International Journal of Engineering & Technology, 7 (4) (2018) 4519-4527

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 20073
Research paper

PWM and duty ratio switching of multiple input converters using FPGAs: a digital logic circuit and VHDL hybrid approach

Immanuel N. Jiya [1] *, Wian Snyman [1], Nicoloy Gurusinghe [2], Rupert Gouws [1]

[1] School of Electrical, Electronic and Computer Engineering, North-West University, Potchefstroom 2520, South Africa
[2] Faculty of Engineering and Physical Sciences, Queen's University Belfast, Belfast, Northern Ireland, United Kingdom
**Corresponding author E-mail: immanueljiya@ieee.org*

Abstract

This paper presents work that addresses the need for the simultaneous switching of the gate signals of controllable switches in multiple input DC-DC converters. A hybrid approach of implementing the gate signal switching using a combination of digital logic circuits and the conventional VHDL is used in this research. This approach reduces the complexity of the gate signal switching of multiple input converters when compared to the conventional methods. A new method of designing the digital logic circuits from the steady state waveforms of the multiple input DC-DC converter is also introduced, the logic circuit was verified in simulation and validated experimentally by implementing it on an FPGA development board. From the experimental results presented, the switching of the multiple input converter was achieved with the possibility of using any type of system controller without affecting the operation of the switching signals. A dead time of up to 400 nanoseconds was achieved between the switching signals and ultimately, the new method of designing the digital logic circuit of the converter operation from the steady state waveform was validated.

Keywords: *PWM Switching; Multiple Input DC-DC Converters; FPGA; Digital Logic Circuit.*

1. Introduction

A lot of attention in research is shifting in the direction of designing and implementing multiple input DC-DC converters [1]–[3]. These converters being designed are utilized in applications ranging from hybrid energy storage systems for electric vehicles to hybrid renewable energy sources to mention just a few [4]–[7]. This paradigm shift to multiple input converters is due to the increasing energy demand and the concurrent demerits of the traditional fossil energy sources [8]–[10].

Over the years, quite a number of multiple input converters have been designed and are still being designed in literature [11]–[14]. However, one common challenge across the different multiple input converters proposed in literature is the numerous amount of power electronic switches, these switches in most cases need to be controlled in real time in pairs of two or more for effective operation of the respective DC-DC converters [15]–[17]. Therefore bringing about the need for high performance digital signal processing (DSP) methods and devices to facilitate the delivery of the switching signal to the respective switches.

There are different methods of implementing the DSP for switching the gate signals of multiple input converters, some of which includes as application-specific integrated circuits (ASICs), complex programmable logic devices (CPLDs) and field-programmable gate arrays (FPGAs). These DSP technologies provide room for parallel switching of the signals, however, the FPGA technology has been more popularly used in literature [18]–[22]. Although the FPGA technology is very robust but many power electronic engineers find the implementation very complex since the programming language, very high speed integrated circuits hardware description language (VHDL), which is required to implement the designs, and the language is thought by many to be counter-intuitive [23]–[25]. Although many researchers have reported to have implemented the FPGA technology in their design, it has not been reported in literature the actual design steps for the purpose of verification [26]–[31].

This paper presents a detailed method of designing and implementing the digital logic circuits for multiple input converters on an FPGA. A hybrid approach of implementing the gate signal switching using a combination of digital logic circuits and the conventional VHDL is used in this research. This approach reduces the complexity of the gate signal switching of multiple input converters when compared to the conventional methods. It also introduces a less complex method of obtaining the digital logic circuits from the steady state waveforms of the converter. In the following section the multiple input converter is discussed and its control is presented in section 3, the experimental results are discussed in section 4 and a conclusion of the research is presented in section 5.

2. The multiple input DC-DC converter topology

The multiple input converter topology considered in this research was first proposed by [32], the schematic of the converter is presented in figure 1. The converter is a non-isolated bidirectional three input converter. It is a buck-boost converter, therefore the relationship between the output voltage and the input voltage is governed by the general equation for buck-boost converters.

For energy transfer from the energy storages V_1, V_2 and V_3 to the DC link and vice versa, the inductors L_1, L_2 and L_3 are charged and discharged in two time constants as presented in figure 2, following the switching patterns presented in table 1. In both table 1 and figure 2, modes A-C presents the flow of energy from the Energy storages V_1-V_3 respectively while modes D-F represents an instance where there is flow of energy from the DC link to the energy storages V_1-V_3 respectively. Since this research focusses on the PWM switching for the control of the multiple input converter, the DC-DC converter is not discussed in too much detail, however, more about the multiple input converter and its applications can be found in [32], [33].

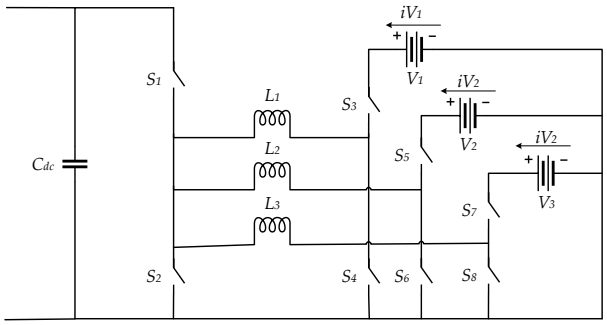


Fig. 1: Schematic of the Multiple Input DC-DC Converter.

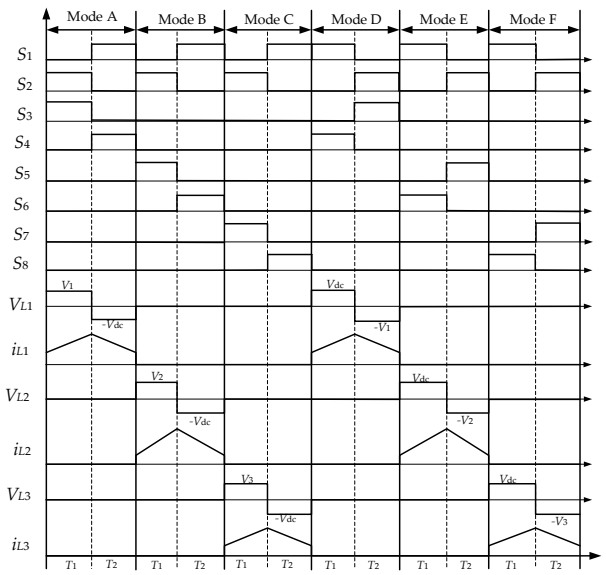


Fig. 2: The Steady State Waveform of the Converter Operation.

