTs for FPGA-based high-demanding applications.

The HLST automatically generates from a C behavioral description of the algorithm the pipelined data-path and the control unit that fulfills the given performance constraints while minimizing the area. It significantly simplifies the design and simulation process and does not require the designer to have specific HDL design skills. In addition, it achieves an optimization level that cannot be easily obtained in a classical design approach. As a consequence, it significantly reduces the design time of FPGA-based power converter systems, enables design exploration, and, therefore, should be considered as an alternative in FPGA design for power converter systems.

ACKNOWLEDGMENT

The authors would like to thank Xillinx Inc. for the donation of the AutoESL tool.

REFERENCES

[1] A. Mohammed, "Digital control of power converters," in *Proc. IEEE Appl. Power Electron. Conf. Expo., Professional Education Seminars Workbook*, 2008.
[2] A. Quagli, D. Fontanelli, L. Greco, L. Palopoli, and A. Bicchi, "Design of embedded controllers based on anytime computing," *IEEE Trans. Ind. Inf.*, vol. 6, no. 4, pp. 492–502, Nov. 2010.
[3] A. Malinowski and Y. Hao, "Comparison of embedded system design for industrial applications," *IEEE Trans. Ind. Inf.*, vol. 7, no. 2, pp. 244–254, May 2011.
[4] T. Orlowska-Kowalska and M. Kaminski, "FPGA implementation of the multilayer neural network for the speed estimation of the two-mass drive system," *IEEE Trans. Ind. Inf.*, vol. 7, no. 2, pp. 436–445, Mar. 2011.
[5] S. Xiaoyin and S. Dong, "Development of a new robot controller architecture with FPGA-based IC design for improved high-speed performance," *IEEE Trans. Ind. Inf.*, vol. 3, no. 4, pp. 312–321, Nov. 2007.
[6] P. Conmy and I. Bate, "Component-based safety analysis of FPGAs," *IEEE Trans. Ind. Inf.*, vol. 6, no. 2, pp. 195–205, May 2010.
[7] O. Lucía, L. A. Barragán, J. M. Burdío, O. Jiménez, D. Navarro, and I. Urriza, "A versatile power electronics test-bench architecture applied to domestic induction heating," *IEEE Trans. Ind. Electron.*, vol. 58, no. 3, pp. 998–1007, Mar. 2011.
[8] A. J. Moradewicz and M. P. Kazmierkowski, "Contactless energy transfer system with FPGA-controlled resonant converter," *IEEE Trans. Ind. Electron.*, vol. 57, no. 9, pp. 3181–3190, Sep. 2010.
[9] L. Idkhajine, E. Monmasson, M. W. Naouar, A. Prata, and K. Boullaga, "Fully integrated FPGA-based controller for synchronous motor drive," *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 4006–4017, Oct. 2009.
[10] E. Monmasson, L. Idkhajine, M. N. Cirstea, I. Bahri, A. Tisan, and M. W. Naouar, "FPGAs in industrial control applications," *IEEE Trans. Ind. Inf.*, vol. 7, no. 2, pp. 224–243, May 2011.
[11] E. Monmasson, L. Idkhajine, and M. W. Naouar, "FPGA-based controllers," *IEEE Ind. Electron. Mag.*, vol. 5, no. 1, pp. 14–26, 2011.
[12] M. W. Naouar, E. Monmasson, A. A. Naassani, I. Slama-Belkhodja, and N. Patin, "FPGA-based current controllers for ac machine drives—A review," *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1907–1925, Apr. 2007.
[13] C. Buccella, C. Cecati, and H. Latafat, "Digital control of power converters: A survey," *IEEE Trans. Ind. Inf.*, vol. 8, no. 3, pp. 437–447, Aug. 2012.
[14] E. Monmasson and M. N. Cirstea, "FPGA design methodology for industrial control systems: A review," *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1824–1842, Aug. 2007.
[15] G. Martin and G. Smith, "High-level synthesis: Past, present, and future," *IEEE Design Test Computers*, vol. 26, no. 4, pp. 18–25, 2009.
[16] J. Cong, L. Bin, S. Neuendorffer, J. Noguera, K. Vissers, and Z. Zhiru, "High-level synthesis for FPGAs: From prototyping to deployment," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 4, pp. 473–491, Aug. 2011.

