Performance comparison of DC-DC controllers with FPGA

Ben Jeppesen

Intel, UK

Corresponding author: Ben Jeppesen, ben.jeppesen@intel.com

Abstract

Electrification of vehicles and smart grids requires efficient power converters. This paper compares the dynamic performance and efficiency of different control schemes applied to a bi-directional buck-boost converter with a specified loading cycle. FPGA-based high-frequency hysteresis (hysteretic) control is proposed to achieve higher efficiency through reduced switching losses. PWM and hysteresis control are implemented on an FPGA using model-based design. It is shown how optimization of numeric precision with automatic code generation can be used to reduce resources used on the FPGA, and how the design can be extended to multiple inductor phases.

1 Introduction

1.1 FPGA control of power converters

Switch-mode power converters are used in many electronic devices. Most are small buck converters, stepping down an unregulated supply to a reliable DC voltage. However, DC-DC converters are being used increasingly to control higher-power systems in applications like HVDC distribution, energy storage or energy flow in electric vehicles. These systems operate continuously for long periods of time, have widely varying loads and bidirectional operation, requiring sophisticated controllers to achieve voltage regulation and energy efficiency.

FPGAs (Field-Programmable Gate Arrays) are a programmable logic technology that enables complex digital circuits to be realized, such as custom 'system-on-chips' with interfacing IP, DSP functions and a soft or hard processor in the chip. FPGAs are often used in research of power converters because the control functions can be executed faster in logic than in standard microprocessors.

In [1], we showed how FPGA-based hysteresis control can be used to reduce the number of switching events compared to PWM control, thereby improving efficiency. The flexibility and parallel processing capability of FPGAs also enables other approaches like multi-phase converters, resonant converters, synchronized control of multiple inverters, charge pumps, switched capacitor networks, advanced digital control, varying voltage requests and special monitoring and shutdown sequences.

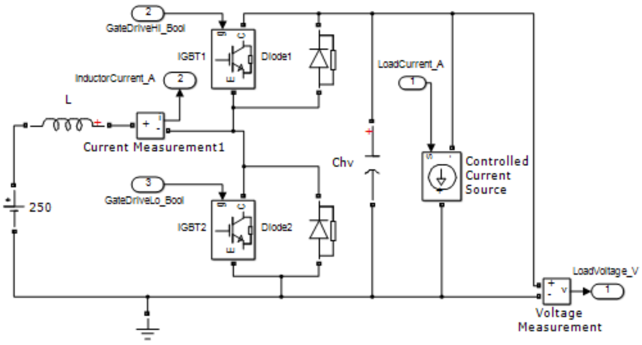


Fig. 1: Single phase converter model.

1.2 Aims of the paper

In this paper, we compare the performance of different controllers on a simulation of a bidirectional buck-boost converter. We show how to implement the controllers in an FPGA using model-based design and automatic code generation, including custom numeric precision to reduce FPGA resources, and extension to multi-phase converters

2 Theory

2.1 Operation of a switching converter

Linear voltage regulators use MOSFET transistors in continuous conduction mode, which achieves very stable voltage output. They often use a 'bandgap reference' voltage and provide a fixed output voltage commonly needed by other devices. However, this mode of regulation is relatively inefficient. For example, a linear regulator with 5V input and 2.5V output would only be 50% efficient.

Higher power voltage regulators use switching circuits where MOSFETs or IGBTs act as switches to alternatively charge and discharge other components. These are often inductors, as in the single-phase bi-directional converter shown in Fig. 1, but can also be capacitors, as used in charge pumps or switched capacitor circuits [2].

Switched capacitor circuits can be efficient but are limited in their voltage ratios to integer fractions and are not as accurate as linear regulators.

Inductor-based circuits are efficient and widely used but generate voltage ripple with each transistor switching event and in response to sudden load current changes. There are many variations of switch mode circuits, such as multi-phase circuits or multi-level designs, and different control methods with the aim of reducing ripple or improving efficiency.

