IOSR Journal of Engineering (IOSRJEN)
e-ISSN: 2250-3021, p-ISSN: 2278-8719, www.iosrjen.org
Volume 2, Issue 10 (October 2012), PP 49-53

FPGA Design Methodology for Time Domain Dead Beat Algorithm

S. Saleem Malik[1], Syed Jahangir[2]
[1]*PG Scholar, Dept. Of ECE, Madina Engg College, Kadapa-516003, A.P., INDIA*
[2]*Associate Professor,Dept. Of ECE.Madina Engineering College Kadapa-516003, A.P., INDIA*

Abstract–*This paper proposes a way of implementing a deadbeat controller in FPGA. The focus is on the FPGA implementation of the digital controller. The emphasis is on the software tools for design and simulation of FPGA based hardware for control applications. The FPGA is interfaced to the controlled process by means of serial analog to digital converter (ADC) and digital to analog converter (DAC). The deadbeat regulator output can be depicted: a strong command at the first sample when the reference signal change is observed followed by smaller changes and reaching stationary value after 3 sample periods. A method to implement a deadbeat controller in an FPGA was presented. The controller is studied as a digital filter and a direct form I implementation is derived. The advantage of the described method is that it allows "bit and cycle accurate" simulation of the implemented controller in the Simulink environment. The hardware interface to the ADC and DAC is also described. The experimental results present the method application to a case study: control of a DC motor. The main contribution is the method for design and simulation of control hardware implemented in FPGA.*

Key words: *FPGA, deadbeat, DC motor, System Generator.*

I. Introduction

The challenge of the verifying a large design is growing exponentially. There is a need to define new methods that makes functional verification easy. Several strategies in the recent years have been proposed to achieve good functional verification with less effort. Recent advancement towards this goal is methodologies. The methodology defines a skeleton over which one can add flesh and skin to their requirements to achieve functional verification. The report is organized as two major portions; first part is brief introduction and history of the functional verification of regular Carry select adder which tells about different advantages Carry select adder and RCA architecture and in this Regular Circuit one drawback is there overcome that complexity problem we go for modified architecture of CSLA .

Carry Select Adder (CSLA) is one of the fastest adders used in many data-processing processors to perform fast arithmetic functions. From the structure of the CSLA, it is clear that there is scope for reducing the area and power consumption in the CSLA. The proposed design has reduced area and power as compared with the regular CSLA with only a slight increase in the delay. This work evaluates the performance of the proposed designs in terms of delay, area, power, and their products by hand with logical effort and through custom design and layout in Xilinx XC series devices CMOS process technology. Architecture of the test bench gives complete description about the components and sub components used to achieve the verification goals and also explain about improvements made in the design of the usb-i2c bridge, test plan identifies all the test case required to meet the goals and finally results of the project..

II. Deadbeat Control Algorithm :

1.1 Theoretical presentation of the algorithm:

It is a time domain design technique based on the imposing of a certain response for the control system, using a discrete model of the plant. The reader should resort to the literature for a complete description of the algorithm, as it is well known. In the next section, the customization of the algorithm for the selected case study is presented.

2.2 Case study: deadbeat motor control:

The case study considered is a DC motor control system. The model of the motor is given by

$$G_p(s) = \frac{K_m}{1 + sT_m} \qquad (1)$$

\

where *K m* is the motor gain and *m T* is the motor time constant.
The desired response of the control system at a step input was considered to be:

$$y[T] = 0; y[2T] = 0.8; y[3T] = 1.05; y[4T] = y[5T] = ... = 1 \qquad (2)$$

In this case, the discrete time transfer function of the closed loop system is:

$$G_0(z) = \frac{0.8z^2 + 0.25z - 0.05}{z^4} \qquad (3)$$

Taking into account the parameter values of the motor used for the experiment, *Km* = 0.66 and *Tm* =1.2s, the discrete transfer function for a sampling period = 0.67108864 *s T* is obtained:

$$G_p(z) = \frac{0.2827}{z - 0.5716} \qquad (4)$$

The sampling period was chosen to be a power of 2 multiple of 40ns, the control system clock period. This allows an easy implementation of the sampling period generation circuitry. Also, the sampling period must respect the restriction mentioned in [4]:

$$\frac{T_s}{T_{\Sigma f}} \geq 0.36 \qquad (5)$$

where *T*Σ *f* is the sum of the time constants of the plant. The sampling period has an influence also on the initial command value. So it was verified that for the proposed design the command value does not exceed the saturation limits of the physical devices involved in the experiment.
The regulator transfer function is obtained from (3) and (4) according to

$$G_R(z^{-1}) = \frac{1}{G_p(z^{-1})} \frac{G_0(z^{-1})}{1 - G_0(z^{-1})} \qquad (6)$$

