Journal of Analog and Digital Devices

Volume 2 Issue 1

Digital Pulse Width Modulation Controlled DC-DC Buck Converter using VHDL Coding

Joseph Anthony Prathap, T. S. Anandhi
Department of EIE, Annamalai University, Chidambaram, India
E-mail: jap1116@yahoo.co.in, ans.instrus@gmail.com

Abstract

This paper presents the control of the DC-DC Buck converter using Digital Pulse Width Modulation (DPWM) methods in MATLAB-SIMULINK using XILINX TOOL. The conventional PWM is compared with the Digital Pulse Width Modulation. The DPWM generators like Counter based Digital Pulse Width Modulation (CDPWM); Delay-line based Digital Pulse Width Modulation (DDPWM) and Hybrid based Digital Pulse Width Modulation (HDPWM) are coded using VHDL. The DPWM generator is used for the control and analysis of the DC-DC Buck converter. The time transient analysis of the DPWM techniques are evaluated and compared.

Keywords: *DC-DC buck converter, digital pulse width modulation, VHDL coding, system generator*

INTRODUCTION

In power electronics, the control engineering is an integral part of the system. In recent years, the power devices are subjected to digital control than the analog control as it merits with high performance, less area and lower power consumption. The DC-DC buck converter is utilized in many power applications. The DC-DC buck converter is considered as mathematical model equivalence to analysis its behavior.

The mathematical model of the DC-DC buck converter is a tool in analysis of system dynamic behavior, transient and small-signal stability analysis [1]. The transfer function equivalence of the DC-DC buck converter is accurate in MATLAB [2]. The simulation environment MATLAB/SIMULINK is quite suitable to design the modeling circuit, and to learn the dynamic behavior of different converter structures in open loop [3].

The DC-DC buck converter when operated in closed loop exhibits increased stability with high efficiency [4]. The simulation results show that the system performs well under system uncertainties and disturbances and accommodates unmodeled parameters and dynamics. The design method is applicable to the other DC-DC converters under the same control structure [5].

Also, the Digital SMPS using PID controller model removes the disadvantages of analog SMPS, like noise, transient spikes by using PWM techniques. Digital SMPS can provide numerous advantages for low power applications like robustness, flexibility and IP reuse and the very compact and lightweight supplies and reduce the cost [6]. The PID controller design with floating point has capability to do high precision arithmetic [7].

The implementation of PID controller in FPGA has speed, accuracy, power,

compactness, and cost improvement [8, 9]. The control of DC-DC buck converter is enhanced by correcting some errors on the run and by using MODELSIM co-simulation tool which provides faster design process [10].

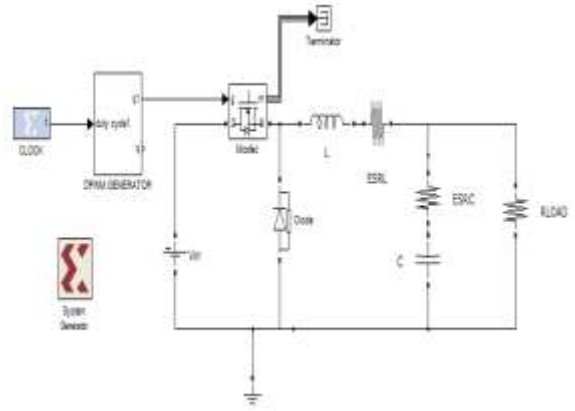

Fig. 1: *Open Loop SIMULINK-XILINX MATLAB Model of DPWM Controlled DC-DC Buck Converter.*

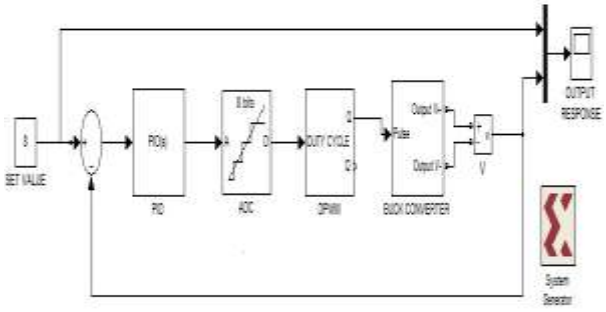

Fig. 2: *Closed Loop SIMULINK-XILINX MATLAB Model of DPWM Controlled DC-DC Buck Converter.*