2.2 Competing requirements

Different requirements drive different design choices:

- Low cost is achieved by fewer switching transistors, smaller passive components and cheaper established control technology. Linear regulators may be combined with switch-mode converters to achieve good output control with acceptable efficiency [3][4];
- Small size in higher power converters requires efficient transistors to reduce cooling requirements but these can be more expensive. Small size in lower powered converters can be achieved by integrating the inductor, transistors and control in a single chip [5];
- High efficiency can be achieved by a few very efficient transistors, a more complex multi-phase or multi-level design needing a larger number of cheaper transistors, or larger passive components so fewer switching events are needed.
- Low ripple requires faster switching, larger passive components (inductor, capacitor) or faster control methods.
- Other qualities may also be required, such as automotive certification or packaging.

2.3 Control method

Many voltage regulators use voltage-mode control, where the output voltage is compared to a reference voltage and a PI controller is used to set the duty cycle of a PWM waveform which drives the transistor gate signals. This may require additional slope compensation for stability due to the nonlinearity of the circuit and the voltage measurement must be filtered to average out ripple, limiting the speed of response. A current measurement is also usually required to trigger shutdown on overcurrent.

Current-mode regulation is an alternative where the voltage PI regulator generates a demand for the inductor current. A PWM waveform is used where the first edge is driven by the clock pulse and the second by a comparator which triggers when the demand current has been achieved. This type of control is robust and fast acting but requires a fast and smooth current measurement.

Hysteretic or hysteresis control does not use a fixed PWM frequency but switches the circuit when the achieved voltage or current falls outside a tolerance band compared to the demand value. This can give accurate control since it acts faster and more efficient control due to fewer switching events but requires fast and accurate measurements and produces different electromagnetic noise spectra compared to a fixed PWM frequency.

Some useful references are [6][7][8].

2.4 Requirements for SiC/GaN MOSFETs

New generation 'wide bandgap' MOSFETs using SiC and GaN technology have very fast switching capability so have low switching losses. They are enabling power electronics in applications like traction drives and micro power generation to use much faster switching than with traditional IGBT transistors, for example changing from around 10kHz to 100s of kHz. If 10 bits of resolution (2^10 = 1024 bit) are required on the PWM waveform edges, this requires 100s of MHz on the PWM edge-generating clock.

2.5 FPGA value

Self-contained power supply chips contain their own control electronics, which are usually analog. High frequency PWM and fast response can be achieved but the control operation such as PWM frequency and control gains have limited adjustment.

Microcontrollers often have PWM generation IP which can be combined with software algorithms to decide the PWM duty cycle. Resolution of the PWM edges can be good due to the combination of analog waveform generation with digital parameters, but methods are still often limited to PWM at a fixed frequency, so current-mode and hysteresis control are not practical.

FPGAs provide fully configurable digital control of all transistor gate signals at high frequencies with flexible interfacing to sensors. There are many advantages to FPGAs, for example:

Choice of switching strategies

- PWM (fixed or variable frequency)
- Hysteresis control
- Other switching strategies, e.g. pulse density control

Integration of several functions in a custom System-on-Chip:

- Lower part count
- Control power converters and motor control in the same chip for synchronized control strategies, for example vary the DC link voltage to save power
- All digital parameters and signals are available to software

Model-based design

- The signal flows through FPGAs are natural to describe in a block-based programming language such as MATLAB Simulink, making it easy to simulate algorithms and generate efficient code.
- Off-the-shelf code generation tools are available like MathWorks HDL Coder or DSP Builder for Intel FPGAs [9]
- Simulink models can be used to generate testbenches for hardware simulation software such as ModelSim to verify the generated code.

Achievable clock frequencies in FPGAs can limit the PWM edge resolution compared to analog PWM. The configurability of FPGAs makes it a more expensive technology, so there is a need to make efficient use of FPGA logic. This paper aims to address both these issues.

3 Method

3.1 DC-DC control development

The simulation uses MATLAB/Simulink with SimPowerSystems to include realistic models of IGBT switches (Fig. 1).

