American Journal of Electrical and Electronic Engineering, 2017, Vol. 5, No. 3, 85-93
Available online at http://pubs.sciepub.com/ajeee/5/3/3
©Science and Education Publishing
DOI:10.12691/ajeee-5-3-3

Power Systems Control Implementation Based on Two Dedicated FPGA Development Boards Using a Hard/soft Communication Interface

Serge Raoul Dzonde Naoussi[1,*], Charles Hubert Kom[1], Jean-Philippe Blonde[2]

[1]Lab. d'Electronique, Electrotechnique, Automatique et de Télécommunication, L2EAT, University of Douala, Cameroon
[2]Lab. des sciences de l'Ingénieur, de l'Informatique et de l'Imagerie, ICUBE, University of Strasbourg, France
*Corresponding author: serge.dzonde@univ-douala.com

Abstract This paper presents an unusual Field Programmable Gate Array (FPGA) development platform for power systems control. This platform is made up of two FPGA boards with a hardware and software communication interface. The one initially realized for direct torque Motor control is now dedicated for analog and digital signals input/output and the other card of much important density is specialized in data processing. A fully digital modules have been developed inside FPGAs from synthesis tools Quartus 2 and Maxplus 2, and a physical link have been realized to manage the communication between the cards. The control strategy takes advantage of the parallelism of the FPGA technology to share in real-time the controller's information with its environment. Simulations and experimental results validate the feasibility of a simple handshaking communication protocol between both cards. On the other hand, with the proposed implementation solution it was shown a safe operation in a power grid switching station, a successful instantaneous frequency measurements used for grid synchronization and Active Power Filter control, followed by a production of Pulse Width Modulation (PWM) gating signals used for inverter in motor control and power quality applications.

Keywords: *FPGA, communication interface, design methodology, power systems control*

Cite This Article: Serge Raoul Dzonde Naoussi, Charles Hubert Kom, and Jean-Philippe Blonde, "Power Systems Control Implementation Based on Two Dedicated FPGA Development Boards Using a Hard/soft Communication Interface." *American Journal of Electrical and Electronic Engineering*, vol. 5, no. 3 (2017): 85-93. doi: 10.12691/ajeee-5-3-3.

1. Introduction

A complete digital integration of control systems on a unique target is made possible thanks to more and more successful components and flexibility inherent to all the programmable solutions. The implementation of various control strategies is not anymore only done on microprocessors or dedicated targets such as DSP (Digital Signal Processing); but hardware reprogrammable chips as FPGA (Field Programmable Gate Array) allow today to obtain remarkable performances by combining a reduced development cost, a high density integration and convenient software tools [1,2]. The FPGA technology is actually used by numerous designers in different application fields such as telecommunications, video, signal processing and embedded control systems. In [1] a power interface for a Fuel Cell system supplying an AC load, includes a prototype of dc–dc and dc–ac converters, and all the neural control for both power converters is implemented in FPGA. This component which will focus our attention was already used successfully for electric systems control in particular the Active Power Filter (APF), one of the most advanced solutions for harmonics compensation and power factor correction. In this way different harmonics detection strategies are discussed in [3] according to FPGA implementation issue. In order to be robust to different power grid parameters variation APF control uses a Phase Locked-Loop (PLL) for phase and frequency tracking. By the same way, a full control system of a grid-connected current-controlled voltage-source inverter (VSI) has been designed and implemented on a FPGA with great performances even in abnormal grid conditions [4]. Another research work discussed in [5] presents a neural PLL based on a Voltage Control Oscillator with significant results under various utility conditions. In [6] a Direct Torque Control (DTC) for DC motor is discussed and the control algorithms are implemented in FPGA FLEX 10k100ARC240-1 chip, available on the realized board with good experimental results. The obtained modular architecture optimized both hardware amount and processing time, making easier design reuse of circuit. But in order to validate various control architectures, this solution presented a number of limitations due to the low integration density of the FPGA on board. In the attempt to provide inherent solution with minor limitations, we chose development kit based on a FPGA component of APEX family (20K200E), with more amount of resources for algorithms integration. But this solution was limited by the lack of analog inputs/outputs. Therefore we propose in this paper to develop a global platform with those two boards linked at the same time by a physical and logical interface. The main objective of the project was to exploit the various analog inputs/outputs of the first card and the high density FPGA of the second card so as to obtain a flexible and intensive computing platform for various electric systems control. Even though some of the systems involving nowadays multiple FPGAs can present high speed parallel, stable, accurate and reliable interface [7,8,9], we intend in this work to obtain a more flexible platform with a simple communication interface based on a handshake protocol.

This paper is organized as follows: Section II presents FPGA description and an overview of related design methodologies as well as their development environment. Then, an introduction to the chosen design methodology will be made with the hardware design flow. In Section 3, we discuss the development of the internal architecture for both FPGAs, FLEX and APEX in order to obtain the newly implementation solution. Henceforth, a model of asynchronous communication is presented and its performances evaluated and validated. Section 4 presents simulation results for functional validation of the proposed platform. In section 5 three applications on power systems control implemented on this dual board FPGA development solution are presented, followed by Section 6 which gives the main conclusions of the paper.

2. FPGA Based Design and Its Related Development Tools

2.1. Description of the FPGA and Hardware Description Languages

2.1.1. Description of the FPGA

FPGAs are fast becoming the digital processors of choice for implementing computationally intensive algorithms in real time due to their high clock speed and inherently parallel hardwired architecture [10]. This specificity allows significantly reduced time-to-market and complexity as well as offering more flexibility. That is why the designers of Application-Specific Integrated Circuits (ASICs) often use FPGAs as prototyping targets although the languages used in both cases are quite the same. However, until recently, their application has been limited to the development of digital control algorithms and Pulse Width Modulation (PWM) gating pattern generators, mainly due to lack of large device capacity, and intellectual property cores for the modeling of complex power electronic systems. These components do not stop developing by benefiting from advances in the field of microelectronics. Nowadays, components as Stratix 10 EP3SL340 offers up to 338.000 logical elements and a memory capacity of 16 Mbits [11]. The figure 1 presents the generic architecture of a FPGA. More recently, we knew inside these architectures, the introduction of dedicated blocks such as RAM, DSP, interfaces PCI and even hearts of processors so allowing a high-level programming and an optimization of the integration for the implemented parts.