If u[k] is the regulator output (command signal) and ε [*k*] is the regulator input, we get:

$$\frac{U(z^{-1})}{\mathrm{E}(z^{-1})} = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2} + q_3 z^{-3} + q_4 z^{-4}}{1 - 0.8z^{-2} - 0.25z^{-3} + 0.05z^{-4}} \qquad (7)$$

where
q0=0; *q1* =2.8297; q 2 =-0.7333; *q3*=-0.6823; *q4* =0.1011
A relation to compute u[k] is obtained from (7):

$$u[k] = \mathrm{q}_1\varepsilon[\mathrm{k}-1] + \mathrm{q}_2\varepsilon[\mathrm{k}-2] + \mathrm{q}_3\varepsilon[\mathrm{k}-3] + q_4\varepsilon[\mathrm{k}-4]$$
$$+ 0.8u[\mathrm{k}-2] + 0.25u[\mathrm{k}-3] - 0.05u[\mathrm{k}-4] \qquad (9)$$

III. Algorithm implementation:

It is obvious from (9) that the deadbeat regulator is an infinite \impulse response (IIR) discrete time system [5]. Fig. 1 presents the direct form I of the regulator. For physical realization of the regulator, direct form I, direct form II, cascade, parallel, lattice or lattice-ladder structures can be used [5]. Taking into account that registers (for implementation of digital delays) are not a critical resource in FPGAs, direct form I was chosen for implementation.

The advantage of the direct form implementation is the fact that addition and multiplication operations are independent. This allows the use of fixed precision for multiplication, but growing precision for the additions. The result is then truncated to the desired precision. Because of the classical structure of the algorithm, a data flow graph [6] is not necessary for algorithm study. Potential parallelisms of the algorithm can be depicted from the direct form I representation in Fig. 1. Because of small cost in hardware reported to the available resources, 18 bit precision is used for the multiplications, and full precision is used for the adders. The result is then truncated (converted) back to 18 bits. The conversion is made with hardware saturation. This uses

more hardware resources, but eliminates the possible catastrophic effects of overflow [3]. Fig. 2 presents the implementation of the deadbeat regulator in Simulink® using Xilinx® System Generator for DSP™ blocks.

IV. Simulation results:

The main advantage of the System Generator for DSP™ tool is that it permits simulation of the designed hardware structures in the Simulink® environment. Simulation results are "bit and cycle accurate". This approach offers a rapid prototyping platform for FPGA-based design. Results showed that the error introduced by the fixed point quantization of the regulator parameters and input signal quantization are very small.

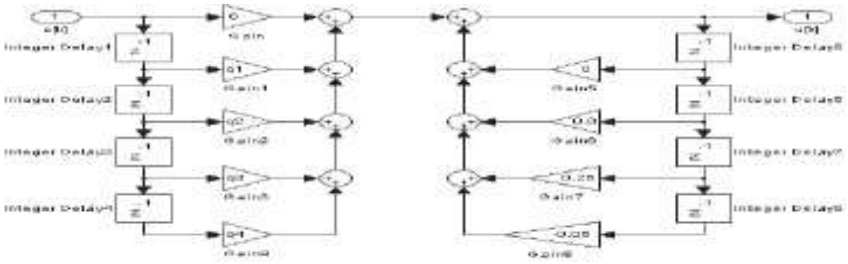

Fig. 1. Direct form I of deadbeat regulator.