Table 1: Conduction of Devices for the Operation Modes of the Converter with Individual Inputs

Modes	T_1	T_2
A	S_3 S_2	S_1 S_4
B	S_5 S_2	S_1 S_6
C	S_2 S_7	S_1 S_8
D	S_1 S_4	S_2 S_3
E	S_1 S_6	S_2 S_5
F	S_1 S_8	S_2 S_7

3. Digital logic circuit design

To control the DC-DC converter under each mode so that the converter gives the right output voltage, the control signal has to be delivered at the gate of each of the switches. In figure 1, it can be seen that there are eight power electronic switches. That is eight different PWM signals that must be delivered simultaneously to the gates of the switches. Figure 3 is a block diagram of the control of the DC-DC converter gate signals, from the block diagram it can be seen that the control topology is basically the voltage mode control (VMC) in which the output voltage is measured and compared to the target output voltage and then the duty ratio is set until the output voltage is equal or satisfactorily similar to the target output voltage. The controller produces one PWM signal which determines the duty ratio D, depending on the input voltage and target output voltage (control objective). This single PWM is then sent to the digital signal processing (DSP) device which splits the signal to each of the gates of the switches. This DSP device was achieved using logic gates. To generate the logic gates used in implementing the individual PWM signals for each switch, the truth table of the converter operation was first generated. The PWM switching was done through logic gates, this was implemented on an Altera 2 DE1 FPGA development board, the switching frequency of 31 kHz was used and the main controller was an ATmega2560 device.

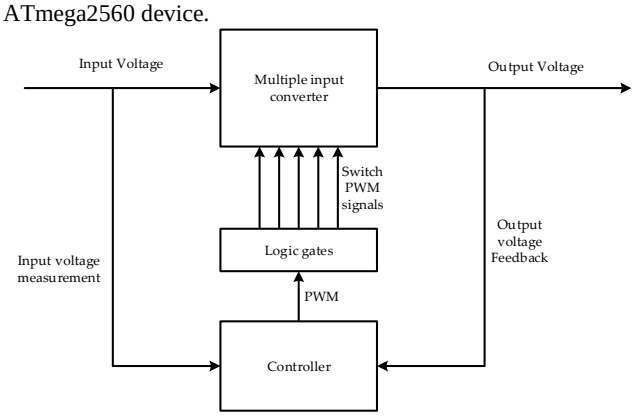


Fig. 3: Block Diagram of the Control Schematic of the Multiple Input Converter.

Table 2 and 3 shows the truth table of the switching pattern for the three input DC-DC converter in figure 1, which was generated from the steady state operation waveform and switching patterns presented in figures 2 and table 1 respectively. The columns I_0, I_1 and I_2 in the tables 2 and 3 are binary sequences used by the controller to determine the mode of operation of the converter starting from 000 to 101 for modes A through to F respectively. Taking table 2 for example, for the converter operation in mode A, it can be seen in figure 2 that during the positive cycle (the ON-state of the PWM signal), that is when the inductor is charging, only switches S_2 and S_3 are ON thereby setting them high (1), the rest are OFF thereby setting them low (0). Furthermore, on table 3 during the negative cycle (the OFF-state of the PWM signal) for mode A, only the switches S_1 and S_4 are ON which indicates the period for inductor discharging. This goes on for the rest of the modes (B-F) in tables 2 and 3.

The next step was to generate the logic gates obtained from the truth table in tables 2 and 3. The truth tables presented in table 2 and 3 was used to design and build the control logic circuits using Logisim 2.7.0, an open source educational development software for designing and simulating digital logic circuits. The simulation results of both positive and negative cycles of the PWM signal for mode A are presented in figures 4 and 5. The signal paths highlighted in light green carries an ON state signal while the line paths with dark green carries the OFF state signals. Flip flops were used to ensure there is smooth operation of the signals and there is no cross operation of the switches that may result in damaging the power electronic switches. It can be observed that the outputs obtained from the logic circuits accurately corresponds to the outputs expected from the truth table in tables 2 and 3 and these in turn correspond to the expected results described in the steady state waveforms presented in figure 2.

The flip-flops observed at the mode selection inputs (I_0, I_1 and I_2) of the digital logic circuits presented in figures 4 and 5, acts as memory to remember the current mode of operation in the occurrence of a mode change. For example, switching from mode A (000) to mode D (011) requires a sequential change from a microcontroller, potentially in the order 000 (mode A) -> 010 (mode C) -> 011 (mode D). The effect of rapidly changing to mode C before mode D will cause some undesirable results. When a mode needs

to be changed, the flip-flop observed at the bottom left corner next to the enable pin disables the input from the controller to prevent sudden switching between undesired modes. Once the correct mode has been selected by the controller, the Enable flip-flop re-enables input from the microcontroller to change to the desired mode. Another function of the Enable flip-flop is to ensure that a full cycle of any mode completes before switching to a new mode, preventing early switching of modes. Lastly, the Fail-Safe flip-flop enables the user to disable all outputs to the MOSFET drivers in case of a critical failure. The Fail-Safe can be re-enabled at any time once the failure has been resolved, whereupon the other flip-flops ensures that the system continues off from the correct mode starting at the beginning of its cycle.

Table 2: Truth Table of ON-State of the PWM Switching Signals Derived From the Steady State Waveform

	Mode selection			Switch selection							
Modes	I_0	I_1	I_2	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
A	0	0	0	0	1	1	0	0	0	0	0
B	0	0	1	0	1	0	0	1	0	0	0
C	0	1	0	0	1	0	0	0	0	1	0
D	0	1	1	1	0	0	1	0	0	0	0
E	1	0	0	1	0	0	0	0	1	0	0
F	1	0	1	1	0	0	0	0	0	0	1

Table 3: Truth Table of OFF-State of the PWM Switching Signals Derived From the Steady State Waveform

	Mode selection			Switch selection							
Modes	I_0	I_1	I_2	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
A	0	0	0	1	0	0	1	0	0	0	0
B	0	0	1	1	0	0	0	0	1	0	0
C	0	1	0	1	0	0	0	0	0	0	1
D	0	1	1	0	1	1	0	0	0	0	0
E	1	0	0	0	1	0	0	1	0	0	0
F	1	0	1	0	1	0	0	0	0	1	1

4. Experimental results

Figure 6 presents the implementation of the logic gates presented in figures 4 and 5 on the Quartus II version 13, the IDE for the FPGA development board. It is seen to be very similar to the figures 4 and 5. The logic gates implemented for the converter has six inputs all coming from the Arduino board, I0, I1 and I2 representing the mode selections, an enable pin which triggers the gates when a mode has been changed, the PWM signal which determines the duty ratio of the switching signal and switching frequency as well the failsafe which shuts down operation when turned OFF by the converter. The only difference between the digital logic circuits presented in figures 2 and 3 obtained for the simulation in Logisim and the experimental implementation of the Altera II DE1 FPGA board presented in figure 6, is that a delay block was added to create a dead-time in the switching signals to prevent cross conduction across adjacent switches (that is the corresponding high side and low side switches) of the DC-DC converter. This delay block is highlighted by the red rectangle in figure 6.

