IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)
e-ISSN: 2278-1676,p-ISSN: 2320-3331
PP 21-27
www.iosrjournals.org

Reconfigurable High Speed Power Electronics Controller

Ajeesh A[1], Sreedevi V. S[2], Nevin Samuel[3]
[1](Power Electronics Group, Center for Development of Advanced Computing, India)
[2](Power Electronics Group, Center for Development of Advanced Computing, India)
[3](Power Electronics Group, Center for Development of Advanced Computing, India)

Abstract: *Off-the-shelf microcontrollers and DSPs used for power electronics applications have got limited resources and becomes obsolete faster. Power electronic systems usually have a lifetime of 10 to 15 years, whereas a conventional processor becomes obsolete within 4 to 5 years making it difficult to provide long term support. This necessitates system redesign, increasing the time-to-market of the product., especially in multi-processor environment. This paper presents the development of a high speed reconfigurable controller for power electronics based on System On a Programmable Chip (SOPC) design methodology. The controller is realized in an Field Programmable Gate Array (FPGA) using integrated soft CPU core. The elements for implementing control algorithms for power electronics such as PI controllers, PWM generators, timers, filters etc. are created as hardware blocks in soft form with custom functions /instructions for accessing them, making the design reconfigurable, reusable and FPGA independent. The soft CPU core IP along with the custom developed reconfigurable hardware IPs are used for implementing advanced algorithms in high level language or in assembly language for control, communication protocols and user interface. The high speed reconfigurable controller thus provides an optimized environment of software and hardware for real-time control of power electronics and cost effective alternative to conventional controllers.*
Keywords: *DSP, FPGA, Power Electronics, Re-configurability, SOPC*

I. Introduction

The presence of power electronic systems is felt in the day to day life of common man beginning from LED lighting to EVs and metro rail coaches. At the core of these systems lies a controller for meeting the system requirements. These controllers are often realized in off-the-shelf microcontrollers or DSPs which are stiff and limited in terms of resources and are at the risk of becoming obsolete. Conventionally these limitations are overcome with the help of other on-board peripherals and PLDs. This method is lopsided by increased power requirement, component density and cost. ASICs were considered as the alternatives for general purpose controllers for long time. However they proved to be a costlier substitute as any change in the system will require re-design and re-fabrication which is expensive as well as time consuming, taking into account the complexity of recent embedded design considerations. This paper presents an alternative in the form of a High Speed Re-configurable Power Electronics Controller developed in an FPGA using SOPC methodology. Reconfigurable architectures are those that can be dynamically configured and reused for varying applications. Reusable components including CPU come in the form of Intellectual Property (IP) cores as soft, hard or firm cores. Generally, soft cores provide the highest flexibility, allowing many core parameters to be changed prior to synthesis, while hard cores provide little or no flexibility [1]. For designing High Speed Re-configurable Power Electronics Controller, some of the typical power electronics applications such as converters and inverters were analysed so as to identify the requirements of control algorithms and modules like PWM, PI, UART, Phase Transformations etc. Since their inception in the mid-1980s, Field Programmable Gate Arrays (FPGAs) were a popular choice for reconfigurable prototyping and production of products in small to moderate quantities [1], [2].System design in FPGA chips is typically done using CAD (Computer Aided Design) software which supports tools for integrating IP cores of processors and peripherals including memory and also provide application software development environment. Modern CAD tools support design entry using several methods. As the complexity of the circuit grows, Hardware Description Languages (HDLs) become the only practical choice. HDLs support circuit description using high-level language constructs. Low-level implementation details are handled by the CAD tool automatically, so that the designer can focus on the design functionality. The required modules are developed in HDL as generic IP cores. These cores are added on to the IP library of the CAD tool so that they can be placed as add-on peripherals to the generic CPU soft core IP which is readily available. The drivers for integration of these IPs to the CPU core and custom functions for using them in the application software are also developed.The System Architecture, Design Methodology, Implementation and Application Development are discussed in further sections.

II. Sopc Based Pe Controller Architecture

Conventional designs of Power Electronic control are based on microcontrollers and Digital Signal Processors (DSPs) [3], [4]. Microcontrollers market is so dynamic that for a designer, overheads are high to keep pace with changing technology. Traditional DSPs are well suited for algorithmic intensive applications, but are limited in performance in terms of clock rate and the sequential nature of their internal design. Typically, three or four clock cycles are required per arithmetic logic unit (ALU) operation which is again limited by wait states requirement of external memories. Also limited resources such as PWM channels and communication interface controllers offer other challenges. Several solutions have been proposed in the past, including using multiple-core DSPs within a device or multiple DSPs on a board; however, such schemes often increase costs significantly and simply shift the problem to another arena.