[17] D. Varma, D. Mackay, and P. Thiruchelvam, "Easing the verification bottleneck using high level synthesis," in *Proc. 28th VLSI Test Symp.*, 2010, pp. 253–254.
[18] J. Cong, "A new generation of C-base synthesis tool and domain-specific computing," in *Proc. IEEE Int. SOC Conf.*, 2008, pp. 386–386.
[19] C. Economakos and G. Economakos, "FPGA implementation of PLC programs using automated high-level synthesis tools," in *Proc. IEEE Int. Symp. Ind. Electron.*, 2008, pp. 1908–1913.
[20] K. Vissers, S. Neuendorffer, and J. Noguera, "Building real-time HDTV applications in FPGAs using processors, AXI interfaces and high level synthesis tools," in *Proc. Design, Autom. Test Europe Conf. Exhib.*, 2011, pp. 1–3.
[21] X. Changliang, W. Meng, S. Zhanfeng, and L. Tao, "Robust model predictive current control of three-phase voltage source PWM rectifier with online disturbance observation," *IEEE Trans. Ind. Inf.*, vol. 8, no. 3, pp. 459–471, Aug. 2012.
[22] M. Zamora and M. P. Henry, "An FPGA implementation of a digital Coriolis mass flow metering drive system," *IEEE Trans. Ind. Electron.*, vol. 55, no. 7, pp. 2820–2831, Jul. 2008.
[23] I. Urriza, L. A. Barragán, D. Navarro, J. I. Artigas, and O. Lucía, "Word length selection method for controller implementation on FPGAs using the VHDL-2008 fixed-point and floating-point packages," *EURASIP J. Embedded Syst.*, vol. 2010, no. 1, pp. 1–12, 2010.
[24] P. G. Paulin and J. P. Knight, "Force-directed scheduling for the behavioral synthesis of ASICs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 8, no. 6, pp. 661–679, Jun. 1989.
[25] M. E. Elbuluk, G. C. Verghese, and J. G. Kassakian, "Sampled-data modeling and digital control of resonant converters," *IEEE Trans. Power Electron.*, vol. 3, no. 3, pp. 344–354, Jul. 1988.
[26] H. Matsuo, F. Kurokawa, and K. Higashi, "Dynamic characteristics of the digitally controlled DC-DC converter," *IEEE Trans. Power Electron.*, vol. 4, no. 4, pp. 419–426, Sep. 1989.
[27] P. Murphy *et al.*, "Study of digital vs analog control," in *Proc. CPES Seminar*, 2002.
[28] A. de Castro, T. Riesgo, O. Garcia, and J. Uceda, "A methodology to design custom hardware digital controllers for switching power converters," in *Proc. IEEE Power Electron. Specialists Conf.*, 2004, vol. 6, pp. 4676–4681.
[29] B. J. Patella, A. Prodic, A. Zirger, and D. Maksimovic, "High-frequency digital PWM controller IC for DC-DC converters," *IEEE Trans. Power Electron.*, vol. 18, no. 1, pp. 438–446, Jan. 2003.
[30] S. Hong-Seok and N. Kwanghee, "Dual current control scheme for PWM converter under unbalanced input voltage conditions," *IEEE Trans. Ind. Electron.*, vol. 46, no. 5, pp. 953–959, May 1999.
[31] R. Jos *et al.*, "Predictive current control of a voltage source inverter," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 495–503, Jan. 2007.
[32] S. Buso, L. Malesani, and P. Mattavelli, "Comparison of current control techniques for active filter applications," *IEEE Trans. Ind. Electron.*, vol. 45, no. 5, pp. 722–729, May 1998.
[33] F. Wang, S. Wei, D. Boroyevich, S. Ragon, V. Stefanovic, and M. Arpilliere, "Voltage source inverter," *IEEE Ind. Applic. Mag.*, vol. 15, no. 2, pp. 24–33, Mar./Apr. 2009.
[34] N. Pongratananukul and T. Kasparis, "Cosimulation for verification of digital control implementation," in *Proc. IEEE Power Electron. Specialists Conf.*, 2004, vol. 5, pp. 3609–3614.
[35] L. A. Barragán, I. Urriza, D. Navarro, J. I. Artigas, J. Acero, and J. M. Burdío, "Comparing simulation alternatives of FPGA-based controllers for switching converters," in *Proc. IEEE Int. Symp. Ind. Electron.*, 2007, pp. 419–424.
[36] O. Lucía, J. M. Burdío, I. Millán, J. Acero, and L. A. Barragán, "Efficiency oriented design of ZVS half-bridge series resonant inverter with variable frequency duty cycle control," *IEEE Trans. Power Electron.*, vol. 25, no. 7, pp. 1671–1674, Jul. 2010.
[37] O. Lucía, J. M. Burdío, L. A. Barragán, J. Acero, and I. Millán, "Series-resonant multiinverter for multiple induction heaters," *IEEE Trans. Power Electron.*, vol. 24, no. 11, pp. 2860–2868, Nov. 2010.
[38] J. Acero, C. Carretero, I. Millán, O. Lucía, R. Alonso, and J. M. Burdío, "Analysis and modeling of planar concentric windings forming adaptable-diameter burners for induction heating appliances," *IEEE Trans. Power Electron.*, vol. 26, no. 5, pp. 1546–1558, May 2011.
[39] O. Lucía, I. Urriza, L. A. Barragan, D. Navarro, O. Jiménez, and J. M. Burdío, "Real-time FPGA-based hardware-in-the-loop simulation test-bench applied to multiple output power converters," *IEEE Trans. Ind. Applic.*, vol. 47, no. 2, pp. 853–860, Mar./Apr. 2011.
[40] O. Lucía, J. M. Burdío, I. Millán, J. Acero, and D. Puyal, "Load-adaptive control algorithm of half-bridge series resonant inverter for domestic induction heating," *IEEE Trans. Ind. Electron.*, vol. 56, no. 8, pp. 3106–3116, Aug. 2009.
[41] O. Jiménez, O. Lucía, L. A. Barragán, D. Navarro, J. I. Artigas, and I. Urriza, "FPGA-based test-bench for resonant inverter load characterization," *IEEE Trans. Ind. Inf.*, vol. 99, p. 1, 2012, DOI: 10.1109/TII.2012.2226184.
[42] H. Sarnago, O. Lucía, A. Mediano, and J. M. Burdío, "Modulation scheme for improved operation of a RB-IGBT based resonant inverter applied to domestic induction heating," *IEEE Trans. Ind. Electron.*, vol. 60, no. 5, pp. 2066–2073, May 2013.
[43] A. Dominguez, A. Otín, L. A. Barragán, O. Lucía, and J. I. Artigas, "Modeling of resonant inverters with high harmonic content using the extended describing function method," in *Proc. IEEE Annu. Conf. Ind. Electron. Soc.*, 2012, pp. 5953–5958.
[44] L. Ljung, *System Identification*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1999.