To create an accurate delay between PWM pulses, the 24 MHz internal clock of the FPGA board is used as a timer and has a period of 41.67 ns. If the rising edge of the 24 MHz clock signal is used, the amount of rising edges can be counted and therefore a delay can be created. The delay of a signal can be implemented as follows: Suppress a signal until a predefined integer reaches 0 after it is decremented on each rising edge of the internal clock. The delay can also be adapted for the use of PWM signals, but it is crucial that the delay is always less than the period of the PWM signal. Each time the value HIGH of the PWM signal is detected, the FPGA decrements a user defined integer 'x' using the internal clock of the FPGA. When 'x' reaches 0, the HIGH signal of the PWM pulse is allowed. By inverting the PWM signal, the same delay can be achieved for LOW values of the PWM signal.

The accuracy and resolution of the implemented delay is determined by the period of the internal FPGA clock speed. When using a 24 MHz clock speed, the minimum delay achievable is 41.67 ns. More accurate delays can be achieved by using a faster internal clock speed. The addition of a delay also influences the capabilities of duty cycle. The minimum and maximum duty cycle allowed is equivalent to the percentage of delay time with regards to the clock cycle period, the values of the maximum and minimum duty cycle can be obtained using equations (1) and (2).

$$\min \text{DUTY}\,(\%) = \frac{\text{delay (s)}}{\text{internal clock period (s)}} \times 100 \tag{1}$$

$$\max \text{DUTY}\,(\%) = 1 - \frac{\text{delay (s)}}{\text{internal clock period (s)}} \times 100 \tag{2}$$

Figure 7 (a) illustrates the flow diagram to implement the delay on a FPGA board. The variable 'Count' is defined by the user and determines the length of the delay. For a 24 MHz internal clock speed, the effective delay is given in equation (3). In this case, the value is set to ten which effectively produces a delay of 416.67 ns. To realise the delay, the HIGH value of the PWM signal will only be set after the delay is complete. The delayed PWM HIGH signal is denoted by the variable 'Output' - a LOGIC HIGH value which is enabled after the delay is complete. When a LOW value of the PWM signal is detected, 'Count' is reset to the user defined value.

$$\text{Delay (s)} = \text{Count} \times \frac{1}{24 \times 10^6} \tag{3}$$

The realisation in VHDL code of the delay is shown in figure 7 (b). There are 2 logic inputs, 'clk' and 'start_delay' which represents the internal clock and PWM signal respectively. Only 1 logic output is necessary, namely 'D', which is the delayed 'HIGH' value for the PWM signal. The length of the delay is defined by the constant variable 'counts'. The variable 'countTo' is initially set to the value of 'counts', but is then decremented on each rising edge of 'clk' whenever 'start_delay' is 'HIGH'. When 'countsTo' reaches 0, the delay is complete and 'D' is set to HIGH. Whenever 'start_delay' is 'LOW', the delay is rset and 'countTo' takes on the value of 'counts' and 'D' is set to 'LOW'.

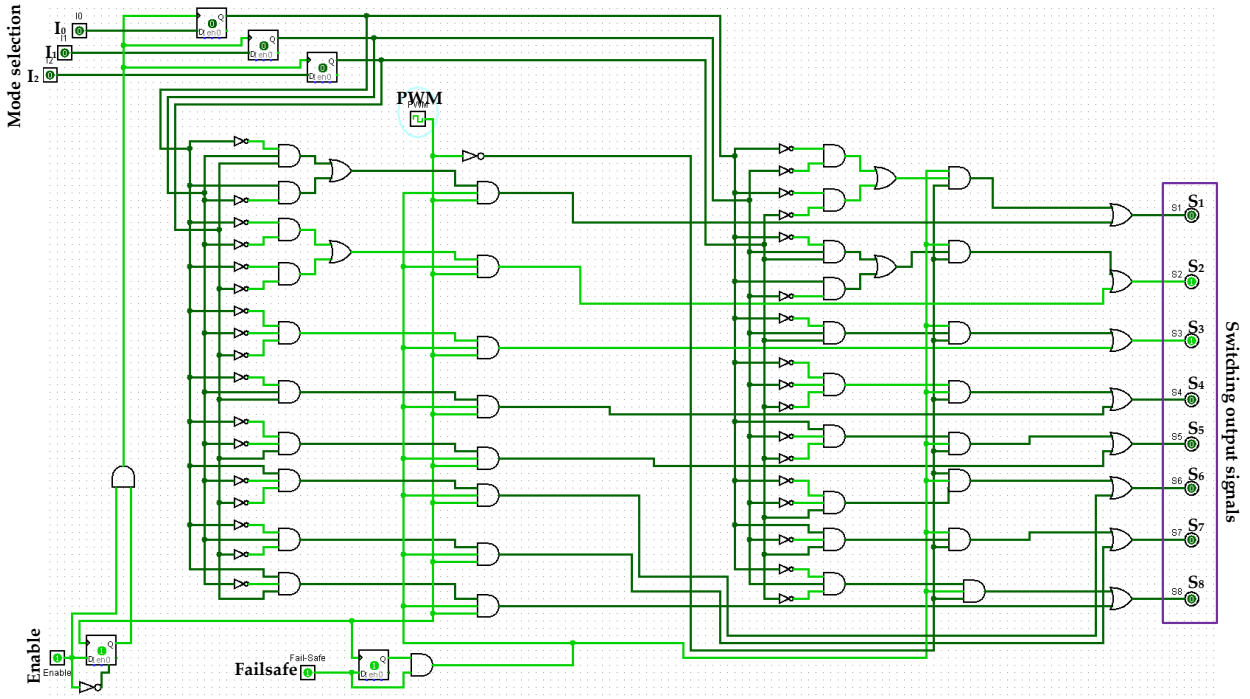


Fig. 4: Simulation Result for the ON-State of the PWM Signal for the Converter Operation in Mode A Having S2 and S3 Turned ON as Presented in Table 2.