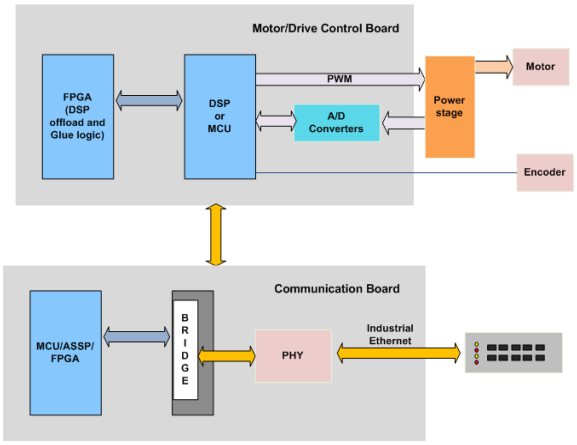

Fig.1: Block diagram of PE controller system implemented in Conventional Method

A solution to the increasing problems of DSP implementations came with the introduction of FPGA technology which allows re-configurability. In contrast to the traditional DSP, FPGAs have massively parallel structures containing a uniform array of configurable logic blocks (CLBs), memory, DSP slices and other elements such as hardware multipliers. Besides soft processor cores, several other IP cores including memory controllers, interrupt controllers, Ethernet controllers, UARTs , timers, buses, and others are readily available for direct use in a highly optimized form within the FPGA. Designer can additionally create specific IP cores and place them along with in-built cores. Using these modules and on-chip memory, all components of a computer system can be placed on a single FPGA. This concept is known as a System on Programmable Chip (SOPC). The 'system builder' tools available with CAD software help in realizing this.

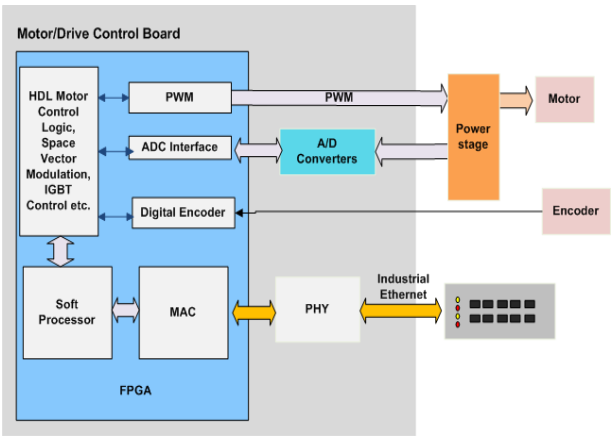

Fig.2: Block diagram of PE controller system implemented in SOPC Method

Any change in application will result mainly in changes in IP configuration inside FPGA whereas changes at board level hardware will be very limited. Unlike the DSP environment where the application software is highly linked to the hardware, the IP cores in FPGA environment is realized in HDLs, making it easy to port between different variants or even between FPGAs from different vendors. Thus reliability and reusability of the design is more and a better time to market can be achieved [5],[6],[7].

III. Design Methodology

The methodology, as shown in the Fig. 3, is a 2 stage process. After analysing the system requirements, the controller hardware is created with a soft processor IP core [8] at its center. The processor should have the important feature of integrating custom hardware or peripherals inside the processor and if necessary to include custom instruction to support that hardware. The system requirements are primarily met by integrating off-the-shelf IP cores with the soft processor. IP cores that are not available in the system builder software are created in HDL and added on to the library for integration with the processor core inside an FPGA. Depending on the soft core processor, only the top level entity of HDL code needs to be modified. The controller software is developed during the second stage in a software development environment for the soft processor. A hardware information file generated in stage 1 using the system builder tool provides the necessary information regarding the hardware to the software environment. Off-the-shelf hardware modules have built-in driver files for the software interface. For the custom developed IP cores, the onus is on designer to create these driver files.