This paper discusses the model, design and analysis of the DPWM control for the DC-DC buck converter MATLAB-SIMULINK-XILINX TOOL model in open loop and closed loop as depicted in Figures 1 and 2 respectively. The VHDL codes for the DPWM generators are also simulated using MODELSIM software and the same VHDL coding are utilized in the black box blocks available in the SIMULINK XILINX library. Also the concept of black box is utilized for the translation of VHDL code into SIMULINK block.

DC-DC BUCK CONVERTER

The buck converter is a DC-DC converter which accepts a DC voltage signal and produces a reduced regulated voltage signal by a switching control. This switching control is analog in most of the applications. The digital switching control has more advantages than the analog switching control like easy designing, high manipulation power, easy up gradation, immune to environmental changes, easy debugging.

The operation of the buck converter is related to the duty cycle as

$$Vout = d.Vin$$

Where "Vin" is input voltage of buck converter
"Vout" is the load voltage of the buck converter
"d" is the duty cycle.

The duty cycle "d" is the ratio of the ON period and the period of the controlling square pulse.

$$d = \frac{Ton}{T}$$

where "Ton" refers to the ON period
"T" refers to the time period of the cycle

DPWM GENERATORS

The PWM generation involves the modulating sine wave and high frequency carrier triangular wave. The sine wave is overlapped with the triangular wave to produce the PWM. This conventional PWM method is analogous. Recently, the PWM generation is conveniently substituted with the DPWM. The advantages

of the DPWM include high Performance, upgradable, low power consumption, less design complexity, easy design.

Basically, the DPWM generation is classified in three ways. They are Counter based DPWM (CDPWM), Delay line based DPWM (DDPWM) and Hybrid based DPWM (HDPWM).

COUNTER BASED DPWM (CDPWM)

The CDPWM has a counter circuit in its design to generate the carrier is compared with the DC signal. The CDPWM uses a counter, comparator circuit, clocking circuit and a SR-flip flop. The Zero Value Match (ZVM) of the counter circuit is considered as SET and the DC Value Match (DCVM) is considered as the RESET.

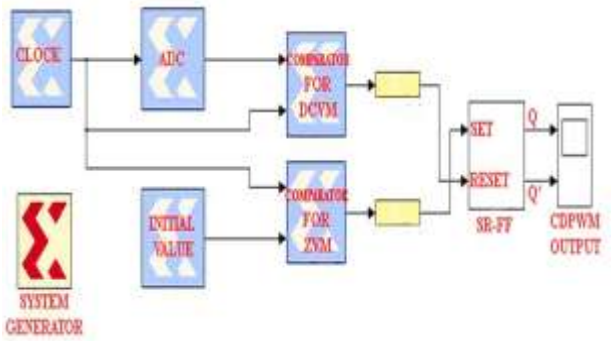

Fig. 3: *SIMULINK-MATLAB Model for the CDPWM using Xilinx Blockset.*

The SET and RESET are fed into the SR-flip flop to generate the DPWM. The clocking circuit provides the clock for the CDPWM.

The block diagram in Figurw 3 depicts the SIMULINK-MATLAB XILINX model. The VHDL codes are translated into black boxes. The resolution is an important aspect of the DPWM techniques. The number of bits used in the design is related with the switching frequency and clocking frequency as given below:

$$F_{clk} = 2^n f_s$$

Where f_s is the switching frequency.

F_{clk} is the clock of the CDPWM circuit.

The disadvantage of the CDPWM is the requirement of high clock frequency. The resolution of the CDPWM design is directly proportional to the clock frequency used. If the resolution of the CDPWM is increased, then the requirement of clock frequency is very high.