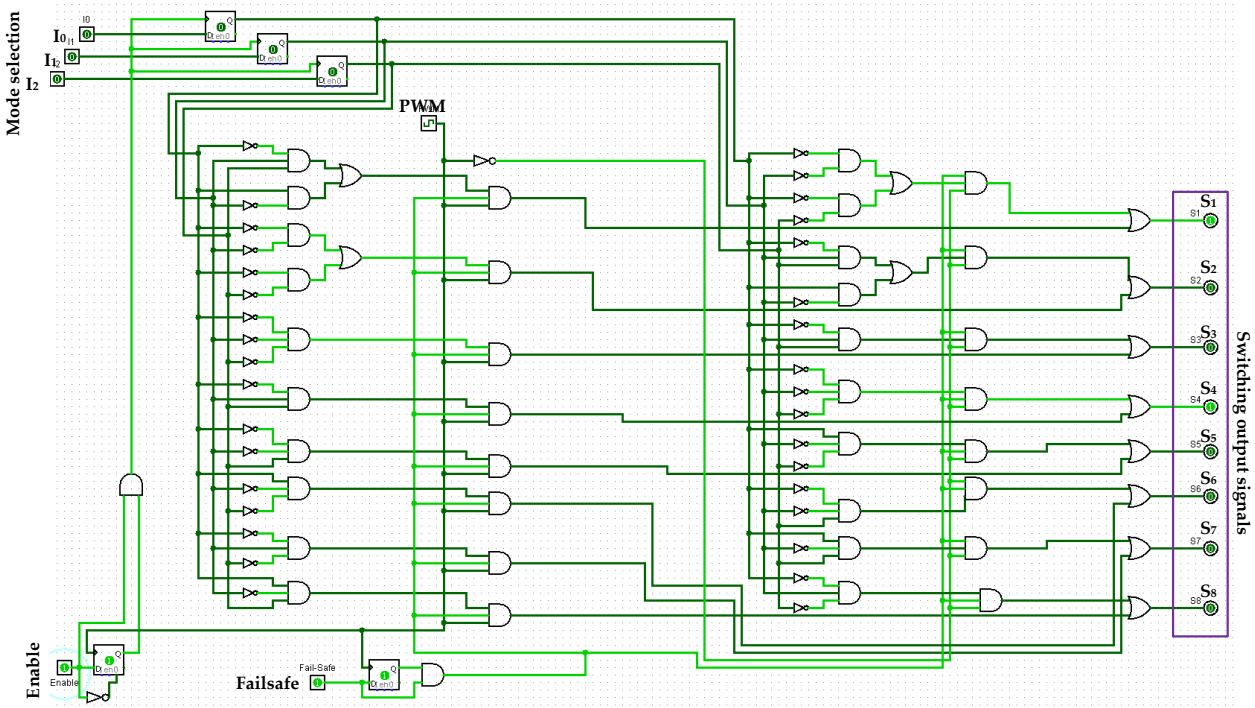


Fig. 5: Simulation Result for the Off-State of the PWM Signal for the Converter Operation in Mode A Having S1 and S4 Turned on As Presented in Table 3.

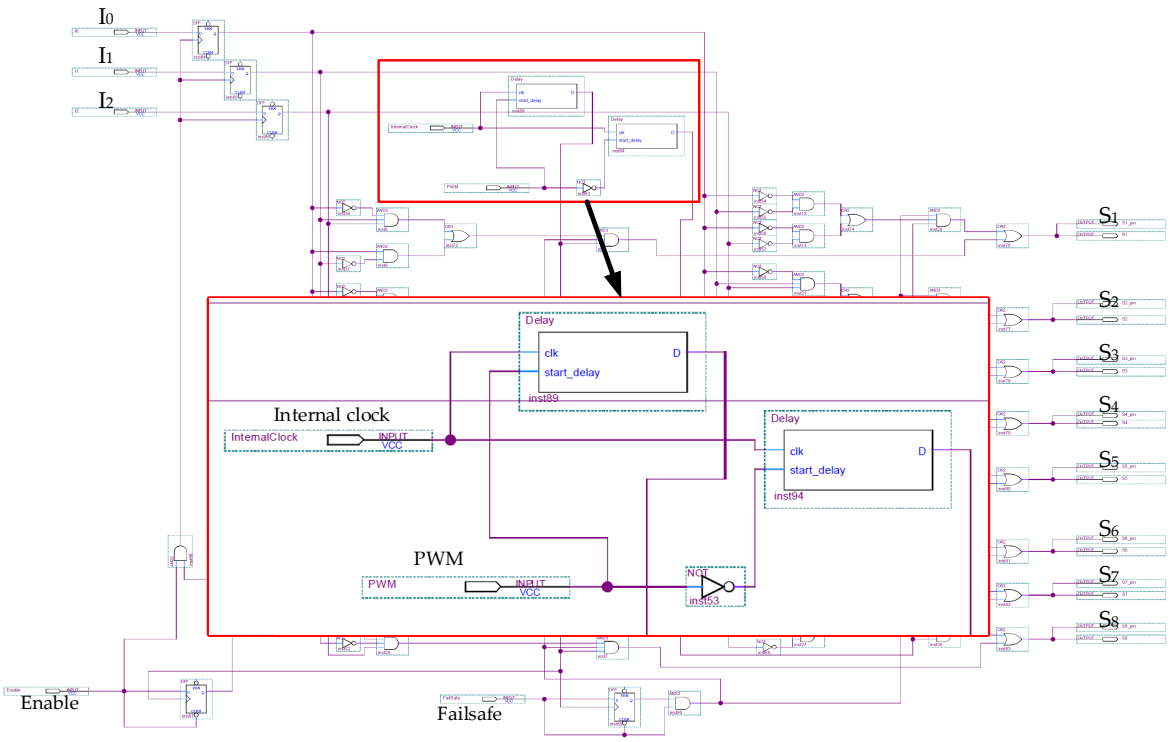


Fig. 6: Logic Circuit of the Gate Switching Signals as Implemented on the Quartus II Version 12 IDE for the Altera DE2 FPGA board.

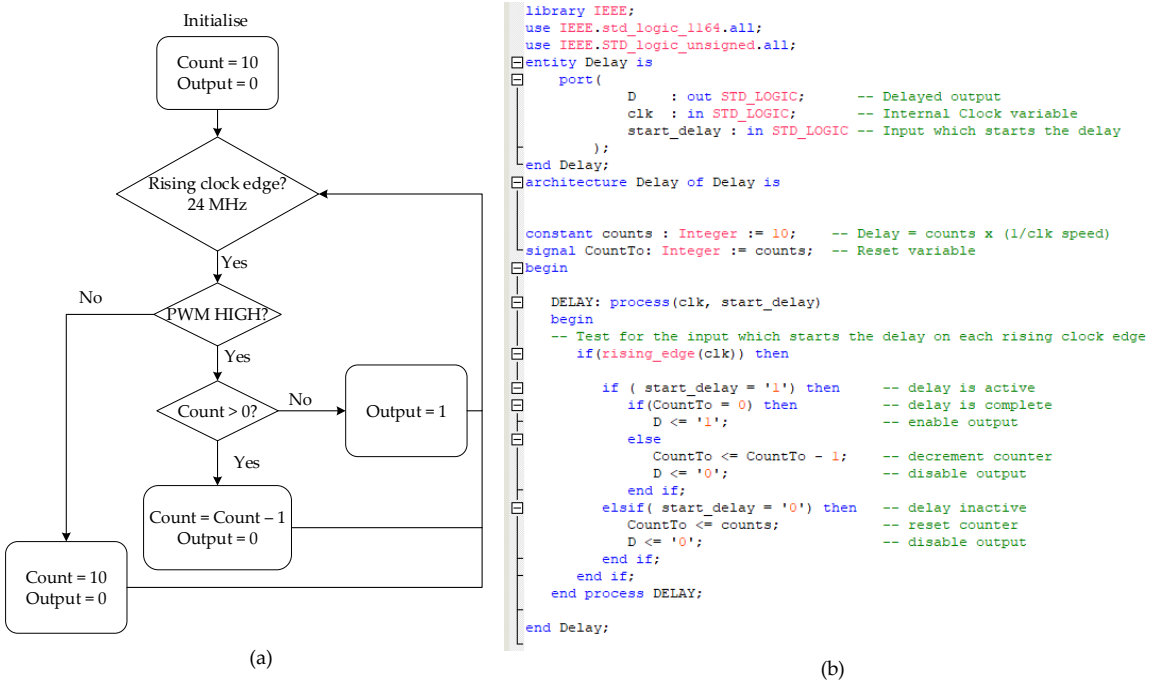


Fig. 7: (A) Flowchart of the Delay Implementation on the FPGA Board and (B) the VHDL Source Code of the Delay Toolbox.