Denis Navarro received the M.Sc. degree in microelectronics from the University of Montpellier, Montpellier, France, in 1987, and the Ph.D. degree from the University of Zaragoza, Zaragoza, Spain, in 1992.

Since September 1988, he has been with the Department of Electronic Engineering and Communications, University of Zaragoza, Zaragoza, Spain, where he is an Associate Professor. His current research interests include CAD for VLSI, low-power ASIC design, and modulation techniques for power converters. He is involved in the implementation of new applications of integrated circuits. In 1993, he designed the first SPARC microprocessor in Europe.

Dr. Navarro is a member of the Aragon Institute for Engineering Research (I3A).

Óscar Lucía (S'04–M'11) received the M.Sc. and Ph.D. degrees in electrical engineering from the University of Zaragoza, Zaragoza, Spain, in 2006 and 2010, respectively.

He is an Assistant Professor with the Department of Electronic Engineering and Communications, University of Zaragoza, Zaragoza, Spain. His main research interests include induction-heating applications, multiple-output power converters, resonant inverters, and digital control and modulation strategies applied to power converters.

Dr. Lucía is a member of the Aragon Institute for Engineering Research (I3A).

Luis A. Barragán received the M.Sc. and the Ph.D. degrees in physics from the University of Zaragoza, Zaragoza, Spain, in 1988 and 1993, respectively.

He is an Associate Professor with the Department of Electronic Engineering and Communications, University of Zaragoza, Zaragoza, Spain. He has been involved with different research and development projects on induction heating systems for home appliances. His research interests include digital circuits design and digital control of inverters for induction heating applications.

Dr. Barragán is a member of the Aragon Institute for Engineering Research (I3A).

Isidro Urriza received the M.Sc. and Ph.D. degrees in electrical engineering from the University of Zaragoza, Zaragoza, Spain, in 1991 and 1998, respectively.

He is currently an Associate Professor with the Department of Electronic Engineering and Communications, University of Zaragoza, Zaragoza, Spain. He has been involved with different research and development projects. His main research interests include digital implementation of modulation techniques for power converters.

Dr. Urriza is a member of the Aragon Institute for Engineering Research (I3A).

Óscar Jiménez (S'10) received the M.Sc. degree in electrical engineering from the University of Zaragoza, Zaragoza, Spain, in 2009, where he is currently working toward the Ph.D. degree.

He currently works in the field of domestic induction-heating, resonant inverters, and digital control applied to power converters.

Mr. Jiménez is a member of the Aragon Institute for Engineering Research (I3A).