2.1.2. Hardware Description Languages

Today, the FPGA is widely used to implement more and more complex functions such as Arithmetical and Logical Units (ALU), memories, communication units, etc. This evolution is due to the development of tools and design methods which refer to the integration technologies of the large-scale circuits. These tools are based for the greater part on the Hardware description languages (HDLs) such as the Very high integrated circuits Hardware Description Language (VHDL) or Verilog [10,12,13]. The standardization of the VHDL language by the Institute of Electronics and Electrical Engineers (IEEE) generalized its use by allowing the creation of various Computer-Aided Design (CAD) tools in the field of the microelectronics. In this way, this language can be used for modelling, simulation and logical synthesis to build a personalized circuit by a modular and hierarchical approach defined at every level of abstraction. The level of abstraction corresponds to the degree of technological independence of the design with regard to a circuit. Besides, the language C comes to complete the list for FPGA circuits with heart of processors and corresponds to the highest level of abstraction.

2.2. Contribution and Limits of the FPGA for the Control of Electrical Systems

2.2.1. Domain of Use of the FPGA

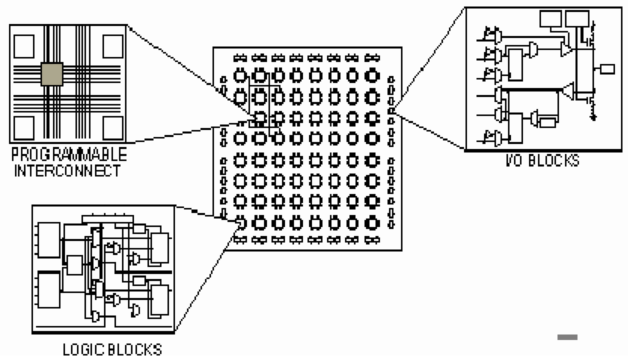


Figure 1. Generic Architecture of a FPGA

In a design process, the following criteria are to be considered: the cost, the energy consumption (essential in the case of the embedded systems), the experience and the brand image of the manufacturer of the chosen target, its capacity in terms of available material resources for implementations. Indeed it is described in [14] the compounding effects power consumption has on system design, including bill of material cost and how the Artix-7^{TM} family from Xilinx addresses these challenges for a variety of applications. The time constraints of the algorithm correspond to the type of dependence on the data. The more the dependence is big, the more the algorithm is sequential [6]. In this case it is obvious that the DSP software solution is the most appropriate. If on the other hand, the program reveals several possibilities of parallelism (low dependence on the data and the conflicts between operations), then the material solution FPGA becomes the most interesting.

However, the time constraints are not sufficient to characterize an algorithm. His complexity estimated in term of number of operations and their regularity, is also a key element. In certain cases, the DSP is preferred for a historic reason because the design process is based on programming. However, the progress in designing methods and CAD tools nowadays makes more and more the balance on the side of the FPGA. By using the VHDL, the portability of the codes is assured, nevertheless a graphical interface as Matlab/Simulink with Altera Dsp Builder^{TM} block set facilitates the design, reducing time-to-market.

2.2.2. Dynamic Reconfiguration of FPGA

The contribution of FPGAs in electrical systems control is not limited compared to the DSP. The possibility of dynamic reconfiguration of the material architectures which offer some FPGAs although still under exploited is of a major interest. It also allows to assure continuity in the control or at least the reset of a process in safety conditions [6].

2.3. Design Methodologies

The digital technology is less sensitive to the ageing and to the thermal constraints in comparison to the analog technology. To benefit from advantages of FPGA and performances of the CAD tools, several design methods are proposed:

- The Top-Down method

The top-down approach goes from the system to its circuits and then the sub-circuits towards the scheme made up of transistors. At every level of abstraction is a corresponding model.

- The method based on Reusable Blocks

A reusable block also called IP is a functional element of a library which the designer can directly use or instantiate. It can be hard (already physically integrated, dependent on the technology and very optimized) or soft (described in high-level language such as VHDL or Verilog and often customizable and synthesizable). The efficiency of a Reusable Blocks method is evaluated by intuitive criteria such as: the selection, the specialization and the integration of the chosen blocks [13].

- The method of modular Design

It consists in partitioning a design problem by organizing it into a sum of functions enough independent and generic to obtain functional modules if possible reusable [15].

2.4. Hardware Design Flow

The design to implement on FPGA can be obtained directly in Quartus 2^{TM}. But it is easier to develop the model in discrete form on Matlab/Simulink^{TM} before converting it by using functional blocks from Altera DSP Builder^{TM} (ADB) library. Particularly, by using a modular design combined with Top/Down integration process, the design flow follows those steps:

- Conversion of the design into a VHDL program;
- Analysis and logic synthesis;
- Architectural study and RTL synthesis according to Quartus 2 or equivalent software;
- FPGA programming.

3. Modelling of FPGAs Internal Architecture on the Proposed Platform

3.1. FLEX Internal Architecture

The implemented platform is based on a communication of type Master - Slave where the FLEX card (slave) is being specialized for the acquisition of digital/analog signals, and the generation of those coming from the FPGA APEX via the Interface. On the other hand, the Excalibur Kit (master) is for data processing. The architectural model of logical interface implemented inside both FPGAs is shown in Figure 2. It shows up the following blocks inside the FPGA FLEX dedicated for the communication:

- Conv12to16: assures the conversion of 12 bits getting out of the Analog Digital Converter (ADC) to the 16 bits data format used during the exchange.
- MUX6x1: is a multiplexer 6 inputs and 1 output.
- FLEX Interface: allows the card with FLEX target to communicate with the Excalibur kit.
- Out_Select: chose the appropriated output between the driver exit and the Digital Analog Converter (DAC)
- Gen_Clk: generates the clocks signals to synchronize Analog Digital Conversion.
- Conversion Module of 16/12 and 16/6 bits: is made up of a conversion block 16/12 bits to adapt the data bus size to the DAC format and a 16/6 conversion block towards the driver exit.

The proposed FLEX interface algorithm is based on a finite 6-states machine presented in Figure 3, with 3 major states: ADC conversion, reception from the APEX and emission towards the same chip. The variables used for transition between states correspond to the wires of the communication interface.