To test the results of the implementation on the FPGA board, the controller was set to operate in open loop by sending a PWM signal of 15% through to 90% duty ratio in increments of 15% respectively for modes A to F. That is, mode A was operated at 15%, mode B 30%, mode C 45%, mode D at 60%, mode E at 75% and mode F at 90%. These modes of operation represent the flow of energy from and to the input ports of the multiple input converter. Modes A to C represents flow of energy from the sources V_1, V_2 and V_3 respectively while modes D to F represents the flow energy from the DC bus to V_1, V_2 and V_3 respectively. The images in figure 8 are the switching signals measured at the output of the FPGA development board before reaching the inputs of the respective MOSFET drivers for the operation in mode A to F respectively for figure 8 (a) to (f).

Comparing these results to the analytical steady state waveforms and the switching truth table developed and presented in figure 2 and tables 2 and 3 respectively, it is seen that the logic circuit implemented on the FPGA board was working perfectly.

Recall that, in figure 6, a switching delay was introduced to the logic circuit of the switching signals, however, before implementing the delay, it was important to see the extent of overlap in the switching time in order to avoid an over compensation in the dead-time. To test for the overlap, the delay circuit was isolated from the logic circuit and the rest of the circuit was allowed to run. Figure 9 (a) is the overview of both the high and low side switch signals at the gate of the MOSFETs before the delay was added, more interestingly the images in figure 10 presents a detailed view of the switching overlap. At the falling edge of the high side switch it is seen that the total switch OFF time is about 80 nanoseconds while in figure 10 (b), it is seen that the low side switch

starts to turn ON 52 nanoseconds before the high side switch is completely OFF thereby resulting in an overlap of 52 nanoseconds. The rising edge of the high side switch is also examined and presented in figure 11. It is observed again, that there is an overlap of about 76 nanoseconds in the switching ON time of the high side and low side.

However, when the delay was added, there was a great improve ment in the dead-time compensation. The image in figure 9 (b) is the scope result of the overview of both high side and the low side switch, comparing this image to the result obtained before the delay was applied, it is easily observed that there is a reduction of about 1.7% duty cycle after the delay was applied, this is due to the dead-time compensation after the delay, a more detailed view of the dead-time compensation is presented in the scope results in figure 12, just like in figure 10 before the delay was added, the falling edge is presented in figure 12 (a) and (b) and it can be observed easily that there is now a dead-time of about 400 nanoseconds between the ON and OFF time of the high side and low side switch, which will go a long way in avoiding a cross conduction of the MOSFET switches. Also, in figure 13, the rising edge of the high side switch is presented and the dead-time of about 400 nanoseconds is also observed.

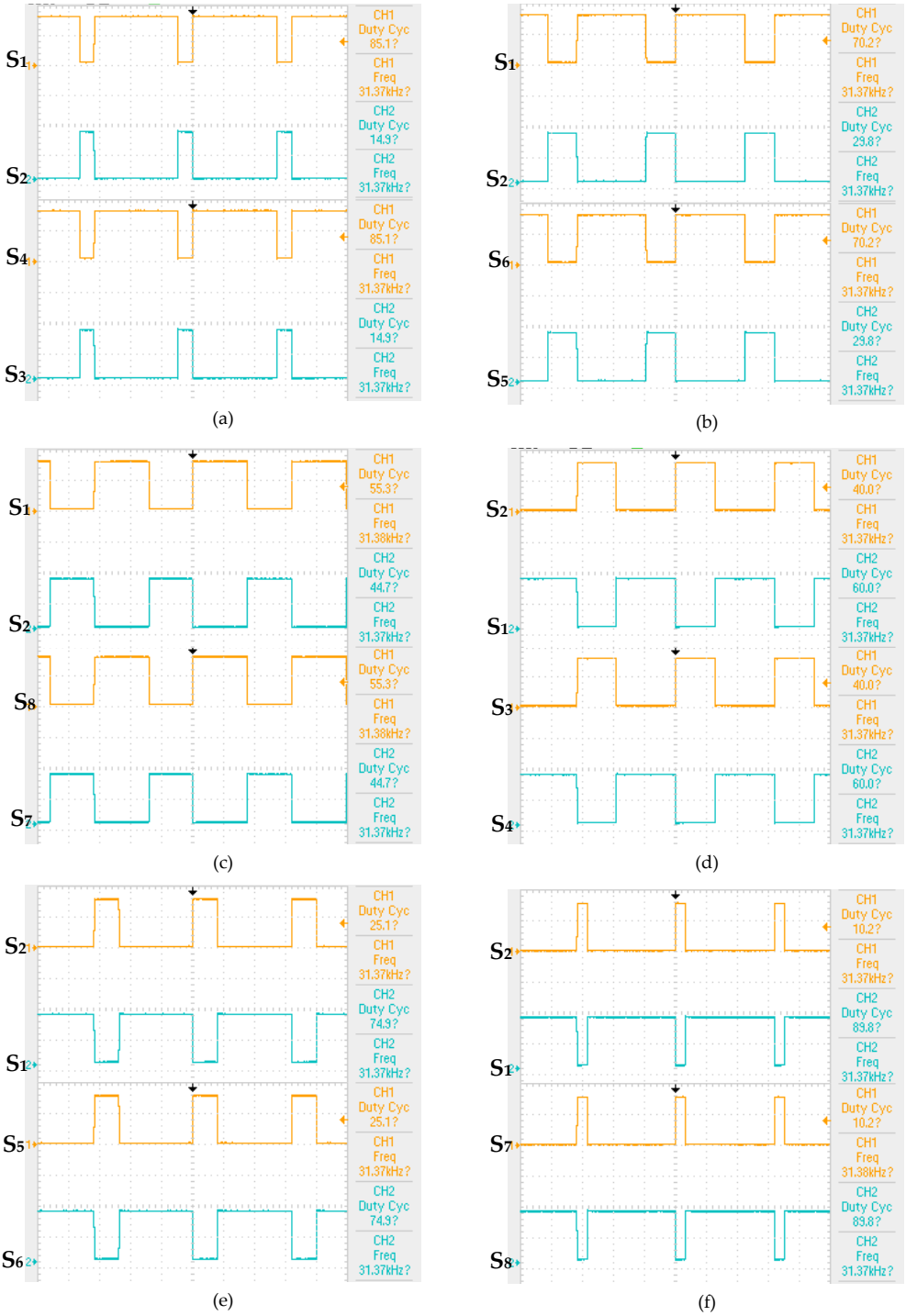


Fig. 8: Output of the FPGA Board While Operating in (A) Mode A at 15% Duty Cycle (B) Mode B at 30% Duty Cycle (C) Mode C at 45% Duty Cycle (D) Mode D at 60% Duty Cycle (E) Mode E at 75% Duty Cycle and (F) Mode F at 90% Duty Cycle.