The same test waveform for the load current and load voltage command is used for each controller Fig. 2):

- A stepped DC link voltage command is used to create switch-on and switch-off transients
- Stepped load currents are used to explore control performance across the operating range of the load current.

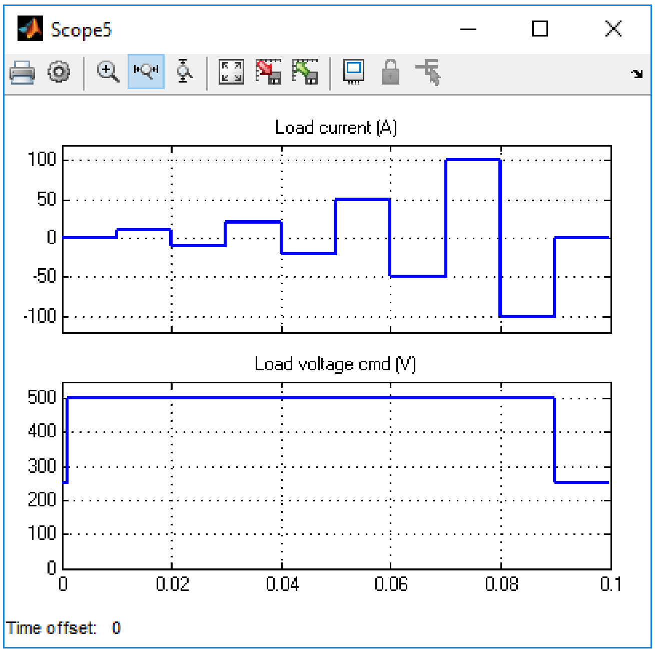


Fig. 2: Input test waveforms.

Control performance is compared using these measures:

- Worst voltage error
- Converter energy loss. This must be calculated by integrating energy flows in both directions during the simulation

Different control algorithms will be compared:

1. Voltage PI control with inductor current PI+PWM control (the original method from the Intel DC-DC Converter reference design [10], with an inner loop controlling inductor current.
2. Voltage PI control with inductor current hysteresis control: The output of the voltage PI controller is the inductor current command, which is delivered by a hysteresis controller.

The algorithms are developed in standard Simulink blocks, then converted to use DSP Builder Advanced Blockset (Fig. 3) to enable simulation at the FPGA clock frequency, automatic FPGA HDL code generation and ModelSim test bench generation. The power electronic simulation can also be converted to HDL code so the complete system can be simulated in real-time on the FPGA.

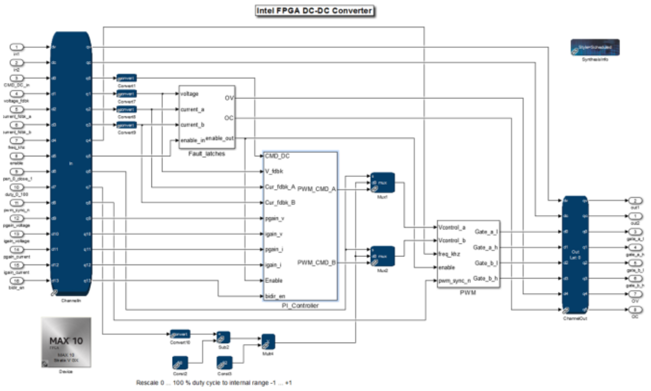


Fig. 3: DC-DC controller with PWM in DSP Builder.

3.2 Testing on FPGA with simulated power electronics

Intel FPGA's DC-DC converter reference design [10] shows how the FPGA algorithm can be tested on the FPGA hardware in real-time at the intended FPGA clock rates, by simulated the changing load and behavior of the power electronics. Both the control algorithm and electronic simulation code can be generated automatically using DSP Builder. However, there is still the challenge of how to view the real-time data to check the operation. Two options are to:

1. Send real-time data to DACs on the development board for connection to an external oscilloscope
2. Add FPGA IP (functions composed of logic and memory) to capture signals of interest and transfer them digitally to a PC.