DELAY-LINE BASED DPWM (DDPWM)

Figure 4 shows the SIMULINK-Xilinx block set for the N-bit resolution delay line based DPWM method. Delay line based DPWM generation method involves the 2^8 ring counter, 256:1 multiplexer and SR-flip flop.

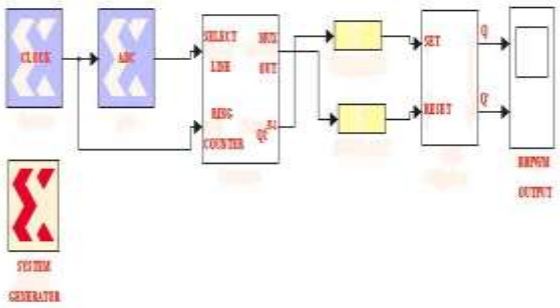

Fig. 4: *SIMULINK MATLAB Model for the DDPWM using Xilinx Blockset Tools.*

In this DDPWM, the resolution is 8 bits. The converted 2^8-bit DC value is fed as select line for the 256:1 multiplexer. 2^8 Ring counter is connected to the input side of the 256:1 multiplexer through 256 D-flip flops as in Figure 5. The SET signal of SR-flip flop is enabled by the 255th pin of the ring counter through its corresponding D-flip flop. The RESET signal of SR-flip flop is enabled by the 256:1 multiplexer output. Thus, the DPWM is generated by the SR-flip flop to operate the DC-DC buck converter.

The disadvantage of the DDPWM is the complexity in the design. The DDPWM has more logic circuit to design, thus leading to the increase in the power consumption and area.

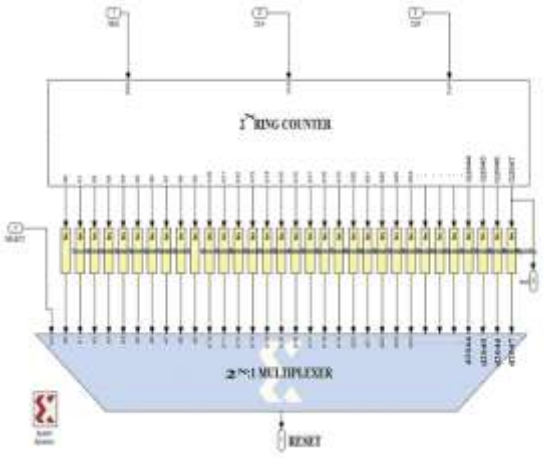

Fig. 5: *SIMULINK MATLAB Model for the Interconnection of Ring Counter with the Multiplexer.*

HYBRID BASED DPWM (HDPWM)

Hybrid based DPWM generation method is the combination of both counter based DPWM and delay based DPWM methods. The DC input signal of 2^8-bit resolution is bit split into 2^5-bit and 2^3-bit resolutions. In which 2^5-bit resolution is used for the generation of the DDPWM and 2^3-bit resolution for the generation of the CDPWM generation. The HDPWM has two SETs and RESETs signals (each from CDPWM and DDPWM generations).These two SET signals are logically AND to give SET signal and two RESET signals are logically AND to give RESET signal. Now these SET and RESET are input for the SR-flip flop as shown in Figure 6.

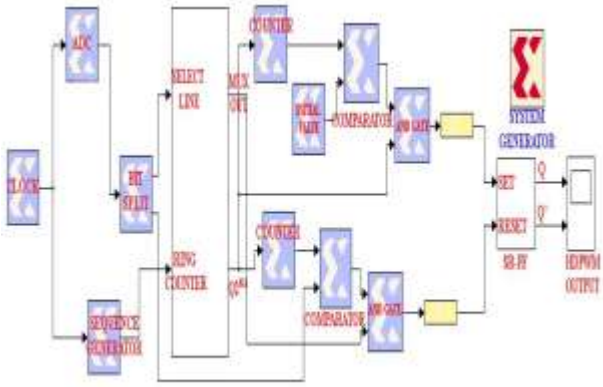

Fig. 6: *SIMULINK MATLAB Model for the HDPWM using Xilinx Block-Set Tool.*

The drawbacks of the CDPWM and DDPWM are overcome by using the HDPWM. This generator requires less clock frequency and design complexity is low.