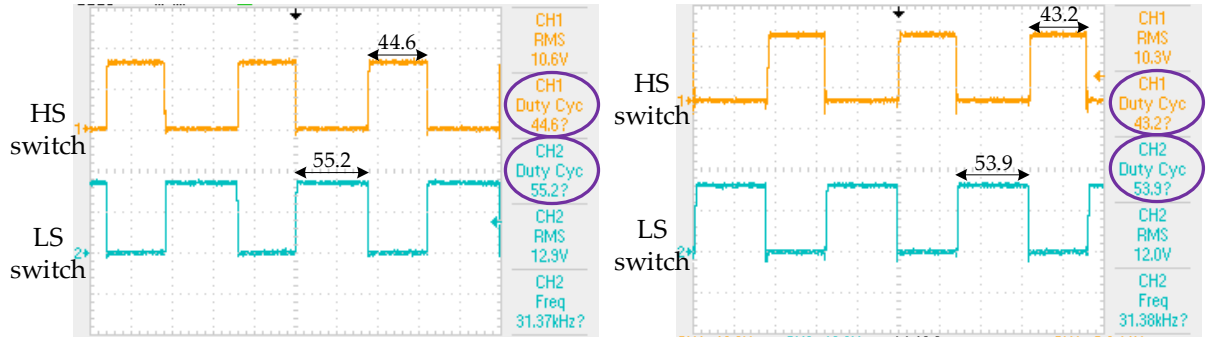


Fig. 9: Overview of the High Side (HS) and Low Side (LS) Switching Signal From the MOSFET Driver (A) Before the Addition of the Dead-Time and (B) After the Addition of the Dead-Time.

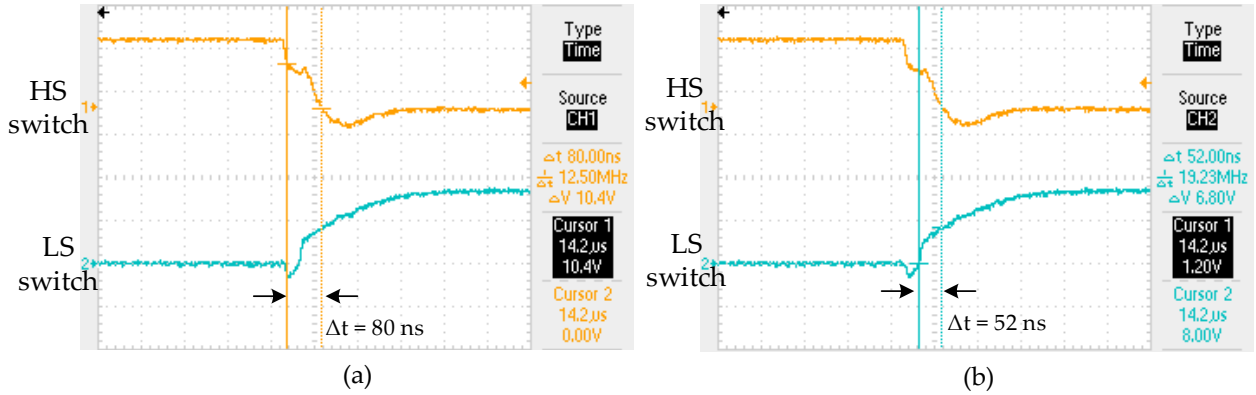


Fig. 10: Falling Edge of the High Side Switch before the Addition of the Delay with (A) Cursor 1 Measuring the Turn OFF Time of the High Side Switch and (B) Cursor 2 Showing the Average Overlap Time between the High Side and the Low Side Switch Signals.

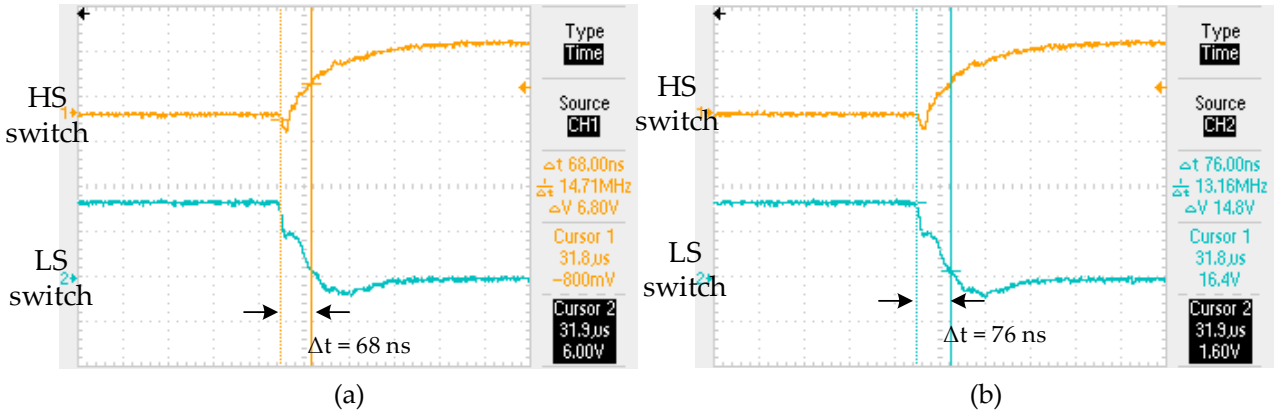


Fig. 11: Rising Edge of the High Side Switch before the Addition of the Delay with (A) Cursor 1 Measuring Average the Turn ON Time of the High Side Switch and (B) Cursor 2 Showing the Average Overlap Time between the High Side and the Low Side Switch Signals.

