

ARCHITECTURE OF A FPGA-BASED CONTROL OF A STATIC POWER BUCK CONVERTER

Yuzo Iano, Leandro Lima

FEEC DECOM

Universidade Estadual de Campinas (Unicamp)

yuzo@decom.fee.unicamp.br, lblima@decom.fee.unicamp.br

Abstract – A methodology for the development of a digital controller for a static power converter is presented in this paper, including the behavioral model of the analog part of it. The digital controller is realized using hardware description language VHDL on FPGA, and the analog part is modeled using the Matlab Simulink. The integration of the whole structure – converter and control system - allows that the control system may be simulated, tested and validated with no need of a prototype.

Keywords: FPGA, VHDL, Simulink, Functional Verification

1. Introduction

This work shows the design of a digital controller for an "SPC" using the hardware description language VHDL on FPGA, and its interaction with the analog model made on Simulink. We will be presenting all the architecture and the digital blocks used during design on FPGA.

A top level diagram block of the whole structure is presented, shown on Figure 1.

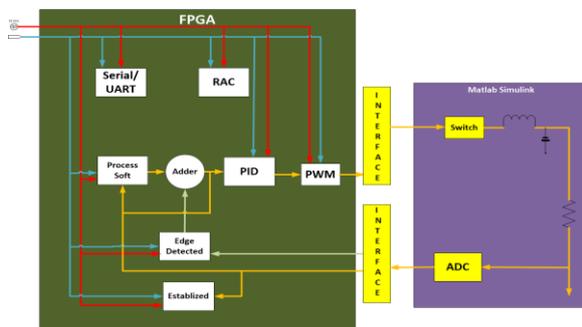


Figure 1. Static Power Converter Diagram

Due to an implementation decision this work was performed in 3 (three) parts: the first part is the architecture and design of the digital controller on FPGA, the second part represents the interface between FPGA and Simulink, and the third part represents the modelling of the static power converter with Simulink.

2. FPGA Digital Controller

The FPGA adopted was the Xilinx Spartan 3E-500, which contains 500K system gates and 10.476 equivalent logic cells, what is more than enough for this work purpose.

2.1. Sub-Blocks

Depending from the signal received from serial interface, the digital controller block generates a signal to the analog part, and keeps its signal as stable as possible using a PID, the figure below was extracted from a schematic design that illustrates the PID control and PWM has on the system, shown on Figure. 2.

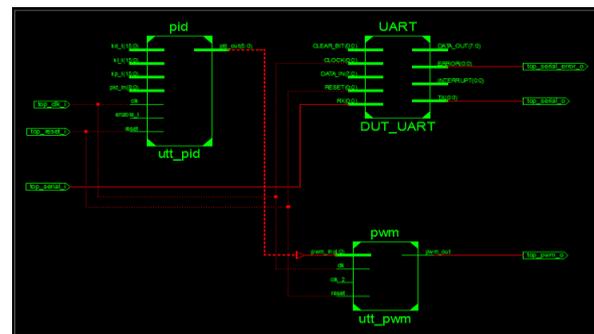


Figure 2. Schematic of Control Power Converter.

The ADC converter sends its output to the adder, and this device does a comparison between the serial input and the ADC output. The comparison result is sent to PID, in order to generate a correct duty-cycle to keep the output voltage in the required operation range, using for achieve this PWM.

The Digital Control block, implemented using FPGA, is based on three blocks: configurable logic blocks (CLB), Input Output blocks (IOB) and interconnection switches.

These blocks are, usually, made using NOR and NAND logic ports.

The Digital Controller was developed using VHDL – VHSIC Hardware Description Language. Using this language, it is possible to

describe digital electronic circuit's behavior and turn it on a physical circuit.

2.2. Serial Interface

The sub-block serial interface, known as Universal Asynchronous Receive/Transmitter is a circuit that sends parallel data through a serial line [2].

These lines receive 8 bits of data and 1 parity bit, and, in case of an error, a high logical signal is sent to signalize the error. The data is received bit to bit, and saved into an intern read-only register, through RAC (Register Access Controller). Whenever an 8 bit data is received with no error an interruption signal is sent to RAC in order to make it ready for the next 8 bits. UART needs to wait RAC clear bit, signaling that UART can read another data, once the last one received has no error.

The UART block is also capable to transmit intern data from ADC, PID, ADDER and PWM, and in order to read data from these blocks there is a set of opcodes, which one corresponding to one internal device. Internally, UART has a Baud Rate, with a 9600 bits per second to TX, and 153600 bits per second to RX.

As a security enforcement, the data received is verified, using an algorithm that compare the data bit to bit, in order to avoid an invalid value caused due to a glitch, for example.

2.3. Register Acces Controller

RAC sub-block is responsible to program PID coefficients and the input voltage value (VIN), this feature is necessary to become RAC reprogrammable. Table 1 show opcodes addresses.