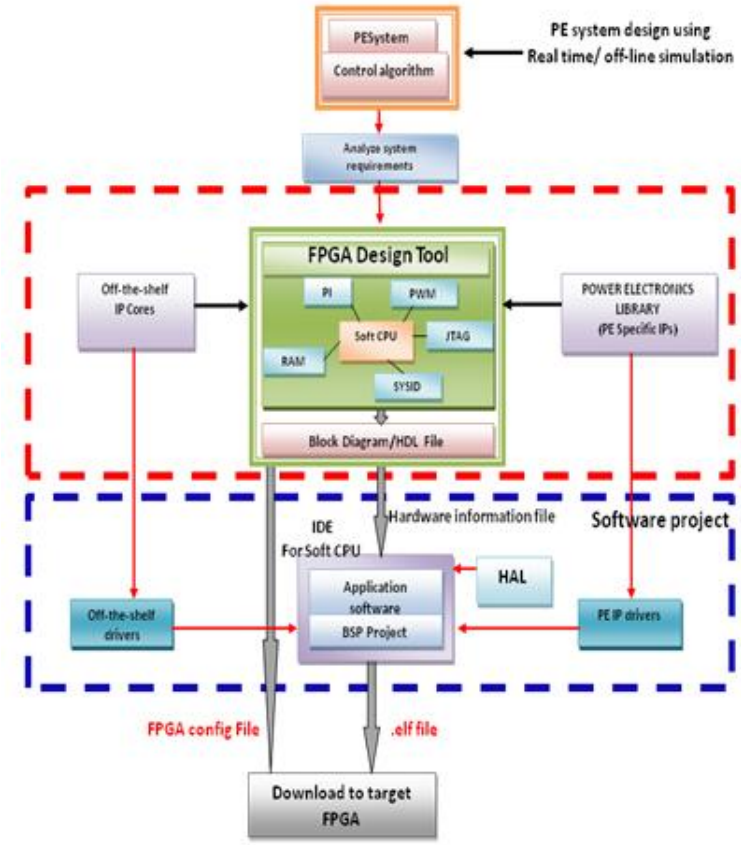

Fig.3: Design Methodology for SOPC system development

III.1 Identification of Intellectual Property (IP) Cores Requirement

A drive used in traction application is selected as the reference for analyzing the system requirements for the controller developed. The motor drive has Power Electronic modules - active front end rectifier converter and AC drive inverter. On analyzing the rectifier control and inverter control, it is obvious that the controller in both cases consists of many common blocks and some specific blocks.

Table 1: IP cores for rectifier and inverter application

Rectifier Control	**Inverter Control**
IP cores for basic control	
Processor core	
PID Controller	
Unit Vector Generator	Motor Dynamic Model
PWM(4 Channel)	PWM(6 Channel)
Limiter	Phase Transformers

Of the modules specified in TABLE 1, in a conventional DSP environment, the processor core and PWM generation modules are available as hardware modules where as the others are to be realized as software functions, making their operation sequential and interdependent. The controller software thus becomes lengthy and complicated with higher memory requirement for context switching, for a fellow designer to understand or debug easily. Even the hardware modules will such as PWM channels and generation modules will be limited, forcing the designer opt for multiple processors. In SOPC methodology, while the processor core is readily available in soft form, other modules required are realised as hardware modules in HDL. Unlike the software function, these module can operate in parallel and multiple instances are possible, limited only by the size of hte FPGA. These control and data registers of these modules are mapped in the processor memory, thus reducing the control software to a data dumping and reading routine.

III.2 Hardware design

The FPGA design tool is used for the hardware development tool. By using the tool, one can select the controller components from the SOPC library including the custom designed IP cores and can generate the interconnect logic manually or automatically. This procedure generates HDL files that define all components of the system, and a top-level HDL file that connects all the components together. Also user can import HDL modules and entities using Verilog HDL or VHDL as custom components external to the controller.The custom IPs are developed in HDL and they need to be synthesizable. The functionality and timing requirements of these modules can be properly monitored and verified using test bench or IP verification methods [9]. This is a time consuming but, one time process. After this, the top level of the IP is edited as per the bus interface specifications specific to the processor soft core and is incorporated into SOPC library. In this way, a POWER ELECTRONICS library was developed, consisting of the custom developed Power Electronics IPs. Also, specific functions are developed for these IPs to access them from the application software. The functions developed for custom IPs are simple and are easily understandable for a software designer. The custom functions developed for PI controller IP are as follows:

- initialize PI module with Kp and Ki Registers *hw_init_pi(PI_BASE, Kp_val ,Ki_val);*
- enable/disable PI Module *hw_enable_pi(PI_BASE, PI_Cntrl_Reg);*
- input to the PI module *hw_pi_in(PI_BASE, error);*
- out from the PI module *output = hw_pi_read_output(PI_BASE)* 19 custom IPs were developed for the Power Electronics library. These are listed in TABLE 2.