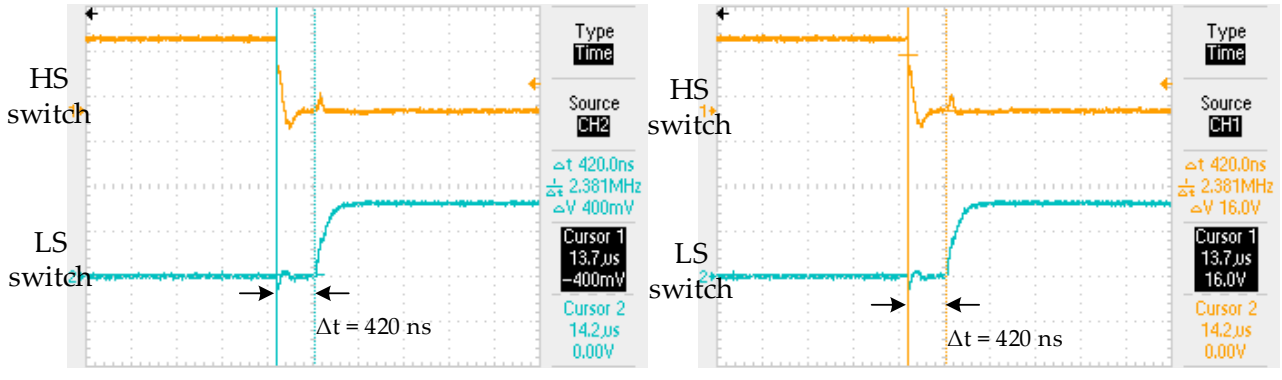
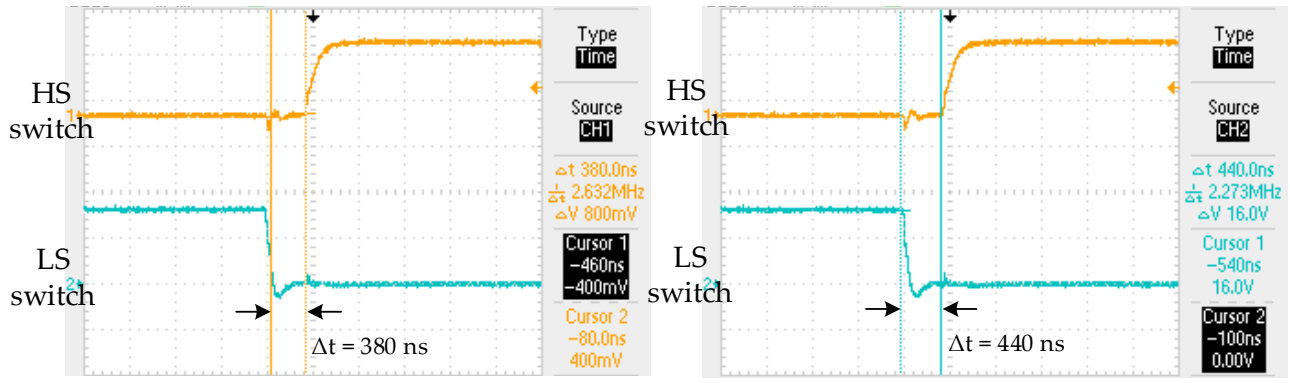


Fig. 12: Falling Edge of the High Side Switch after the Dead-Time Compensation with (A) Cursor 2 Measuring the Average Dead-Time before Turn ON of the Low Side Switch and (B) Cursor 1 Showing the Average Dead-Time between the High Side and the Low Side Switch Signals.



(a) (b)

Fig. 13: Rising Edge of the High Side Switch after the Dead-Time Compensation with (A) Cursor 1 Measuring the Average Dead-Time before Turn on of the High Side Switch and (B) Cursor 2 Showing the Average Dead-Time between the High Side and the Low Side Switch Signals.

5. Conclusion

In this paper, a new method of controlling multiple input DC-DC converters using a single PWM signal from the controller through an FPGA device was proposed. The multiple input converter used for the design was obtained from literature and was discussed briefly with its steady state waveforms presented. A new method of designing the digital logic circuits from the steady state waveforms of each mode of operating the converter was proposed, verified in simulation and validated experimentally. The experimental implementation was achieved without having to program the FPGA device using VHDL, a more complex approach, which is currently the conventional approach. A method of implementing a delay between complementary switch signals was introduced between the complementary switching signals and a delay up to 440 nanoseconds was achieved.

The methods of designing and implementing the digital logic circuits discussed in this paper is not restricted to the multiple input converter presented but can be further extended to other converters with multiple controllable switches in which more than one switch needs to be controlled concurrently. The purpose of this paper was to address the FPGA control by bringing to light a new method of multiple input converter control and this has been achieved.

Acknowledgement

This material is based on research/work supported wholly / in part by the National Research Foundation (NRF) of South Africa (Grant Numbers: 112236). The research findings are that of the authors and not that of the NRF.

References

[1] A. Nahavandi, M. T. Hagh, M. B. B. Sharifian, and S. Danyali, "A nonisolated multiinput multioutput DC-DC boost converter for electric vehicle applications," *IEEE Transactions on Power Electronics*, vol. 30, no. 4, pp. 1818–1835, 2015. https://doi.org/10.1109/TPEL.2014.2325830.

[2] S. Palanidoss and T. V. S. Vishnu, "Experimental analysis of conventional buck and boost converter with integrated dual output converter," in *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, 2017, pp. 323–329. https://doi.org/10.1109/ICEECCOT.2017.8284521.

[3] F. Sobrino-Manzanares and A. Garrigos, "Bidirectional, interleaved, multiphase, multidevice, soft-switching, FPGA-controlled, buck–boost converter with PWM real-time reconfiguration," *IEEE Transactions on Power Electronics*, vol. 33, no. 11, pp. 9710–9721, Nov. 2018. https://doi.org/10.1109/TPEL.2018.2792302.

[4] X. J. X. Jun, Z. X. Z. Xing, Z. C. Z. Chongwei, W. C. W. Chengyue, and L. S. L. Shengyong, "Design of multiple-input DC-DC converter control system for fuel cell electrical vehicle," in *2009 International Conference on Energy and Environment Technology*, 2009, vol. 2, pp. 123–126.

[5] Y. Li, S. Member, D. Yang, X. Ruan, and S. Member, "A Systematic Method for Generating Multiple-Input DC / DC Converters," *IEEE Vehicle Power and Propulsion Conference*, pp. 1–6, 2008.

[6] L. Solero, A. Lidozzi, and J. A. Pomilio, "Design of multiple-input power converter for hybrid vehicles," *IEEE Transactions on Power Electronics*, vol. 20, no. 5, pp. 1007–1016, 2005. https://doi.org/10.1109/TPEL.2005.854020.

[7] S. Bae and A. Kwasinski, "Dynamic modeling and operation strategy for a microgrid with wind and photovoltaic resources," *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 1867–1876, 2012. https://doi.org/10.1109/TSG.2012.2198498.

[8] B. Wang, L. Xian, V. R. K. Kanamarlapudi, K. J. Tseng, A. Ukil, and H. B. Gooi, "A digital method of power-sharing and cross-regulation suppression for single-inductor multiple-input multiple-output DC-DC converter," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 4, pp. 2836–2847, Apr. 2017. https://doi.org/10.1109/TIE.2016.2631438.

[9] R. G. Kale and A. A. Nilangekar, "Implementation of multiple input and multiple output boost converter for electric vehicle charging system," in *2017 International Conference on Nascent Technologies in Engineering (ICNTE)*, 2017, pp. 1–6. https://doi.org/10.1109/ICNTE.2017.7947940.

[10] S. N. Shetty and M. A. Raheman, "Modelling of dual input DC/DC converter for hybrid energy system," in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, 2017, pp. 1152–1155. https://doi.org/10.1109/RTEICT.2017.8256779.