3.2. APEX Internal Architecture

As shown in Figure 2, the following blocks should be available inside the FPGA APEX to handle the process:

- MUX4x1: allows to select according to the code LdMux, the data input to be loaded into the registers
- APEX Interface: allows the Excalibur kit to communicate with the FLEX card.
- Register Blocks: it is a bank of 8 16-bits registers corresponding to the data from 4 ADC and 4 digital inputs/outputs. The writing is possible only on a register at the same time through the 3 to 8 Decoder. The register identified by the address AddIn is loaded while synchronized by the basic clock Clk and validated by the signal We.
- Control Block: contains the VHDL code for sequencing of the process and eventually an ALU for the arithmetic and logical operations.

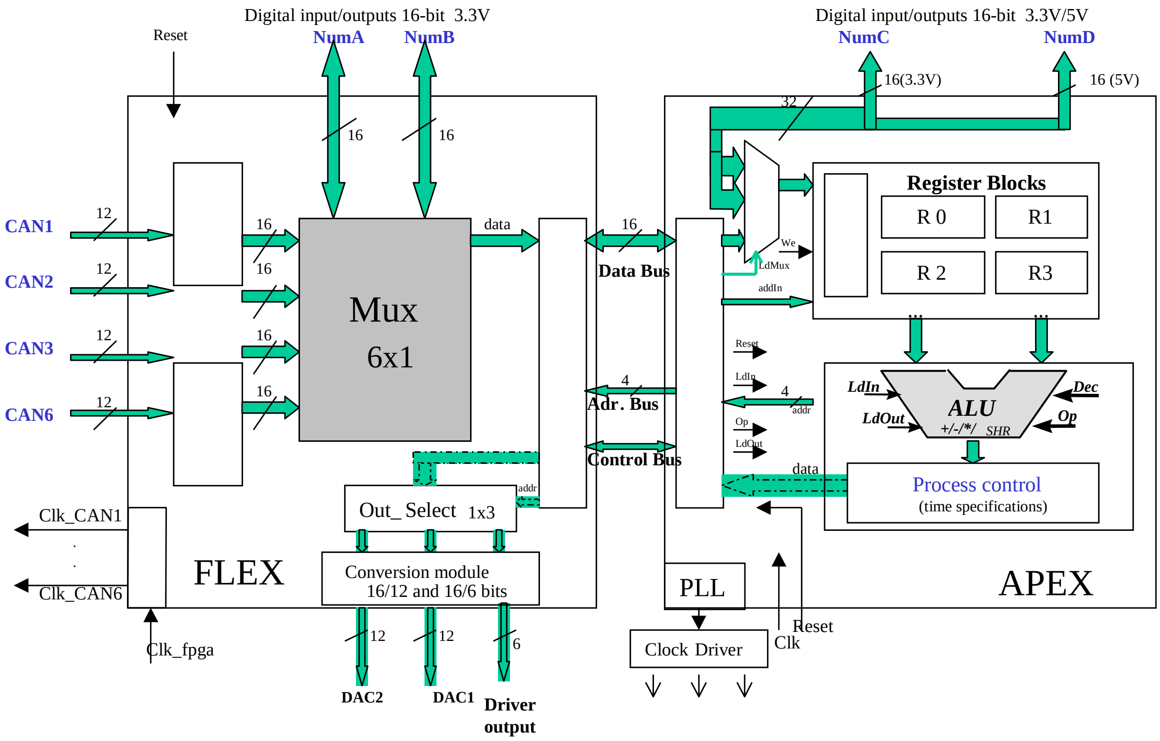


Figure 2. Modular Architecture of the implemented communication system

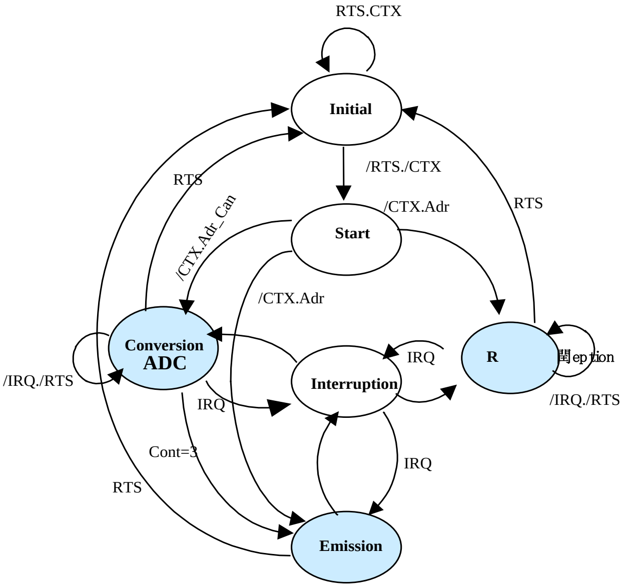


Figure 3. State machine of the interface model on the FPGA FLEX side

3.3. Communication Interface between FLEX Card and Excalibur Kit

All the signals are transmitted the ones beside others in a 40-pin Integrated Drive Electronics (IDE). It is therefore a parallel transmission which drives a 0-3.3V voltage compatible with the standard digital voltage level of both cards. In Figure 4 each wire's name and function on the physical interface used on the proposed platform is presented. The half-duplex exchange is made possible by the 3-states buffer configuration of the FPGAs data bus pins. Its principle is shown in Figure 5.

A 3-states port works as follows:

- Control input at state 0: it behaves as an opened circuit. Henceforth, its output can be forced to 0 or to 1 by another circuit (FLEX pin on a reception mode)
- Control input at state 0 (low impedance state): it behaves as short-circuit, the output is directly connected to the input (FLEX pin on a transmission mode).