3.2.1 Sending signals to analog outputs

Intel FPGA's DC-DC Converter Reference Design shows how to connect outputs to DACs on the Arrow BeMicro Max10, FPGA Evaluation Kit [11]. This can be done similarly on other kits, for example you can connect outputs to the DAC8551 on the Intel MAX 10 Development Kit [12].

3.2.2 Log signals at high frequency using SignalTap Logic Analyzer

Real-time data can be visualized using the Intel SignalTap Logic Analyzer software and FPGA IP [13]. A SignalTap IP is included in the FPGA design to monitor the signals of interest. Then, when the FPGA is running, blocks of sampled signals are captured and downloaded over JTAG and USB to be visualized interactively on a PC. Data can be logic or integer signals, so graphs of high-frequency signals can be generated.

3.3 Optimizing designs

FPGA resource usage can be reduced by considering port widths, numeric precision, and reformulating calculations. See for example "Exploring Design Tradeoffs" in [14].

3.4 Testing with real power electronics

Finally, the algorithm can be applied to a real two-phase bi-directional DC-DC converter in Intel's Drive-on-Chip reference design (Fig. 4)[15][16].

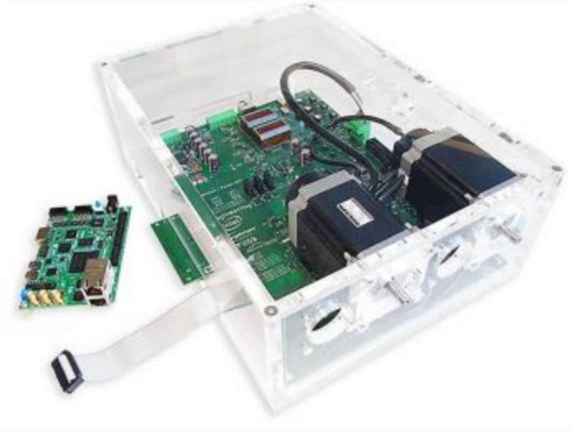


Fig. 4: Tandem Motion-Power 48 V kit showing FPGA control of a DC-DC converter

3.5 High-frequency PWM for MOSFET control

There are several methods to increase the clock frequency that can be used to generate PWM waveform edges in the FPGA.

3.5.1 Optimization of logic in the FPGA fabric

By simplifying logic and adding pipeline registers to store intermediate results between clock pulses, logic on the FPGA can be run at frequencies approaching the maximum available from the FPGA clock trees. For example, the MAX 10 FPGA clock trees can run at 382-450 MHz depending on the device specification (table 26 in [17]).

3.5.2 Using double-date rate I/O

LVDS I/O pins are designed to be able to run at double the FPGA clock frequency [18]. They are fed with multiple-bit data per FPGA clock pulse using soft SERDES IP that uses some FPGA logic. For example, MAX 10 can support LVDS transmission from 640Mb/s to 720Mb/s depending on the device (table 26 in [17]).

3.5.3 Using clocks with different phases

The PLLs (Phase-Locked Loops) which generate FPGA clock pulses can generate clocks with 90-degree phase shifts. Therefore, it is possible to multiplex logic driven by four differently phased clocks to achieve resolution of 4 times the logic clock frequency at an I/O pin. This would require custom logic like the SERDES IP but could enable PWM edge resolution of less than 1 ns.

3.5.4 Using transceivers

Some FPGA families include I/O pins driven by transceivers at GHz frequencies. These can be used to achieve PWM edge resolution less than 1ns. For example, Cyclone V FPGAs can transmit at data rates from 2500Mb/s to 5000Mb/s depending on the device (table 23, [19]). However, these also require SERDES logic and the transceiver I/O may need amplification (level shifting) to drive power electronics.