VHDL CODE FOR DPWM METHODS

This paper concentrates on the VHDL code of the DPWM generator used. The Very High Speed Integrated Circuit Hardware Description Language (VHSIC HDL) codes the digital logic circuits like combinational logic circuits and sequential logic circuits. The VHDL has four styles of modeling. They are

- Dataflow modeling: Deals with the Boolean expression in the design.
- Behavioural modeling: involves the Truth table character of the design.
- Structural modeling: interlinks the Structure of the design.
- Mixed modeling: combinational of the above three modelling.

The DPWM generators are coded in all these styles within the VHDL. The VHDL code for the CDPWM was challenging in containing its high clock frequency requirement. To overcome CDPWM, the resolution of the design is limited. The design of the DDPWM

generator uses more logic circuits of the three DPWM methods. Thus the number of lines in the VHDL code of the DDPWM is high compared to the other codes. The HDPWM coded in VHDL was relatively easy to combine the CDPWM and DDPWM with low design resolution.

VHDL CODE AS SIMULINK MODEL

The VHDL codes of the DPWM generators are to be translated to the SIMULINK MATLAB to interface with the DC-DC buck converter in open and closed loop. The procedure for converting a VHDL code into a SIMULINK block is as follows:

- The VHDL code should be simulated and complied using any simulator (ModelSim/Altera/Xilinx).
- The compiled VHDL code is stored with the extension of *.vhdl in the work library of the default drive.
- This extension of *.vhdl is changed to *.vhd.
- The *.vhd file is moved to the folder where the Matlab Simulink design is saved with extension *.mdl.
- The *.mdl file is opened along with tools box. In the Tool box menu, the Xilinx block set tool is opened.
- The Xilinx block set tool has a block called “Black Box” in the basic elements tools box.
- The black box tool is dragged to the saved *.mdl file.
- A window pops-up asking for the link of *.vhd file to import in the black box.
- The *.vhd file which already saved in the folder of the *.mdl file is selected for importing. Note that if the *.vhd file is not within the same folder of the *.mdl file. Then it would not import the *.vhd file in the black box.
- When imported, a configuration file is generated for the imported black box.
- The black box would have the ports (input and output) of the vhdl code as block ports.
- This black box generated could be easily connected with any other block of Xilinx blockset tool.
- If the black box is to be linked with basic matlab block, the Gateway In block is used.
- Now the black box of VHDL code is simple block which is connected with any blockset of the Simulink Matlab.
- This importing of the VHDL code in matlab uses only fixed values.
- The System Generator block should be used whenever the *.mdl file is using the Xilinx block set tools for error free simulation.

RESULTS AND DISCUSSIONS

These simulated output are obtained from the VHDL simulation tool namely MODELSIM. The VHDL codes for the above were designed using the 2^{10} bit resolution. The responses of the DC-DC buck converter with the counter based DPWM technique is shown in Figures 7 –9. The input voltage V_{in} = 20V and the corresponding output voltage V_{out} is measured in the response graph. Figures 10–12 show the response of the DC-DC buck converter using the delay-line DPWM techniques. The loop response of the DC-DC buck converter using the hybrid based DPWM technique are given in Figures 13–15. Table 1 shows the specification of the buck converter used in this work. The DPWM uses the switching frequency of 16 MHz. The Table 2 & 3 are tabulated by analyzing the time transient parameters for both transfer function equivalence and DC-DC buck converter. The DPWM generation for the counter based DPWM, delay line based

DPWM and hybrid based DPWM are shown in Figures 16–18 respectively.