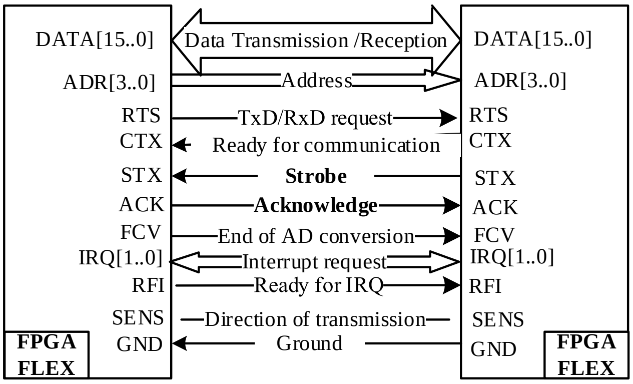


Figure 4. Asynchronous link between FPGAs FLEX and APEX

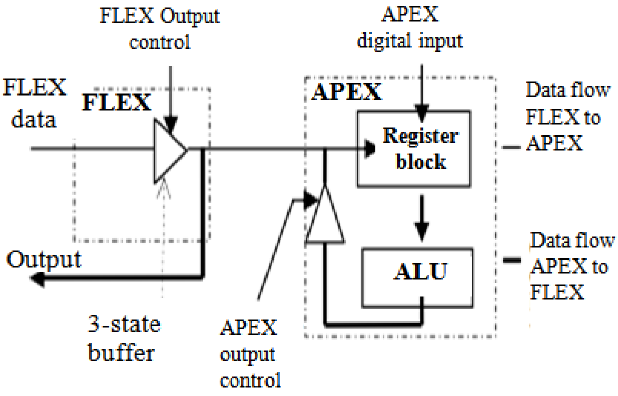


Figure 5. Principle of the bidirectional exchange between FPGAs FLEX and APEX of the Excalibur kit

4. Simulation Results for Functional Validation of the Proposed Platform

We present here an example of asynchronous communication between both FPGA boards.

4.1. Simulation of the Algorithm inside the FLEX

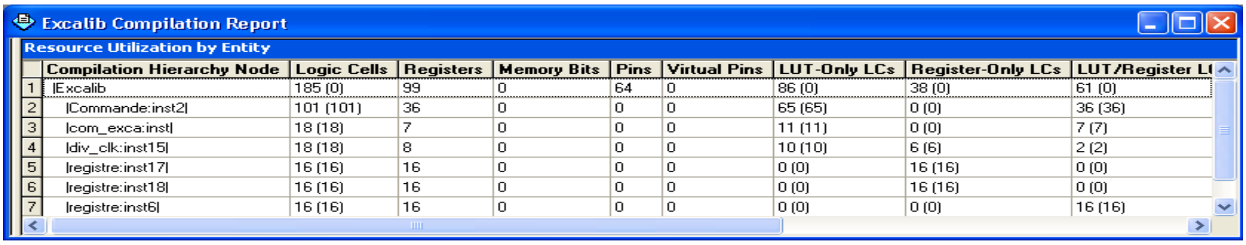
First of all, we present in figure 6 the VHDL code represented in the graphic editor of the software Maxplus 2. On this side, the algorithm has been presented in a simple form. The block Inter_FPGA particularly represents the interface module and Bustri12, the 12-bits 3-state buffer. The input/output values of FPGA FLEX are presented in the hexadecimal system of numbering. We notice that the ADC inputs and the DAC outputs are of 12 bits format while the digital input/output are on 16 bits. The simulation results of Figure 7 presents the 2 main functioning phases of this FPGA board.

1. Acquisition of the data "AAA" coming from the analog input CAN1 of the FLEX (address "0011") then effective loading to the APEX as soon as STR passes in "0"
2. Broadcast of the data previously recorded in a register of the APEX towards the analog output CNA1 (address '0001') of the FLEX.

4.2. Simulation of the Algorithm inside the APEX

The development of the VHDL code and the graphic model named Excalib was done under Quartus 2 software through its Graphic Editor. The implementation inside the APEX 20k200E highlights 19 registers, 1 clock divider, 1 3x1 multiplexer, 2 8x1 multiplexers, 1 control block, 1 3x8 decoder and a particular Interface block itself. The details on the resources utilization is presented in Table 1. In summary, 185 logical elements on 8.320 (2 %), 64 input/output pins on 376 (17 %) have been used, what corresponds to low rate of use.

Table 1. Resources consumption in the Apex for the chosen example



Excalib Compilation Report

Resource Utilization by Entity

	Compilation Hierarchy Node	Logic Cells	Registers	Memory Bits	Pins	Virtual Pins	LUT-Only LCs	Register-Only LCs	LUT/Register L(
1	|Excalib	185 (0)	99	0	64	0	86 (0)	38 (0)	61 (0)
2	|Commande:inst2|	101 (101)	36	0	0	0	65 (65)	0 (0)	36 (36)
3	|com_exca:inst|	18 (18)	7	0	0	0	11 (11)	0 (0)	7 (7)
4	|div_clk:inst15|	18 (18)	8	0	0	0	10 (10)	6 (6)	2 (2)
5	|registre:inst17|	16 (16)	16	0	0	0	0 (0)	16 (16)	0 (0)
6	|registre:inst18|	16 (16)	16	0	0	0	0 (0)	16 (16)	0 (0)
7	|registre:inst6|	16 (16)	16	0	0	0	0 (0)	0 (0)	16 (16)