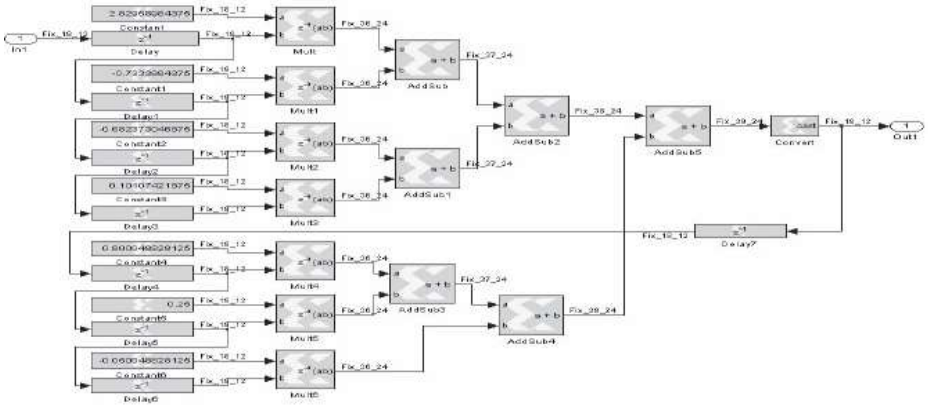

Fig. 2. System Generator implementation of the deadbeat regulator

Fig. 3 shows the simulation environment. The part between "Gateway in" and "Gateway out" blocks is the designed regulator which will be implemented inside the FPGA. The gateway in signal will be read from a 14 bits ADC with signed output. Taking into account that the motor speed is converted to a positive voltage between 0 and 8V, the ADC input will be always positive. Only 12 bits are used from the ADC, and they are interpreted as an unsigned fixed point number, with 3 integer bits and 9 fractional bits.The gateway out signal will be fed to 12 bits DAC. It is also a positive number. It must be converted from the precision of the deadbeat controller calculations (18 bits) to a 12 bit format. Because the command signal saturates at 13V, it will be interpreted as 4 integer bits and 8 fractional bits.

The difference between the double precision floating point Mat lab implementation of the deadbeat regulator and the fixed point FPGA implementation can be observed by means of the Scope1 in the simulation. It has shown not higher than the order of 10−3 for an input step with amplitude of 1. This corresponds, for a simulation length of 30s, to a SNR of around 60dB. Much of the difference comes from the limited resolution of the ADC and DAC, not from the quantization error of the parameters. Running simulations with higher word lengths for the ADC and DAC has shown a decrease of the difference shown by Scope1 by an order of magnitude.

V. Digital Control System:

In this section, some of the implementation details of the control system are outlined. The control system was designed at the schematics level and is composed from three entities: the ADC control circuitry, the signal processing circuitry and the DAC control circuitry. The ADC and DAC used are low cost with serial interface. Because of the plenty resources available on the FPGA chip, the serial interface is implemented by hardware, without using a soft-core microprocessor.

A finite state machine (FSM) is necessary for each part of the control circuitry. The communication between the three entities is done via handshake signals of the type ready/strobe. The circuits are synchronized

by a 25MHz clock signal. Each entity remains in an idle state until data is available from the preceding entity. The ADC control entity is activated by a Ts signal, which is a periodic signal, giving the sampling period.
The sampling period is obtained from the 25MHz clock by using a divisor by 224, implemented by two cascaded counters, one 16 bits wide and the other 8 bits wide. So, the sampling period is around 0.6711s. It was chosen to respect the restriction in (5) and to insure a proper command value at the beginning, avoiding saturation.

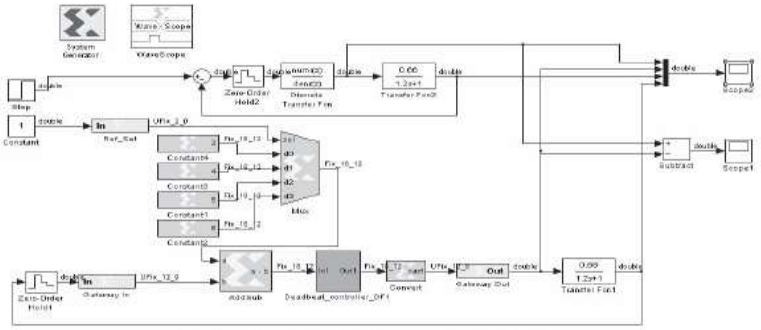

Fig. 3. Simulation of deadbeat controller in Simulink

From the System Generator block schematic a Verilog module is automatically generated. This Verilog module is then instantiated in the schematic, inside the signal processing entity. It is driven by the same 25MHz clock signal, and the FSM generates a clock enable pulse at every sampling period, when the A/D data is available for processing. The computation of the next command value is done in only one clock cycle. Using the FPGA embedded multipliers; the equation (9) is computed in only 40ns.