Address	Name	Type
10100000	Vin	Write
10100001	Ki	Write
10100010	Kp	Write
10100011	Kd	Write
10100100	ADC	Write
10100101	PID	Write

TABLE I. TABLE OP-CODE ADDRESSES

RAC receives from UART the opcode, and it is decoded to select the respective register to be written. After that, the data is sent to be written into the corresponding register.

In order to configure the PID coefficients, it is necessary to receive 16 bits, but RAC operates receiving only 8 bits each time, then, it receives at first 8 bits and after that the last 8 bits.

As the coefficients are float numbers, the higher 6 bits describes the integer part and the lower 10 bits describes the fractional part.

Name	Definition
Vin	Voltage value desired, program the Soft Process.
Ki	PID Integral Coefficient
Kd	PID Derivative Coefficient
Kp	PID Proportional Coefficient
ADC	Sends ADC data to UART
PID	Sends PID data to UART

TABLE II. TABLE DEFINITIONS

2.4. Soft Start

Soft Process is a sub-block responsible to reduce the difference between data from UART and ADC. On this way, voltage levels are created until it achieves the desired value.

This block implements, using ADC output and serial output as input parameters of an adder, a controller used to reduce the difference of voltage between them. The control is used to regulate the analog device current gain.

The result from adder is the control signal to adequate the step value to increase or decrease the serial signal sent to adder [3].

When the adder result is stable during 10 clock cycles, the ADC is increased or decreased by 0.66 V and the value is sent to Soft Process block.

The increment or decrement are controlled by an intern stabilizer block, this block sends an update signal when adder output is stable during one clock delay time.

This operation is repeated while the voltage is not the desired one and the circuit is not stabilized. For that block the voltage variation marge allowed to considering it stable is 0.1 V. Then, Soft Process updates the values until it achieves a near range when compared to the required value. It is near because PID block add small margin of error during calculations.

2.5. Stabilizer

The stabilizer sub-block has the function to analyze the input signals and send a trigger when the circuit has a stabilized voltage.

The main function is the comparison between input and output voltage. When the sampled value is equal, considering the margin error, during a 10 clock cycles a signal is sent to the coupled device, it means the circuit voltage is stabilized and ready to be used.

The necessary time to send the trigger signal to the coupled device is 10 continuous ADC cycles.

The voltage margin allowed is 0.1 V, to more or to less.

The signal sent can be seen as a flag warning about the circuit stabilization. This flag must be active high when signaling stabilization, and hold on that until a destabilization or a system shut down occurs. When that happens, the flag must be active low, indicating a non-operational system.

2.6. Adder

Adder sub block has as function the error calculation between the data received from ADC and Serial (readjusted by Soft Process block).

This error is calculated for each ADC conversion, that means, each value generated by ADC is compared with a previous value saved on Soft Process block.

This difference is only used to feed PID block if a trigger signal from Edge Detect is able. This signal is always low and just is high during one clock cycle, when necessary.

This block results is used as feed-back on Soft Process block in order to adequate the voltage step on its output. As there is a loop between Adder and Soft Process, the best approach to implement it is a sequential circuit solution.

2.7. Edge Detector

This sub block is responsible to analyze the trigger signal from ADC and convert it in a unique digital pulse clock. It is also responsible to synchronize the digital data from ADC.

The Edge Detector converts the ADC trigger that operates on 1 MHz frequency to a 64 MHz signal. It is a required conversion, in order to allow digital blocks more time to process and execute their functions.

The data from ADC is saved on a 8 bits register, and can be accessed by other digital blocks, together with the synchronized trigger.

2.8. PID

PID sub block [4], has as function the calculation of the duty cycle adjust required to keep the output voltage at the desired range. The PID coefficients, as shown previously, may be programmed by user. PID has 3 (three) 9 bits registers, 3 (three) 16 bits registers and 2 (two) 24 bits registers.

The 24 bits registers are used to save the previously control signal and the actual control signal. The higher 14 bits are used to represent

the integer part, and the lower 10 bits are used to represent the fractional part.

The 16 bits registers are used to save the Kp, Ki and Kd constants, and, by default, they have the following values, shown on Figure 3.

Kp = 0; Ki = 0,0078125; Kd = 0;

Figure 3. Constants.

The 9 bits registers are used to save the actual error and two previously errors.

The Ki, Kd and Kp registers work with fixed point, using 6 bits to represent the integer part and 10 bits to represent the fractional part.

Every time the adder sends a trigger signal to the error register, it is updated.

The PID output is calculated following the equation below:

$$O(n) = O(n-1) + k_0 * E(n) + K_1 * E(n-1) + K_2 * E(n-2), \quad (1)$$

where:

O(n) = New control signal.

O(n-1) = Previously control signal,

E = error,

N = actual sample,

K = constant

PID block sends 6 bits to PWM. This data is used to generate the PWM frequency.

2.9. PWM

The PWM - Pulse Width Modulation - decrease the difficulty to implement physically the circuit. It is a fair implementation, because it has a great control over voltage and current level, keeping the same power level.