Table 2: IP cores PE Library

No	***IP***	***POWER ELECTRONICS LIBRARY Component name***	***Specification***
1	UART	UART_PEG_V1	Data width: 8-32 bit
2	Serial ADC Controller	ADC_PEG_V1	Resolution: Up to 16 bit
3	Serial DAC Controller	DAC_PEG_V1	Resolution: Up to 16 bit
4	PWM Generator	PWM_PEG_V1	Resolution: Up to 16 bit Max. Clock Freq - 160 MHz for Cyclone III FPGA
5	3 phase PWM generator	PWM_3PH_PEG_V1	Resolution: Up to 16 bit Max. Clock Freq - 160 MHz for Cyclone III FPGA
6	Timer for interrupts	TIMER_PEG_V1	Resolution: Up to 16 bit
7	Serial Peripheral Interface	SPI_PEG_V1	Resolution: Up to 16 bit
8	PI Controller	PI_PEG_V1	Signed Resolution: Up to 16 bit
9	ABC to DQ transformation	ABC_TO_DQ_PEG_V1	Signed Resolution: Up to 16 bit
10	DQ to ABC transformation	DQ_TO_ABC_PEG_V1	Signed Resolution: Up to 16 bit
11	Integrator (Backward Euler)	INTEGRATOR_PEG_V1	Signed Resolution: Up to 16 bit
12	Alphabeta to ABC transformation	ALPHABETA_TO_ABC_PEG _V1	Unsigned Resolution: Up to 16 bit
13	ABC to alphabeta transformation	ABC_TO_ALPHABETA_PEG _V1	Signed Resolution: Up to 16 bit
14	DQ to alphabeta transformation	DQ_TO_ALPHABETA_PEG_ V1	Signed Resolution: Up to 16 bit
15	External counter	COUNTER_PEG_V1	Resolution: Up to 16 bit

16	'Terminator' IP for streaming interfaces	TERMINATOR_SRC_PEG_V 1	Resolution: Up to 16 bit
17	Data streaming IP	STREAM_DATA_PEG_V1	Resolution: Up to 16 bit
18	Parallel ADC Controller	PARALLEL_ADC_PEG_V1	Resolution: Up to 16 bit
19	Parallel DAC Controller	PARALLEL_DAC_PEG_V1	Resolution: Up to 16 bit

III.3 Software design

The software development tool (compiler, assembler, linker etc.) depends on the soft core CPU used in the design. Using the IDE for the soft CPU used, all software development tasks for the soft processor system can be performed. The software build tool flow includes the following steps to create software for the system:

1. Create a Board Support Package (BSP) for the system. The BSP is a layer of software that interacts with the development system.
2. Create the application software.
3. Iterate through one or both of these steps until the design is complete.

The control application software written in C/assembly can be ported to the processor on the FPGA with required boot loaders. The application code contains only simple and easily understandable instructions/functions. In the context of the CPU, these operations are basically memory read and memory write operations.The IPs are implemented as digital hardware and hence the speed of operation of the processor is high as hardware performs faster than its equivalent software implementation. The number of lines in the software code is also less as compared to equivalent software implementation.

III.4 Simulation

A simulation procedure is formulated for the controller design, both for hardware and software i.e., the Hardware – Software co-verification methodology. ModelSim or any other tool can be used for evaluating the hardware developed in SOPC methodology along with the application software developed in the IDE. Simulation using these tools will give exactly how the design (both hardware and software configuration) works if it is ported into an FPGA.

IV. Hardware Implementation

The controller and the power interface hardware developed is shown in Fig. 4. The controller PCB is designed as a daughter board, that can be stacked on to the interface card. This makes it possible for the controller to be used in different applications by configuring the interface PCB accordingly. The SOPC hardware configuration along with the application software is downloaded into the serial FLASH memory in the controller PCB using a JTAG programmer.

Fig.4: Controller and power stage interface hardware

V. Boost Converter Using HSRPEC

The SOPC controller along with a custom designed interface card for power stage is used in a 60V-150V DC-DC boost converter for performance comparison with a conventional DSP. Off-the-shelf DSP operating at 150M Hz consumes 6µ sec. for executing the control loop, whereas the HSRPEC operating at 100M Hz took 800n sec. for the control loop execution.

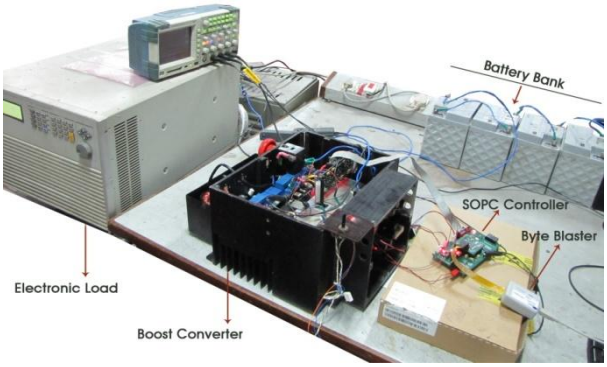

Fig.5: Boost converter with HSRPEC

A battery bank is used as the source and an electronic load is used. The control algorithm was implemented in C language with features like soft start. The waveforms of output voltage, output current and inductor current are shown in Fig.6.