VI. Experimental Results:

For the experiments, a 24V Leybold Didactic 73411 Motor Generator Set was used. The motor voltage is applied by means of a Leybold Didactic 73413 Power Amplifier Block. The DAC on the FPGA board has an output range of 0 to 3.3V, so a Leybold Didactic 73419 Gain and Offset Adjust Block was used to extend the range to 0 to 12V. A gain value of 3.64 was used.The motor set has an analog output that provides a voltage proportional to the motor speed. The range of this output is 0 to 8V. The ADC on the FPGA board has a range of +/-1.25V around 1.65V and a signed output. Because of the environmental noise, only 12 bits of the ADC 14 bit output resolution will be used. The input is scaled down and offset to the range 1.65V to 2.9V. For this, a gain value of 0.156, followed by an offset of 1.65, was used.

The reference value for the motor speed is generated inside the FPGA in a digital form, and the error signal is computed numerically by means of a subtracter. Fig. 4 shows a comparison between the Simulink® full precision simulation results (upper) and the experimental results obtained by the FPGA implementation (lower). The motor output reaches approximately 80% from the reference value in the first sample period. A small difference can be observed in the response of the real device from the simulated model, caused by nonlinearity of the motor and the amplifiers chain.

VII. CONCLUSION:

A method to implement a deadbeat controller in an FPGA was presented. The controller is studied as a digital filter and a direct form I implementation is derived. The advantage of the described method is that it allows "bit and cycle accurate" simulation of the implemented controller in the Simulink® environment. This way, the designer can proceed to the following steps with high confidence in the "correct first time" operation.

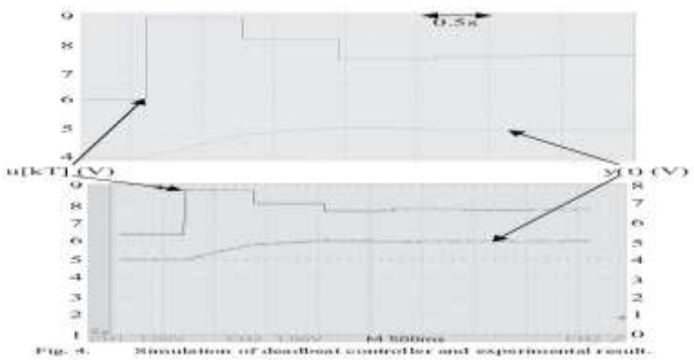

Fig. 4. Simulation of deadbeat controller and experimental result.

Due to the dedicated embedded multipliers available even in the low cost FPGAs, very fast implementations of control algorithms are possible. The presented design uses 7 multipliers in parallel and only 9% of the available logic cells. The massive possibility of parallelism that control algorithms provide is very appropriate for FPGA high speed implementation. The algorithm computation is performed in only one clock cycle, taking less than 40 ns. Future work must be done to investigate the possibility of using the same methodology for other control algorithms. Fast processes with fast sampling rates should be considered, to take advantage of the fast FPGA implementation.

References:

[1] C. Maxfield, FPGAs: world class designs, Newnes Elsevier, 2009.
[2] "Spartan-3E FPGA family: complete data sheet", Xilinx, 2008.
[3] A. Onea, Prelucrarea semnalelor, Politehnium, Iasi, 2006, in romanian.
[4] R. Isermann, Digital control systems, Springer-Verlag, New York, 1981.
[5] J. Proakis, D. Manolakis, Digital Sygnal Processing: Principles,Algorithms ,and Applications, 3rd Edition, Prentice-Hall, New Jersey,1996.
[6] E. Monmasson, M. N. Cirstea, "FPGA Design Methodology for Industrial Control Systems - A Review", IEEE Transactions on Industrial Electronics, vol. 54, no. 4, August 2007.
[7] M.-W. Naouar, E. Monmasson, A.A. Naassani, I. Slama-Belkhodja, N. Patin, "FPGA-Based Current Controllers for AC Machine Drives – AReview", IEEE Transactions on Industrial Electronics, vol. 54, no. 4,August 2007.

AUTHORS

S.SALEEM MALIK received the B.Tech, Electronics and Communication from JNTUA, Anantapur, India in 2010 and pursuing his Master's degree in VLSI design from JNTUA, India. He has active research interests in the areas of vlsi design.

Syed Jahangir Badashah received B.E. degree in Electronics & Communication Engineering from Gulbarga University in 2002,M.E.in Applied Electronics from Sathyabama University in 2005.He is currently doing research in image processing from Sathyabama University. He is having an experience of 10 years, in the field of teaching, presently working as Associate Professor in the department of ECE, Madina Engg College, Kadapa. He is a life time member of IETE & ISTE.