Table 1: *Specification of Buck Converter.*

V_{in}	20V
L	16µH
C	15nF
Load	10 Ω
$ESR_{(L)}$	16µΩ
$ESR_{(C)}$	0.6mΩ
DPWM Frequency	16 MHz

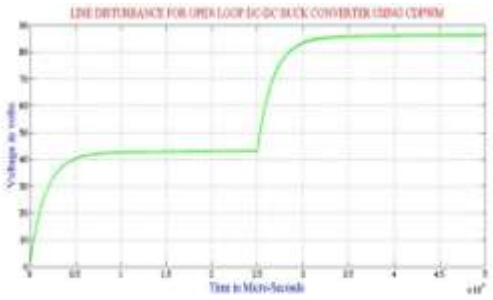

Fig. 7: *Output Response of DC-DC Buck Converter with Line Disturbance using CDPWM.*

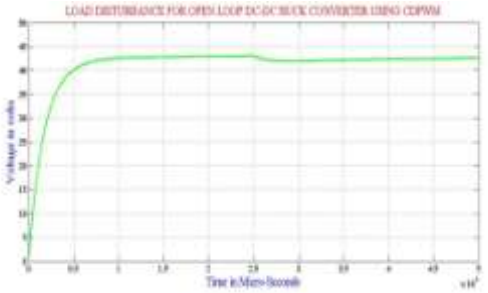

Fig. 8: *Output Response of DC-DC Buck Converter with Load Disturbance using CDPWM.*

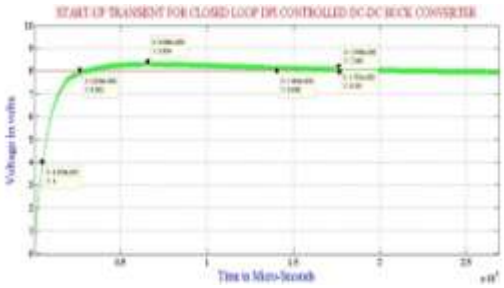

Fig. 9: *Start-Up Transient Closed Loop Response of DC-DC Buck Converter using Counter based DPWM.*

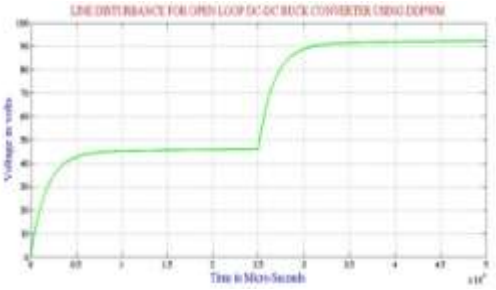

Fig. 10: *Open Loop Response of DC-DC Buck Converter with Line Disturbance using DDPWM.*

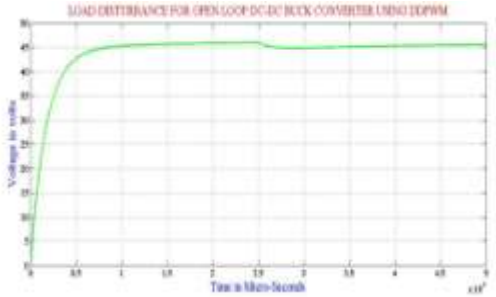

Fig. 11: *Open Loop Response of DC-DC Buck Converter with Load Disturbance using Delay Line based DPWM.*

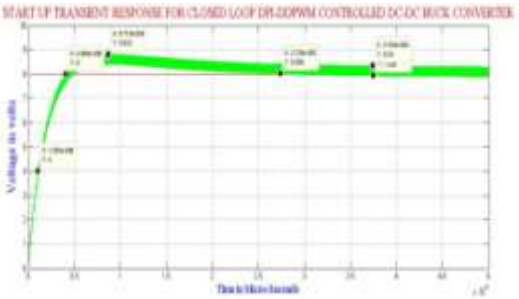

Fig. 12: *Start-Up Transient Closed Loop Response of DC-DC Buck Converter using Delay-Line based DPWM.*