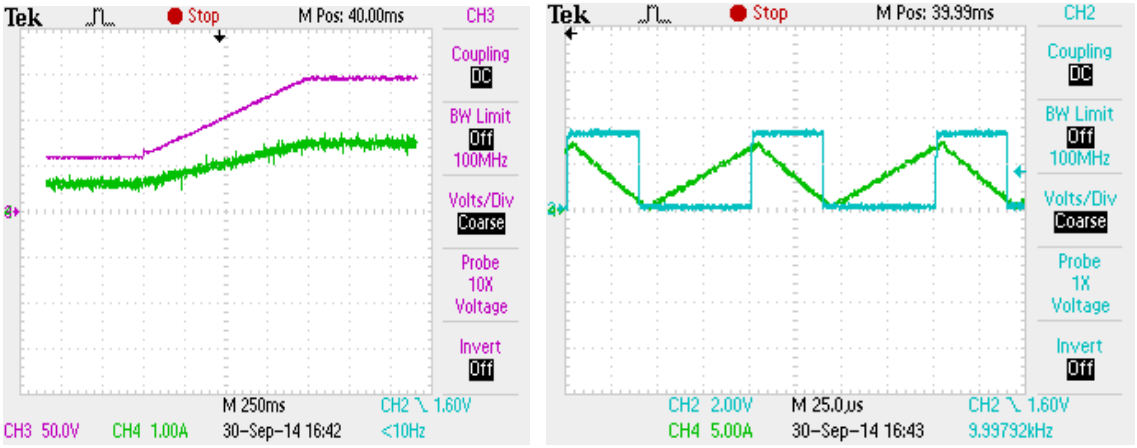

Fig.6: Load voltage and current; inductor current and PWM signal

VI. Conclusion

Power Electronics systems in railway, military etc. require system life of the order of 20-30 years. It is essential for the manufacturing industry to provide support for such systems over its entire service life. Since FPGA technology and built-in soft CPU cores have the potential to integrate system-design into a single FPGA device, they provide several advantages over conventional controller designs. The SOPC method is found to be effective for reconfigurable design and provides a good solution to processor obsolescence and related overheads. The HSRPEC developed along with power electronics specific IP is both cost effective, ensures long term support for power electronic systems and provides better performance over conventional systems, offering the designer, the space and scope for incorporating many features such as protection interlocks, which otherwise would have been impossible due to speed and resource constraints.

References

Proceedings Papers:

[1] Armando Astarloa, Unai Bidarte, Jesus Lazaro, Jon Andreu, and Jose Luis Martın, Configurable-System-on-Programmable-Chip for Power Electronics Control Applications, *International Conference on Reconfigurable Computing and FPGAs*, pp. 169 – 174, 2008

Theses:

[2] Henrik Svensson, *Reconfigurable Architectures for Embedded Systems*, Lund University, Sweden, 2008.

Proceedings Papers:

[3] H. L. Huy, Microprocessors and Digital ICs for Motion Control, *Proceedings of the IEEE*, 1994, pp. 1140-1163

[4] G. J. van Heerden and H du T. Mouton, *Design of a DSP based controller for power electronics applications, 6th Africon Conference in Africa*, vol.2,pp. 743 - 747, 2002

[5] K. Sugahara, S. Oida, T. Yokoyama, High performance FPGA controller for digital control of power electronics applications, *IEEE 6th International Power Electronics and Motion Control Conference*, pp. 1425 - 1429, 2009

[6] K.Sudha, L Kathir, Implementation of FPGA Based Controller for Induction Motor Drives, *International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC),* 2013

[7] E. Monmasson, L. Idkhajine, I. Bahri, M-W- Naouar and L. Charaabi, Design methodology and FPGA-based controllers for Power Electronics and drive applications, *Pacific-Asia Conference on Circuits, Communications and System*, pp. 360-363, 2009

[8] Jason G. Tong, Ian D. L. Anderson and Mohammed A. S. Khalid, Soft-Core Processors for Embedded Systems, *International Conference on Microelectronics*, 2006

[9] Ding Zheng, Yichen Wang and Zou Xueyi, The methods of FPGA software verification, *IEEE International Conference on Computer Science and Automation Engineering (CSAE),* 2011