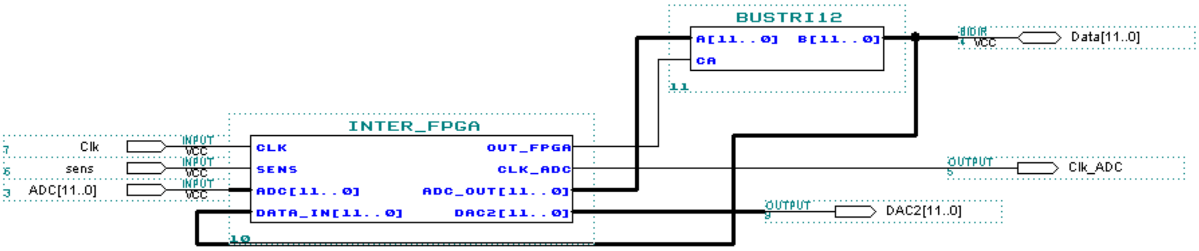
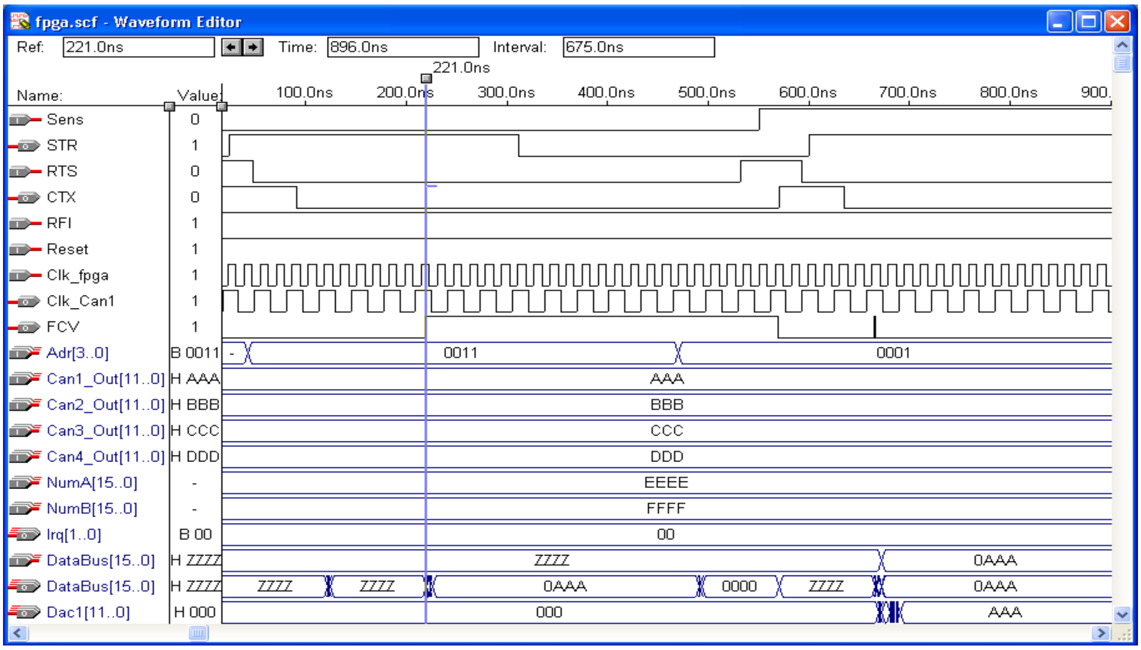


Figure 6. Graphical representation of the FLEX interface code under MAXPLUS 2

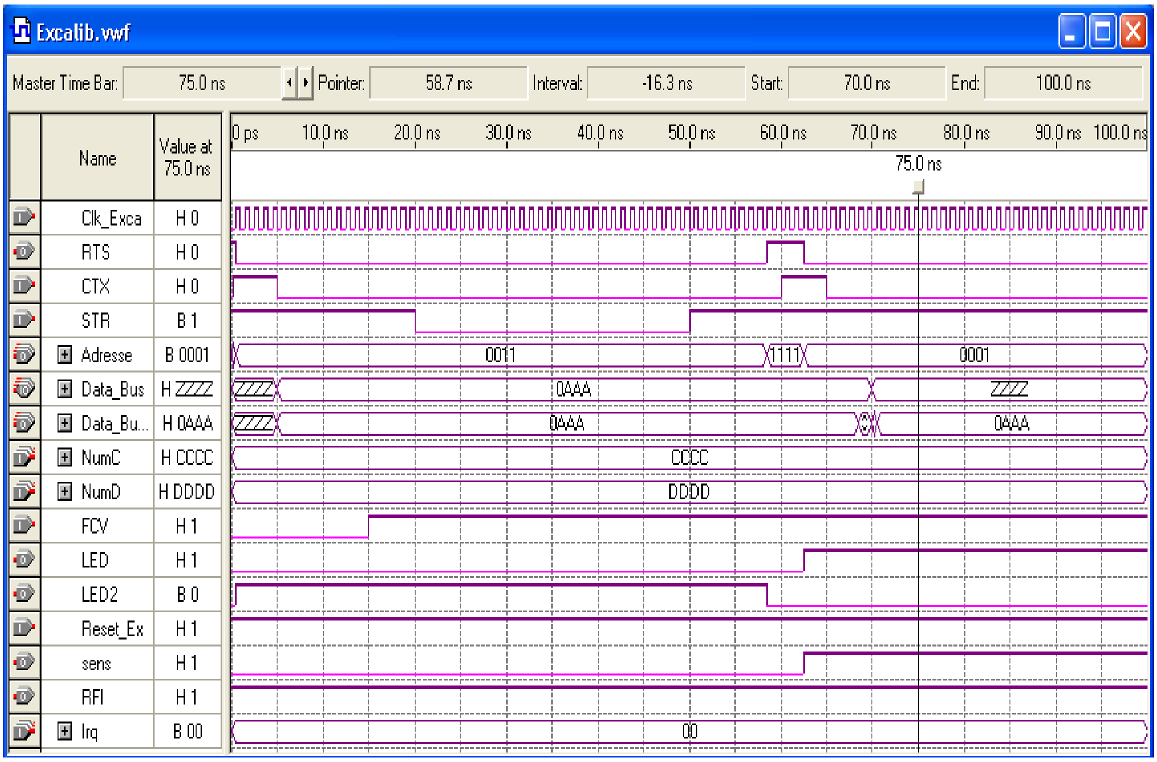


fpga.scf - Waveform Editor

Ref: 221.0ns Time: 896.0ns Interval: 675.0ns

Name:	Value:
Sens	0
STR	1
RTS	0
CTX	0
RFI	1
Reset	1
Clk_fpga	1
Clk_Can1	1
FCV	1
Adr[3..0]	B 0011
Can1_Out[11..0]	H AAA
Can2_Out[11..0]	H BBB
Can3_Out[11..0]	H CCC
Can4_Out[11..0]	H DDD
NumA[15..0]	-
NumB[15..0]	-
Irq[1..0]	B 00
DataBus[15..0]	H ZZZZ
DataBus[15..0]	H ZZZZ
Dac1[11..0]	H 000

Figure 7. Behavior of the FLEX during the asynchronous transmission



Excalib.vwf

Master Time Bar: 75.0 ns Pointer: 58.7 ns Interval: -16.3 ns Start: 70.0 ns End: 100.0 ns

Name	Value at 75.0 ns
Clk_Exca	H 0
RTS	H 0
CTX	H 0
STR	B 1
Adresse	B 0001
Data_Bus	H ZZZZ
Data_Bu...	H 0AAA
NumC	H CCCC
NumD	H DDDD
FCV	H 1
LED	H 1
LED2	B 0
Reset_Ex	H 1
sens	H 1
RFI	H 1
Irq	B 00