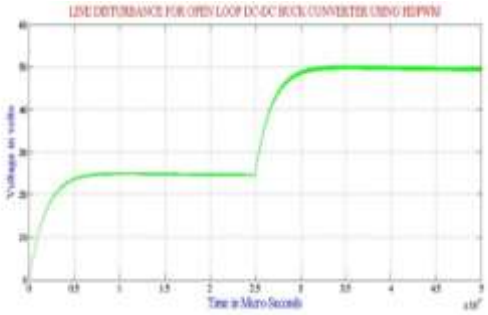

Fig. 13: *Open Loop Response of DC-DC Buck Converter with Line Disturbance using Hybrid based DPWM.*

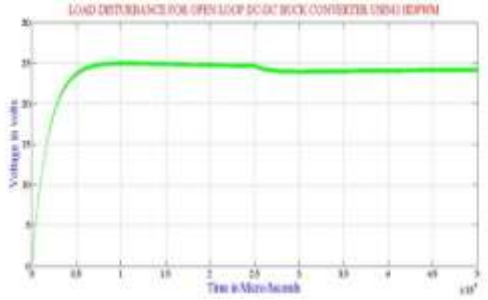

Fig. 14: *Open Loop Response of DC-DC Buck Converter with Load Disturbance using Hybrid based DPWM.*

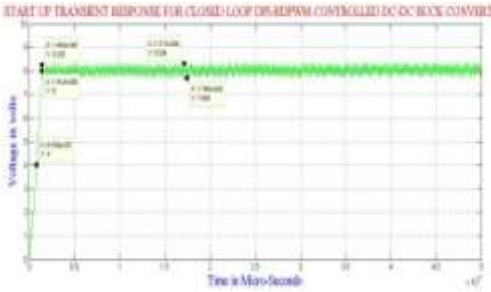

Fig. 15: *Start-Up Transient Closed Loop Response of DC-DC Buck Converter using Hybrid based DPWM.*

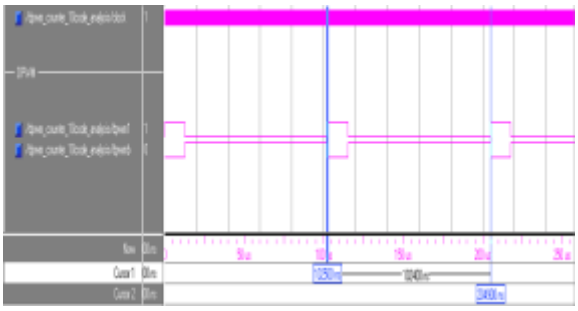

Fig. 16: *Counter based DPWM Generation with 2^8 Bit Resolution.*

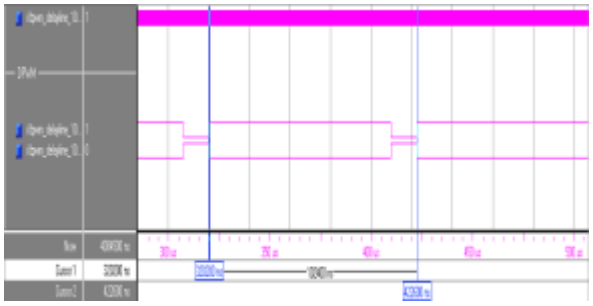

Fig. 17: *Delay-Line based DPWM Generation with 2^8 Bit Resolution.*

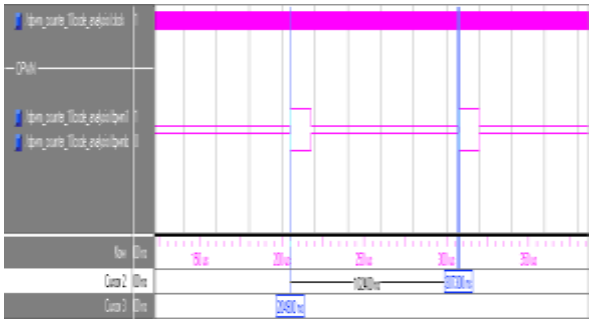

Fig. 18: *Hybrid based DPWM Generation with 2^8 Bit Resolution.*

Table 2: *Comparison of the Start-Up Time Transient Parameters for Transfer Function model and Buck converter model using PWM.*