4 Results

4.1 Performance of hysteresis current control

Performance measurements have been collected from different controllers. Details of the controllers were reported in [1]. All controllers have an outer voltage control loop and inner current control loop. Two electrical circuits are used, one intended for 10kHz PWM and one for 50kHz PWM. The current control loop uses either PWM control or hysteresis control with 30A or 60A tolerance bands. Hysteresis control with revised voltage PI gains was found to improve both voltage control and efficiency, for both 10kHz and 50kHz converters.

4.1.1 Simulation for energy calculations

Fig. 5 shows typical results from a simulation driven by a load test waveform like Fig. 2. Electrical power is calculated during the simulation and then numerically integrated to calculate energy and hence efficiency.

Note how ‘energy loss’ rises quickly at the beginning as the output capacitors are charged, and how energy is recouped near the end (0.09s), when the voltage command is dropped to 0V again, discharging the capacitors.

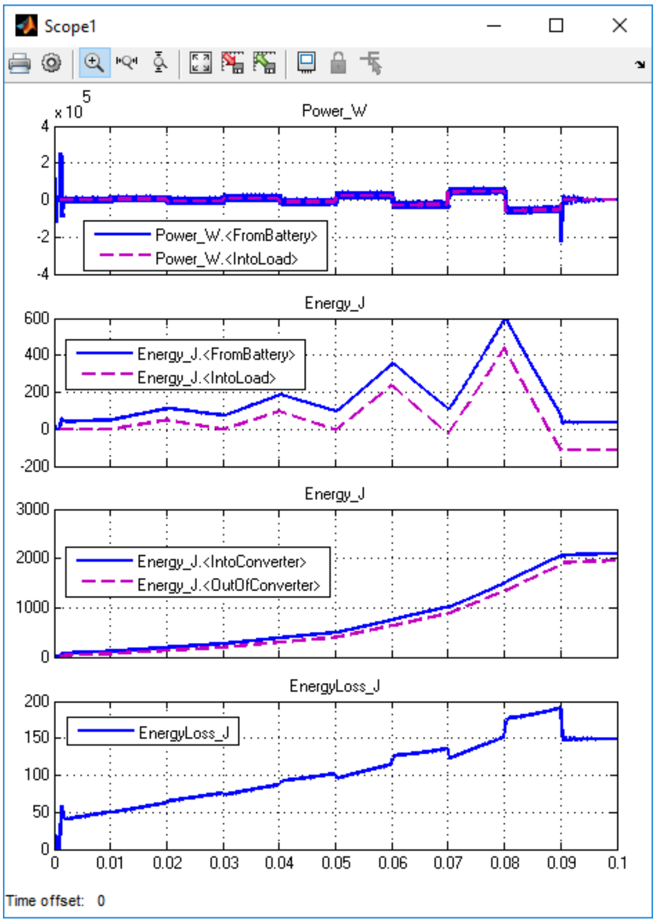


Fig. 5: Example simulation result, 50kHz PWM with 60A hysteresis band.

4.1.2 Energy efficiency

The results for energy efficiency show how efficiency improves with fewer switching events, either by reducing the PWM frequency or increasing the hysteresis tolerance band (Fig. 6).

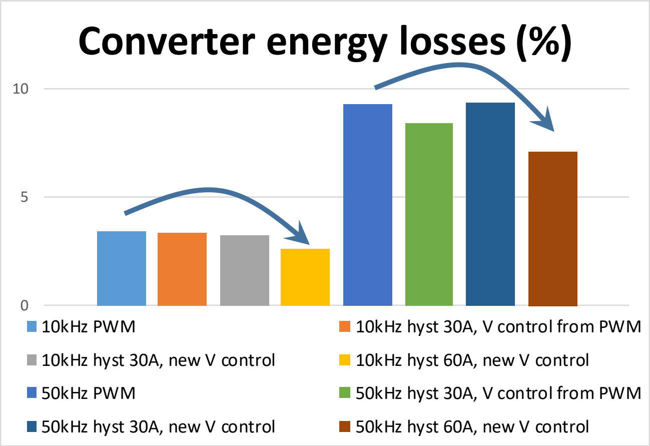


Fig. 6: Converter energy losses depending on control method

4.1.3 Voltage control

Voltage control performance is judged by the worst-case voltage error seen during the simulation. Fig. 7 shows that hysteresis control gives similarly good results in most configurations, especially where new voltage control gains are introduced.