Figure 8. Simulation of the asynchronous transmission from the APEX side

The simulation results of Figure 8 shows the following steps

- 1st step « 0 – 5ns »: After the acquisition request through CAN1 (address 3 in decimal) by the APEX, both FPGAs are in initial state; the data link is in high impedance state.
- 2nd step « 5 – 15ns »: The FLEX is at the Conv_Can or latency state (CAN is waiting 3 clock cycles before the 1st acquisition)
- 3rd step « 15 – 20ns »: the corresponding digital value of the analog signal from CAN is on Data_Bus (‘0AAA’) but is waiting a low state on STR before being loaded inside the APEX. FCV changes to high state and then the FPGA FLEX is on transmission mode.
- 4th step " 20 - 50ns ": Loading state; data acquired are loaded in the register R0 which is dedicated to the CAN1
- t=58ns: the APEX requests the suspension of the acquisition (RTS = 1)
- t=60ns: FLEX takes into account this suspension by putting CTX to 1
- t=62.5ns: the APEX asks to send out the acquired data from the CAN1 towards the output CNA1 of address (0001) on the FPGA board.
- T=65ns: FLEX puts CTX to 0 to acknowledge this request
- t = 71ns: Beginning of the transmission as the control gate of the output 3-state buffer changes to 1. The LED is on meaning that the APEX is in transmission mode.
- t = 100ns: End of the simulation.

NB: the digital inputs of the APEX, NUMC and NUMD have respectively the values ('CCCC') and ('DDDD') in hexadecimal. After succeeding the simulations, we began FPGA programming before the functioning test.

4.3. Functioning Test of the System

Before proceeding to the test of our system, we defined first of all specifications with the aim of being able to observe the progress of the exchange which is made in three main steps. By taking as reference the Excalibur kit, we have:

1: Acquisition CAN1 during t_{ac}
2: No operation during t_{rp}
3: Broadcast of the value acquired towards CNA1 during t_{em1}
4: No operation during t_{rp}
5: Acquisition CAN2 during t_{ac}
6: No operation during t_{rp}
7: Broadcast of the newly acquired value towards CNA1 till the end of the test

All these values can be founded in the Table 2.

Through clock frequency divider block i.e div_Clk inside both targets, we set up the parameters f_{FLEX} and f_{APEX} (operating frequency of FPGAs FLEX and APEX respectively). Besides, the time t_{dr} which FLEX spent to inform the APEX that the present data on the bus are stable was specified. The hardware implementation was successively made on the FLEX and the Apex and therefore the exchange was able to be done. Figure 9 shows the experimental setup with both FPGAs boards. With -2V and +5V at two ADC inputs of the FLEX card, we obtained the same voltages on a two-channel scope. In fact, the corresponding digital values were sent by the FLEX inside the APEX before returning to the FLEX to exit through the DACs, connected to the scope. By succeeding this bidirectional exchange, it was now possible to use this implementation platform to test various applications in power systems control.

Table 2. Specifications for the tests of functioning of the parallel interface

Parameters	f_{FLEX}	f_{APEX}	*tdr*	*tac (ADCs)*	*trp*	*tem1*	*tem2*
Value	500khz	1Mhz	1.5µs	5s	1s	5s	Until 'end test'

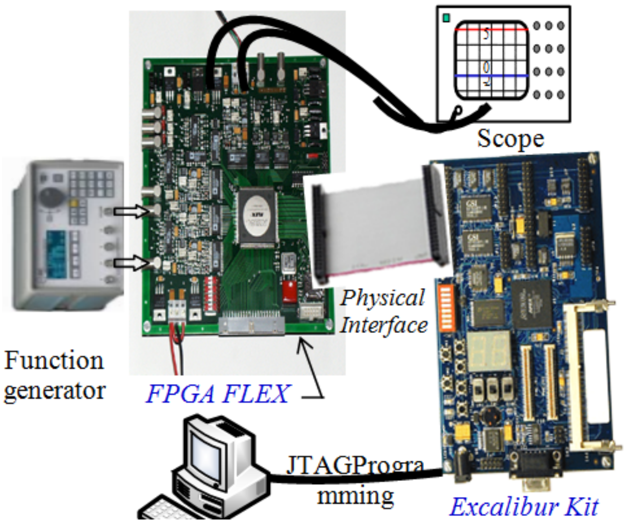


Figure 9. Experimental setup with FPGAs FLEX and APEX boards

5. Application of the Proposed Development Platform in Power Systems Control

5.1. Switching Operations on Two 225KV Busbars in an Electric Substation

It is shown in Figure 10 an electric substation with two busbars at 225KV including transformers to change voltage levels between this high transmission voltage and lower voltage (90KV). The coupling circuit breaker is the transformers are linked on the busbar A, i.e. the switches 2.a and 2.b. are respectively closed and opened. In order to change a departure supply from the busbar A to B we just have to:

- close the coupling circuit breaker (1)
- close the 2nd switch (2b) of the corresponding departure
- open the first one (2a)
- open the coupling circuit breaker (1).

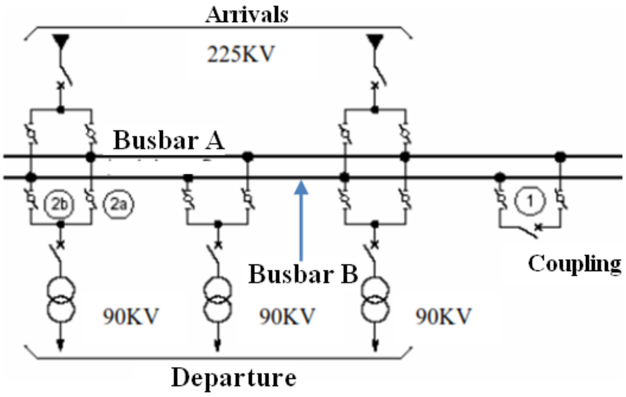


Figure 10. An interconnection station VHV/HV with two busbars

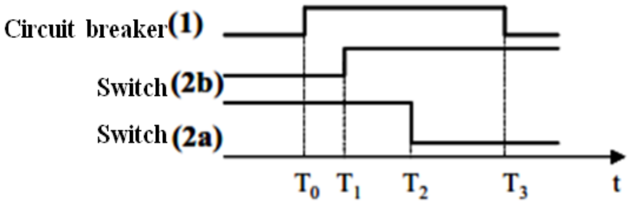


Figure 11. Control signals for breaker (1), switches (2a) and (2b) during a change of departure supply

The departure is then powered by the other busbar B. It should be noted that during the permutation time, all the arrivals are connected in parallel on both busbars; the power of short circuit is then brought up and the electric characteristics of the material must be sufficient to satisfy this mode of operation in case of high probability of fault appearance. In Figure 11 the control signals for the circuit breaker and the 2 disconnects are shown.