The PWM is parameterizable, generating a digital output observing the desirable work frequency.

This feature allows a large number of different duty cycles, as much as 64 variations. The voltage scale varies from 0V, duty cycle equal 0, to 3.3V, duty cycle equal 63. The other voltage values are obtained in this range.

The PWM is configured by a specific register that only can be written when the actual duty cycle is finished. After the work cycle, the register value can be changed, starting a new work cycle. The PWM frequency is 1 MHz.

The figure below was extracted from a PWM schematic drawing illustrating the internal system, shown on Figure 4.

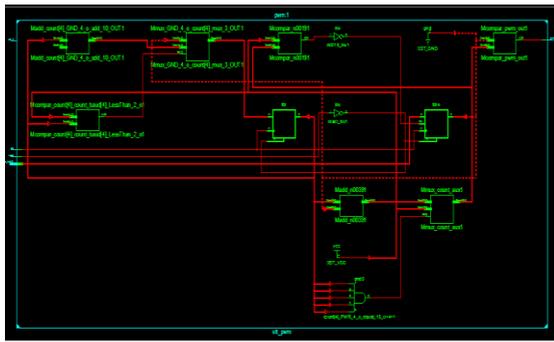


Figure 4. Schematic PWM.

3. Results

The SPC project has been a combination of discipline and hard work to achieve all their goals and meet the required deadlines. The digital control of the static power convert "SPC" has 54576 registers and 1195 Flip-Flops. The main concern on the development was the resources optimization, always performing a trade-off between power consumption and circuit area, making it a considerable alternative in a future use for an ASIC.

The figure bellow shows the top diagram of the SPC digital controller, shown on Figure 5.

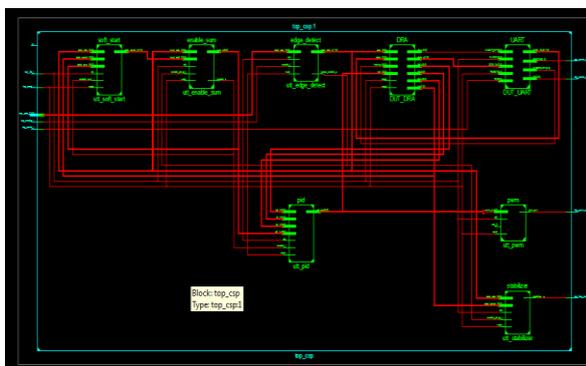


Figure 5. Top Block and IO's.

The functional verification process of the digital controller of the static power converter used ERM methodology, validating the logic functionality of the digital circuit described on VHDL. The environment's components are done using e-language – a hardware verification language. The environment has a stimulus generator, called sequence generator that send signals to the Bus Functional Model, and it converts the inputs generators from a high level approach to a stimulus format understandable by the SPC digital controller.

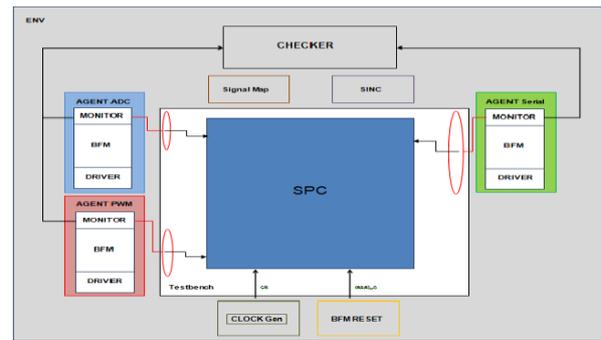


Figure 6. Verification Environment of SPC Digital Controller

4. Conclusion

The design of the digital controller of the Static Power Converter, based on FPGA may be compared with a puzzle, starting with the architecture development, then the implementation with VHDL language, and finally passing by the functional and logical verification through simulation using Nexys 2 kit tools.

This project has been supported by the Communication Department of the University of Campinas, Faculty of Electrical and Computer Engineering, and also by the Visual Communication Department, that provided tools and software used to develop this project.

It's also very important to thanks Dr. Yuzo Iano, responsible to advising the author of this paper.

References

- [1] B J Patella, Prodic, D. Makisimovic, "High-Frequency Digital PWM Controller IC for DC-DC Converter ", IEE TRANSACTIONS ON POWER ELETRONICS, VOL.18, NO.1, JANUARY, 2003.
- [2] Volnei A. Pedroni., Circuit Design With VHDL, MIT Press, Cambridge, Massachusetts, 2004, 369-376.
- [3] Chu Pong P., FPGA Prototyping by VHDL Examples, JOHN WILEY & SONS, INC., PUBLICATION, 2008, 461-471.
- [4] O'Dwyer Aidan, HAND BOOK OF PIAND PID CONTROLLER TUNING RULES, Imperial College Press, Dublin, 2006, 560-564.
- [5] Interfacebus, EIA-232 Bus. http://www.interfacebus.com/Design_Connector_RS232.html. (Acessed March. 19, 2014).