[11] A. Di Napoli, F. Crescimbini, L. Solero, F. Caricchi, and F. G. Capponi, "Multiple-input DC-DC power converter for power-flow management in hybrid vehicles," *Conference Record of the 2002 IEEE Industry Applications Conference. 37th IAS Annual Meeting (Cat. No.02CH37344)*, vol. 3, pp. 1578–1585, 2002. https://doi.org/10.1109/IAS.2002.1043745.

[12] H. Matsuo, T. Shigemizu, F. Kurokawa, and N. Watanabe, "Characteristics of the multiple-input DC-DC converter," in *Proceedings of IEEE Power Electronics Specialist Conference - PESC '93*, 1993, pp. 115–120. https://doi.org/10.1109/PESC.1993.472079.

[13] M. Gavris, O. Cornea, and N. Muntean, "Multiple input DC-DC topologies in renewable energy systems - A general review," *2011 IEEE 3rd International Symposium on Exploitation of Renewable Energy Sources (EXPRES)*, pp. 123–128, 2011. https://doi.org/10.1109/EXPRES.2011.5741805.

[14] M. Forouzesh, Y. P. Siwakoti, S. A. Gorji, F. Blaabjerg, and B. Lehman, "Step-Up DC-DC converters: A comprehensive review of voltage-boosting techniques, topologies, and applications," *IEEE Transactions on Power Electronics*, vol. 32, no. 12, 2017. https://doi.org/10.1109/TPEL.2017.2652318.

[15] Z. Rehman, I. Al-Bahadly, and S. Mukhopadhyay, "Multiinput DC–DC converters in renewable energy applications – An overview," *Renewable and Sustainable Energy Reviews*, vol. 41, pp. 521–539, Jan. 2015. https://doi.org/10.1016/j.rser.2014.08.033.

[16] A. Lavanya, J. D. Navamani, K. Vijayakumar, and R. Rakesh, "Multi-input DC-DC converter topologies-a review," in *International Conference on Electrical, Electronics, and*

Optimization Techniques, ICEEOT 2016, 2016, pp. 2230–2233. https://doi.org/10.1109/ICEEOT.2016.7755089.
[17] M. Truntic and M. Milanovic, "Voltage and current-mode control for a buck-converter based on measured integral values of voltage and current implemented in FPGA," *IEEE Transactions on Power Electronics*, vol. 29, no. 12, pp. 6686–6699, Dec. 2014. https://doi.org/10.1109/TPEL.2014.2301935.
[18] U. Sadek, A. Sarjaš, R. Svečko, and A. Chowdhury, "FPGA-based control of a DC-DC boost converter," *IFAC-PapersOnLine*, vol. 48, no. 10, pp. 22–27, 2015. https://doi.org/10.1016/j.ifacol.2015.08.102.
[19] U. Sadek, A. Sarjaš, A. Chowdhury, and R. Svečko, "FPGA-based optimal robust minimal-order controller structure of a DC–DC converter with Pareto front solution," *Control Engineering Practice*, vol. 55, pp. 149–161, Oct. 2016. https://doi.org/10.1016/j.conengprac.2016.06.016.
[20] M. Milanovic, M. Truntic, and P. Slibar, "FPGA implementation of digital controller for DC-DC buck converter," in *Fifth International Workshop on System-on-Chip for Real-Time Applications (IWSOC'05)*, 2005, pp. 439–443. https://doi.org/10.1109/IWSOC.2005.67.
[21] S. Chander, P. Agarwal, and I. Gupta, "FPGA-based PID controller for DC-DC converter," in *Joint International Conference on Power Electronics, Drives and Energy Systems (PEDES) 2010 Power India*, 2010, pp. 1–6. https://doi.org/10.1109/PEDES.2010.5712454.
[22] S. Kapat and P. T. Krein, "PID controller tuning in a DC-DC converter: A geometric approach for minimum transient recovery time," in *2010 IEEE 12th Workshop on Control and Modeling for Power Electronics (COMPEL)*, 2010, pp. 1–6. https://doi.org/10.1109/COMPEL.2010.5562367.
[23] D. Pellerin and D. Taylor, *VHDL Made Easy!* Prentice Hall PTR, 1996.
[24] D. Pellerin, E. A. Thibault, and S. Thibault, *Practical FPGA Programming in C*. Prentice Hall PTR, 2005.
[25] D. O. Neacsu, *Switching Power Converters: Medium and High Power, Second Edition*. CRC Press, 2017.
[26] G. M. Dousoky, M. Shoyama, and T. Ninomiya, "FPGA-based spread-spectrum schemes for conducted-noise mitigation in DC–DC power converters: Design, implementation, and experimental investigation," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 2, pp. 429–435, Feb. 2011. https://doi.org/10.1109/TIE.2010.2049708.
[27] P. Zumel, C. Fernandez, M. Sanz, A. Lazaro, and A. Barrado, "Step-by-step design of an FPGA-based digital compensator for DC/DC converters oriented to an introductory course," *IEEE Transactions on Education*, vol. 54, no. 4, pp. 599–609, Nov. 2011. https://doi.org/10.1109/TE.2010.2100397
[28] J. M. Blanes, R. Gutierrez, A. Garrigos, J. L. Lizan, and J. M. Cuadrado, "Electric vehicle battery life extension using ultracapacitors and an FPGA controlled interleaved buck–boost converter," *IEEE Transactions on Power Electronics*, vol. 28, no. 12, pp. 5940–5948, Dec. 2013. https://doi.org/10.1109/TPEL.2013.2255316.
[29] E. Jamshidpour, P. Poure, and S. Saadate, "Photovoltaic systems reliability improvement by real-time FPGA-based switch failure diagnosis and fault-tolerant DC–DC converter," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 11, pp. 7247–7255, Nov. 2015. https://doi.org/10.1109/TIE.2015.2421880.
[30] S. Natarajan and R. Natarajan, "An FPGA chaos-based PWM technique combined with simple passive filter for effective EMI spectral peak reduction in DC-DC converter," *Advances in Power Electronics*, vol. 2014, pp. 1–11, 2014. https://doi.org/10.1155/2014/383089.
[31] E. Guzman Ramirez, I. Garcia, E. Guerrero, and C. Pacheco, "A tool for supporting the design of DC-DC converters through FPGA-based experiments," *IEEE Latin America Transactions*, vol. 14, no. 1, pp. 289–296, Jan. 2016. https://doi.org/10.1109/TLA.2016.7430091.
[32] A. Hintz, U. R. Prasanna, and K. Rajashekara, "Novel modular multiple-input bidirectional DC-DC power converter (MIPC)," in *2014 International Power Electronics Conference (IPEC-Hiroshima 2014 - ECCE ASIA)*, 2014, pp. 2343–2350. https://doi.org/10.1109/IPEC.2014.6869917.
[33] I. N. Jiya, N. Gurusinghe, and R. Gouws, "Hybridization of Battery, Supercapacitor and Hybrid Capacitor for Electric Vehicles," in *2018 IEEE PES-IAS PowerAfrica Conference*, 2018, pp. 351–356.