Methods	**Buck Converter Transfer Function Model**	**Buck Converter Model using PWM**
Settling Time (t_s) **µs**	3.701	3.803
Rise Time (t_r) **µs**	0.311	1
Delay Time (t_d) **µs**	0.234	0.485
Peak Time (t_p) **µs**	0.5	1.837
Steady State Error (e_{ss})	0	0.125
Percentage Overshoot (**%MP**)	63%	46.25%

Table 3: *Start-Up Time Transient Parameters for Buck converter model using CDPWM, DDPWM and HDPWM.*

Methods	**Counter DPWM**	**Delay-line DPWM**	**Hybrid DPWM**
Settling Time (t_s) **µs**	14.04	27.38	1.509
Rise Time (t_r) **µs**	2.636	4.058	1.412
Delay Time (t_d) **µs**	0.4579	1.035	0.8168
Peak Time (t_p) **µs**	6.568	8.713	1.443
Steady State Error (e_{ss})	0.0286	0.04964	0.07466
Percent Over-shoot (**%MP**)	5.05%	10.27%	2.85%

CONCLUSION

The simulation results of the DPWM techniques like CDPWM, DDPWM, HDPWM and PI enabled DPWM techniques are developed by in MATLAB/SIMULINK XILINX blockset. The HDPWM closed loop response is satisfactory for the start-up transience. Future work can be directed towards the real-time implementation of DPWM using Field Programmable Gate Array.

ACKNOWLEDGEMENTS

The authors would like to thank the department of Electronics and Instrumentation Engineering, Annamalai University, Chidambaram, India for the facilities and assistance provided in the laboratory.

REFERENCES

1. Kaur, R., Kaur, N. Mathematical modelling of buck converter. *International Journal on Recent and Innovation Trends in Computing and Communication.* 2014; 2(5): 1226-1229p.
2. Gowda, N.M. M, Kiran, Y., Parthasarthy, S.S. Modelling of buck DC-DC converter using simulink. *International Journal of Innovative Research in Science, Engineering and Technology.* 2014; 3(7): 14965-14975p.
3. Nejad, H.K., Fazly, S.J. Nonlinear modeling and analysis of DC-DC buck converter and comparing with other converters. *International Journal of Engineering and Advanced Technology*. 2014; 4(2): 137-143p.
4. Singh, S.P., Singh, D.K., Kumar, H., Dwivedi, R. Designing and parametric variation of pi controller for buck converter for constant voltage applications. *International Journal of Engineering Trends and Technology*. 2014; 13(4): 169-174p.
5. Zengshi, C. Modeling, analysis and simulation of a buck converter under cascade control. 2011: 1-12p.
6. Patil, S.V., Shukla, T.R., Shah, P.J. Modeling of PID controller based SMPS using FPGA. *International Journal of Innovative Research in*

Science, Engineering and Technology. 2012; 1(2): 159-168p.

7. Alinezhad, P., Ahmadi, A. FPGA design and implementation of digital PID controller based on floating point arithmetic. *The 8th Symposium on Advances in Science and Technology (8thSASTech)*, Mashhad, Iran; 2013.
8. Gupta, V., Khare, K., Singh, R. P. Efficient FPGA design and implementation of digital PID controllers in Simulink. *International Journal of Recent Trends in Engineering*. 2009; 2(6): 147-150p.
9. Gupta, V., Khare, K., Singh, R. P. Efficient design and FPGA implementation of digital controller using xilinx SysGen. *International Journal of Electronics Engineering*. 2010; 2(1): 99-102p
10. Bustamante, E.W.Z., Flores, J.L., Ramírez, E.G., Ramírez, H.S. FPGA implementation of PID controller for the stabilization of a DC-DC buck converter. *Frontiers in Advanced Control Systems.* 2012; 1(10): 215-230p.