5.2. Producing PWM Switching Signals to the Gates of a VSI

PWM technique can be used for VSI control in various applications as APF and Motor control. In this way, a reference analog signal is compared with a triangular wave to produce gating signals for the inverter.

As shown in Figure 12, a PWM signal *Sc* and its opposite are obtained for switching power devices of the same line inside the VSI. By modeling PWM algorithm and adding it to the APEX-side interface model, both of them were implemented inside the APEX. After doing the same operation with the FLEX-side interface model, we use sine and triangular waves from a low frequency generator connected to the FLEX board. From the same card, after PWM processing inside APEX, we obtained the gating signals shown in Figure 13, presented with the sine wave signal. According to the amplitude of the chosen sine wave, i.e. 4Vp-p, it is shown in figure 13.(a) an interrupted PWM signals. In Figure 13.(b), with a 2.24 Vp-p amplitude, the produced signals are no longer interrupted and can then be used in VSI control with better results.

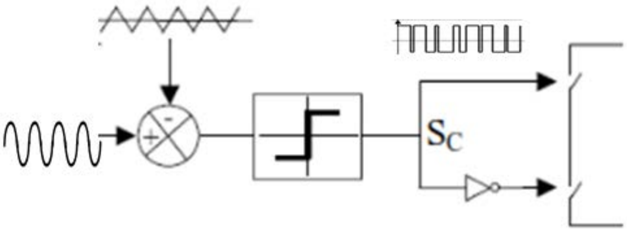


Figure 12. Basic principle of the PWM signals production

5.3. Real-time Frequency Measurement for Power Systems Synchronization

After modelling a PLL in Matlab/Simulink/Altera Dsp Builder^{TM}, as the one detailed in [4], the corresponding structure shown in figure 14 is made up respectively of a symmetrical components ($v^d_{sa}, v^d_{sb}, v^d_{sc}$) extraction from the three-phase voltage system ($v_{sa,}$ $v_{sb,}$ v_{sc}) and a frequency (*f*) estimation modules. The first one is based on a Voltage Control Oscillator (VCO) and uses Concordia/Park Transform for (d-q) to (a,b,c) spaces conversion, its inverse and a Low Pass Filter (LPF). The second one is an ADALINE-based PID model which produces the instantaneous frequency of the three-phase power grid from the first weight, $w_1(k)$, obtained after the learning process of the chosen symmetrical component, i.e v^d_{sa}.

From a real test bench, we connected the three-phase source voltage on analog inputs of the FLEX card which already contains its interface module. On the APEX side, the PLL for frequency measurement is added on the interface module before the implementation. In this way, the instantaneous frequency can be shown either on the seven-segment display of the APEX card and also on a scope connected on an analog output of the FLEX card.

In Figure 15 the source voltage obtained from a real test bench, and delivered by three transformers operating as sensors, is shown (Figure 15.(a)) as well as the corresponding instantaneous frequency (Figure 15.(b)) produced by the PLL. We then notice that in any case this frequency value measurement is between 48.8 and 50.2 Hz. On the other hand, we cannot have its precise value through the seven segment display because only two digits are available instead of four, two to display the integer part, one for the decimal point and one for the digit after the point.

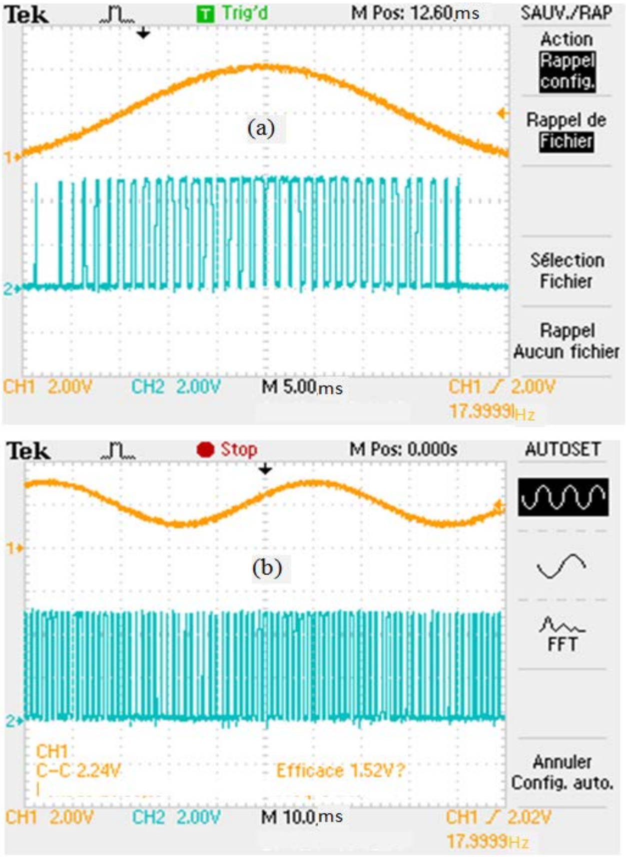


Figure 13. Experimental results (PWM signal) obtained from the comparison of an 18 Hz sine wave and 800 Hz triangular wave: (a) 4Vp-p) sine wave, (b) 2.24Vp-p sine wave

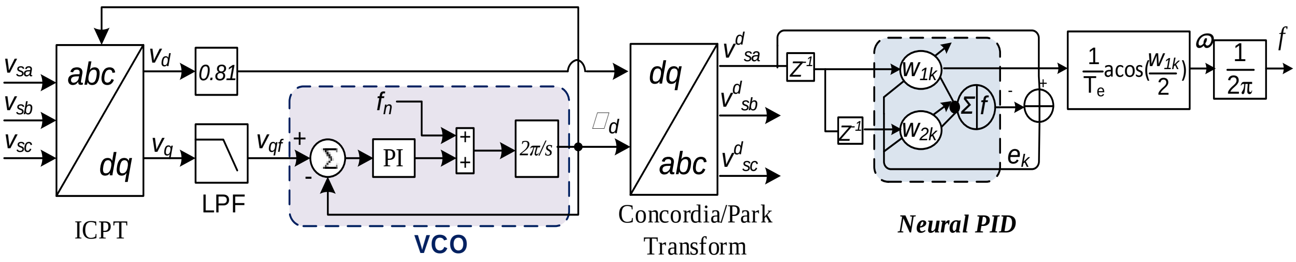


Figure 14. Measurement principle of the instantaneous frequency of a three-phase power system