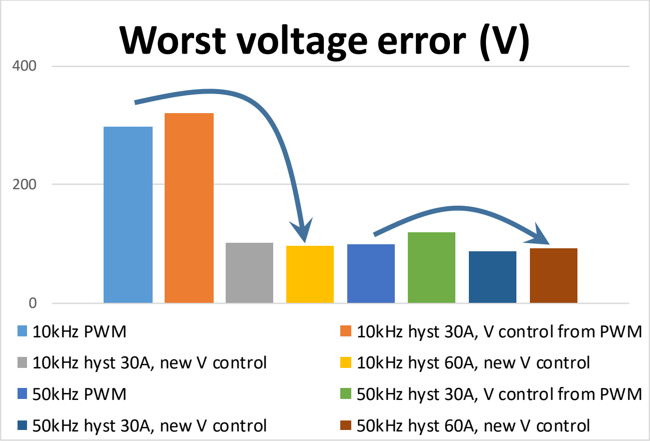


Fig. 7: Converter worst voltage error depending on control method

4.1.4 FPGA resources needed

Table 1: MAX 10 FPGA at 50MHz with 8kLE, resources

Controller	**PWM**	**Hysteresis**
Logic Elements	3,225 (40%)	2,755 (34%)
DSP (9x9)	24 (50%)	18 (38%)
Memory (M9K blocks)	3 (7%)	0 (0%)

The FPGA resources for the compiled code can be compared for the designs based on PWM and hysteresis control. Table 1 shows that the hysteresis controller design uses fewer resources than the original PWM controller, because the hysteresis controller requires simpler logic.

4.2 FPGA resource usage in DC-DC control with multi-phase PWM

FPGA resource usage has been improved by simplifying an original design [15] from 2-phase to one phase, optimizing by using custom numeric types sufficient for the required numeric precision and then extending back to multiple phases, as shown in Table 2. In each case, one PI controller was used for the voltage control loop and a single counter-based PWM carrier signal was used for all phases. However, a separate PI current controller was used for each phase so currents are actively balanced across all phases. The results shown were for a MAX 10 FPGA with 10MHz clock.

Table 2: MAX 10 FPGA resources for multi-phase converters

Phases	Logic Elements	DSP Elements
Original 2	4658	48
1	1705	12
2	2317	16
4	3367	24
6	4679	32

Comparing the results shows that for the optimized designs:

- Around 525 Logic Elements and 4 DSP elements are needed for each additional phase.
- Around 1180 Logic Elements and 8 DSP elements are needed for the outer loop voltage control and PWM waveform generation.

4.3 Achieving high-frequency PWM

4.3.1 PWM in FPGA fabric only

The 6-output PWM IP running in Intel's Drive-on-Chip reference design has been designed to run at 333 MHz on MAX 10 [16], approaching the maximum FPGA clock frequency.

4.3.2 PWM using double-data-rate IP

A design example has been created that achieves 500 MHz PWM edge resolution with 125MHz clock in the FPGA logic, by using counter-based PWM waveform generation in combination with LVDS I/O and soft SERDES IP.

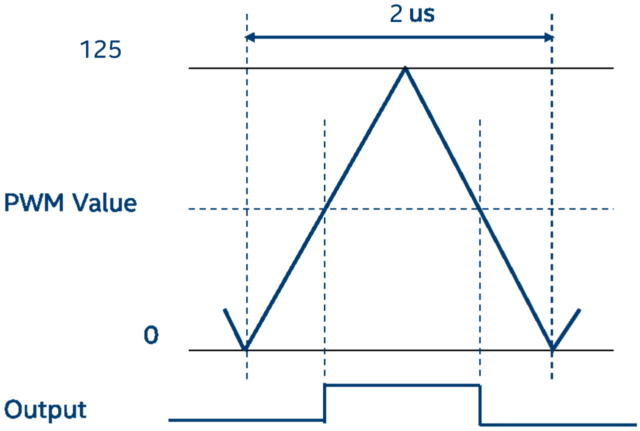


Fig. 8: 500kHZ PWM waveform generation

Fig. 8 shows how the PWM waveform is generated. An up-and-down counter from zero to 125 (achievable with an 8-bit counter) is compared with an instantaneous threshold value from the control algorithm. By comparing a new threshold on the up and down parts of the counter, 1MHz control updates can be achieved with the 500kHz waveform.