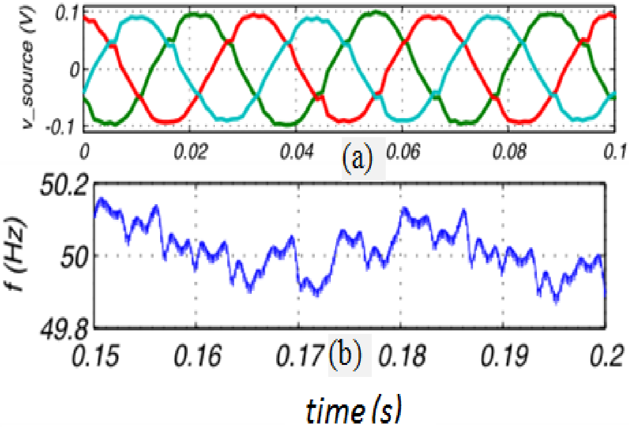


Figure 15. Real-time frequency measurement by the PLL: (a) the three-phase source voltage, (b) the instantaneous frequency of the source voltage

6. Conclusion

The purpose of this paper was the development of a flexible solution for hardware implementation of power systems control architectures. The proposed platform is constituted by a FPGA board with FLEX10k100A target combined to a development kit called Excalibur with FPGA APEX 20k200E on board, with more amount of resources.

We first of all described the FPGA and the related design methods based on hardware description languages. Afterward, we presented the implementation of the Interface between both cards. In that way, according to the criteria of performance, reliability, minimization of the material resources and flexibility, we realize a parallel and asynchronous transmission of up to 16 MBps. The proposed implemented algorithms in both targets allowed to specialize the FLEX card particularly in acquiring analog signals and in generating this type of signals outside the same card after data digital processing inside the FPGA APEX on the other board. Simulation and experimental results illustrate the adaptation of this hardware integration technique for power systems control. In fact, we show how switching operations on two 225KV busbars in an electric substation can be done. We also present an instantaneous frequency measurements used for grid synchronization or in adaptive active filtering. The third application concerns the production of PWM signals used for a VSI control which can also be related to power quality or to motor control.

In future work, we intend to use the platform to implement more intensive algorithms with a parallel processing approach done with both FPGAs functioning as dual core processor. The main objective of this technic is to improve the general performance of a particular power systems controller, after a subsequent logical partitioning and an adequate specialization of FPGA chips.

References

[1] Cardenas, A., Agbossou, K., and Henao, N., Development of Power Interface With FPGA-Based Adaptive Control for PEM-FC System, IEEE Transactions On Energy Conversion, Vol. 30, No. 1, March 2015.

[2] Myaing, A., and Dinavahi, V., FPGA-Based Real-Time Emulation of Power Electronic Systems With Detailed Representation of Device Characteristics, IEEE Transactions On Industrial Electronics, Vol. 58, No. 1, January 2011.

[3] Dzondé Naoussi, S. R., Kom, C-H., Berviller, H., Blondé, J-P., Flieller, D., Kom, M., and Braun, F., Neural Harmonic Detection approaches for FPGA Area efficient Implementation, European Physical Journal – Applied Physics, 56, 30901 (2011).

[4] Nguyen, N. K., Flieller, D., Wira, P., Ould Abdeslam, D., Neural networks for phase and symmetrical components estimation in power systems, 35th Annual Conference of the IEEE Industrial Electronics Society (IECON09), Porto, Portugal, November 3-5, 2009, CD-ROM.

[5] Ekström R., and Leijon M., FPGA Control Implementation of a Grid-Connected Current-Controlled Voltage-Source Inverter, Journal of Control Science and Engineering, Vol. 2013 (2013), Article ID 713293, 10 pages

[6] Monmasson, E., Chapuis, Y.A., « Contribution of FPGA to the Control of Electrical Systems, a Review», IEEE Industrial Electronics Society Newsletter, p.8-15, March-June 2003

[7] Melo L., Santana S., Silva-Filho A., Lima M., Medeiros V., and Marinho M., An inter-FPGA communication bus with error detection and dynamic clock phase adjustment, Journal of the Brazilian Computer Society, (2015) 21:12

[8] Wu A., Jin X., Du X., Guo S., A Flexible FPGA-to-FPGA Communication System, ICACT Transactions on Advanced Communications Technology (TACT) Vol. 5, Issue 3, (2016), 836-843

[9] Zhang D., Pan Y., and Hu X., Design of High-Speed Parallel Data Interface Based on ARM & FPGA, Journal of computers, vol. 7, no. 3, (2012), 804-809

[10] Camposano, R., MacMillen, D., «Design Technology for Systems-on-Chip», proceeding of 11th International Conference, on Very Large Scale Integration of Systems-on-Chip (VLSI'01), Kluwer Academic Publisher, May 3-5, 2001.

[11] Stratix 10 Advance Information Brief, 2016. Available on line at: www.altera.com.

[12] Riesgo, T., Torroja, Y. and Dela Torre, E., "Design Methodologies Based on Hardware Description Languages", IEEE Transactions on Industrial Electronics, Vol. 46, (1999), 3-12.

[13] Dutrieux, L., Demigny, D., «Logique Programmable - Architecture des FPGA et CPLD - Méthodes de Conception – Le Langage VHDL », Edition Eyrolles, p.157, 1997

[14] Mohsen, E., Reducing System Power and Cost with Artix-7 FPGAs, WP423 (v2.4) September 27, 2016: Available on line at www.xilinx.com.

[15] Kebbati, Y., Girerd, C., Chapuis, Y.A, and Braun, F., "Advances in FPGA/ASIC digital integration solutions for vector control of motor drive" The International Power Electronics and Converter conference (IEEE IPEC'2000), Tokyo, Japan, (2000), 1177-1182.