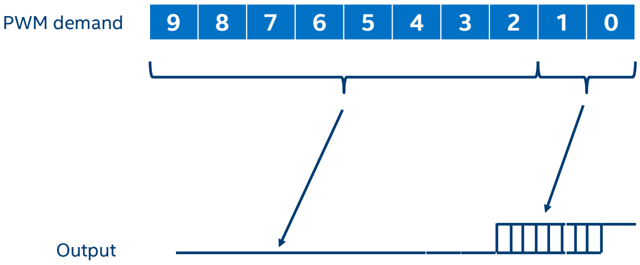


Fig. 9: PWM demand value, 8 bits for counter threshold plus 2 bits for SERDES

Fig. 9 shows how the 10-bit demand value is used: the first 8-bits are used for a threshold to compare to the PWM counter, while the final two bits are used to request the increased resolution from the SERDES IP.

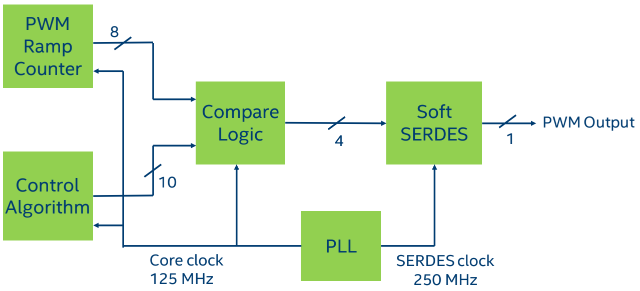


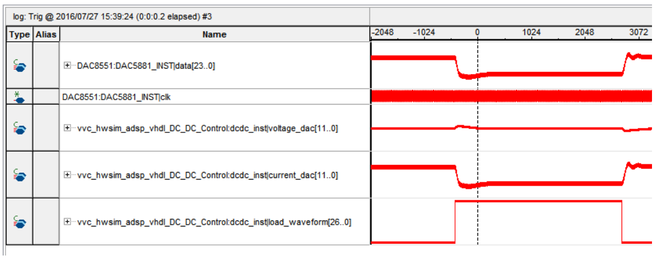
Fig. 10: Schematic of FPGA logic design with signal bitwidths

Fig. 10 shows a schematic of the logic design needed. Each SERDES IP is given a 4-bit signal at a sample rate of 125MHz, which it transmits as a 1-bit signal at 500MHz.

4.4 Testing on FPGA with SignalTap

4.4.1 Logging signals at high frequency using SignalTap

As an example, Intel's single-phase DC-DC Converter Reference design has been modified to add Signal Tap IP to visualize signals on a PC. Based on the memory buffer available and length of time needed to visualize the signals, a sample rate of 400kHz was chosen. A PLL was added to generate a clock at 400kHz to sample the internal signals. Fig. 11 shows example output which can be verified by comparing to data from the Simulink and DSP Builder test simulations.


log: Trig @ 2016/07/27 15:39:24 (0:0:0.2 elapsed) #3

Type
Alias
Name
-2048
-1024
0
1024
2048
3072
DAC8551:DAC5881_INST|data[23..0]
DAC8551:DAC5881_INST|clk
vvc_hwsim_adsp_vhdl_DC_DC_Control:dcdc_inst|voltage_dac[11..0]
vvc_hwsim_adsp_vhdl_DC_DC_Control:dcdc_inst|current_dac[11..0]
vvc_hwsim_adsp_vhdl_DC_DC_Control:dcdc_inst|load_waveform[26..0]

Fig. 11: Output from FPGA captured in SignalTap. Signals are (top to bottom): output voltage, inductor current and load current.

5 Conclusions

- Model-based design has been used to design FPGA-based controllers of a bi-directional DC-DC converter.
- FPGA-based hysteresis control can improve performance and efficiency.
- Control code can be optimized to fit multi-phase DC-DC control into a low-cost FPGA.
- High-frequency, high-resolution PWM waveform generation suitable for SiC and GaN MOSFETS can be achieved on an FPGA by either:
  - optimizing code to approach the FPGA fabric maximum clock frequency or
  - combining logic in the fabric with hard IP such as LVDS I/O, multi-phase PLLs or transceivers.

6 References

[1] B.Jeppesen, G.Gao, J.Katcha: Use of FPGAs to develop energy-saving DC-DC control, PCIM 2017, Nuremberg, Germany

[2] Enpirion EC2650QI 6A Voltage Divider, https://www.intel.co.uk/content/dam/www/programmable/us/en/pdfs/products/Power/ec2650qi-datasheet.pdf

[3] Enpirion EZ6301QI Triple Output Module, https://www.intel.com/content/dam/altera-www/global/en_US/pdfs/literature/ds/ez6301qi_13926.pdf

[4] Bosch Multiple Power Supply – CY320, http://www.bosch-semiconductors.com/media/automotive_systems_ics/pdf_1/ic_engine_management/bosch_ic_cy320.pdf

[5] Enpirion EP5358xUI 600mA PowerSoC, https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ds/ep5358xui_03541.pdf

[6] Microsemi Technical Note TN-203: Voltage-Mode, Current-Mode (and Hysteretic Control), 2012, https://www.microsemi.com/document-portal/doc_view/124786-voltage-mode-current-mode-and-hysteretic-control

[7] Texas Instruments, Rais Miftakhutdinov, “Synchronous buck regulator design using the TI TPS5211 high-frequency hysteretic controller”, 1999, http://www.ti.com/lit/an/slyt186/slyt186.pdf

[8] S. Saggini, et al, “Fully-digital hysteretic voltage-mode control for dc-dc converters based on asynchronous sampling”, APEC 2008. Twenty-Third Annual IEEE

[9] Introduction to DSP Builder for Intel FPGAs, https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/dspb/hb_dspb_intro.pdf

[10] DC-DC Converter Control with ADSP Builder: https://fpgacloud.intel.com/devstore/platform/16.0.0/Standard/dc-dc-converter-control-with-adsp-builder/

[11] Arrow, BeMicro Max10, FPGA Evaluation Kit, https://www.arrow.com/en/reference-designs/bemicro-max10-fpga-evaluation-kit-adopts-alteras-non-volatile-max-10-fpga-built-on-55-nm-flash-process/45cdf7d717bc14e6dfb1044001fb7c4f

[12] Intel FPGA MAX 10 Development Kit, https://www.intel.com/content/www/us/en/programmable/products/boards_and_kits/dev-kits/altera/max-10-fpga-development-kit.html

[13] SignalTap User’s Guide, https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/signal.pdf

[14] DSP Builder Handbook, Volume 3: DSP Builder Advanced Blockset, https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug_dsp_builder_adv.pdf

[15] B.Jeppesen, A.Crosland, T.Chau: An FPGA-based platform for integrated power and motion control, IEEE IECON 2016, Florence, Italy, October 2016

[16] Intel FPGA Application Note AN 773: Drive-On-Chip Reference Design for MAX® 10 Devices, https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/an/an773.pdf

[17] Intel MAX 10 FPGA Device Datasheet, https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/max-10/m10_datasheet.pdf

[18] Intel® MAX® 10 High-Speed LVDS I/O User Guide, https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/max-10/ug_m10_lvds.pdf

[19] Cyclone V FPGA Device Datasheet, https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/cyclone-v/cv_51002.pdf