

High-frequency, Three-phase Current Controller Implementation in an FPGA

M. Hartmann, S.D. Round and J.W. Kolar

ETH Zurich
Physikstrasse 3
Zurich, Switzerland

hartmann@lem.ee.ethz.ch

Abstract- Three phase rectifiers with switching frequencies of 500 kHz or more require high speed current controllers. At such high switching frequencies analog controllers as well as high speed DSP-systems have limited performance. In this paper, two high speed current controller implementations using two different FPGAs - one for switching frequencies up to 1 MHz and one for switching frequencies beyond 1 MHz - are presented to overcome this performance limitation. Starting with the digital system design all the blocks of the signal chain, containing A/D interface, digital controller implementation using HW-multipliers and implementation of a novel high speed, high resolution PWM are discussed and compared. Final measurements verify the performance of the controllers.

I. INTRODUCTION

Modern active rectifiers offer the advantages of unity power factor, high compactness, high efficiency and sinusoidal input currents. The three-level Vienna Rectifier (VR) topology [1] is an ideal choice for telecom and data server power supplies or aerospace applications [2]. Aerospace applications are challenging for current controller realization due to the high input frequency of 360 Hz - 800 Hz. Especially for aerospace applications a high power density is required. An increase in power density requires an increase in switching frequency from today's industry standard of 20 kHz to over 500 kHz. Current control of such high switching frequency three-phase converters poses a challenge. A classical analog approach for current control shows several problems, such as limited accuracy and configurability [3], [4]. Using purely digital current control, by application of a modern high speed DSP, is a good way to control these rectifiers for medium and high switching frequencies ($f_s = 20 \text{ kHz} \dots 200 \text{ kHz}$) [5]. However, for very high switching frequencies of over 1 MHz even these digital systems reach their limit and a current controller can not be realized using just a single DSP. The key to overcome this limitation is "parallelization". This can be implemented by using one DSP for each phase of the three-phase system, but the number of instructions executed between the sampling instants is still limited.

In the realizations presented in this paper for a three phase rectifier (cf. Fig. 1), modern high-speed FPGAs are used, which exhibit intrinsic parallelization. Today's high-speed FPGAs contain DSP-blocks including hardware multipliers that can be used to implement the control algorithm. The effort to implement a controller for switching frequencies above 1 MHz increases rapidly. Hence two implementations

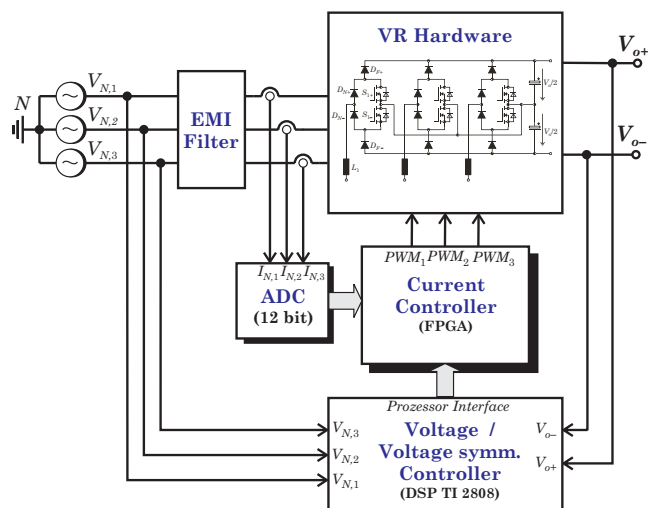


Fig. 1: Structure of the proposed digital current controller for the three-phase six-switch (Vienna-type) rectifier system.

using FPGAs from different vendors are presented. For the first realization (C1) a maximal switching frequency of 1 MHz is intended and an FPGA from Lattice Corp. (ECP2) is used. The second realization (C2) is for switching frequencies above 1 MHz and a high-speed FPGA from Xilinx Corp. (VIRTEX4) is used.

An overview of the implemented systems will be given in Section II. In Section III the design of the control loop, considering the whole signal chain, is discussed. The implementation of the current controller in the two FPGAs including a comparison of performance and chip utilization is the focus of Section IV. In Section V measurement results taken from a laboratory prototype verify the performance of the controller.

II. SYSTEM OVERVIEW

In Fig. 1 the structure of the proposed digital current controller for the three-phase VR-system, operating at a switching frequency of 1 MHz is shown. The digital high-speed current controller is implemented by application of a modern high-speed FPGA. For realization C1 the FPGA

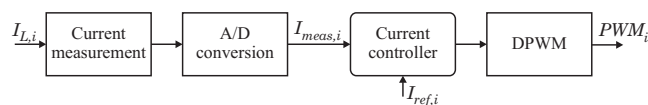


Fig. 2: Single-phase signal chain of the digital current controller for a three-phase current controller.

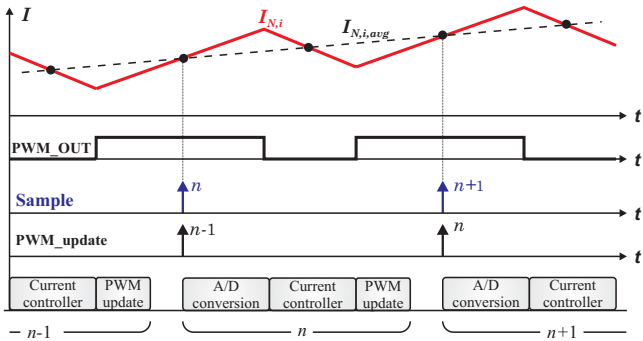


Fig. 3: Timing diagram of the sampling strategy. The current signal is sampled in the middle of the PWM. After the A/D conversion time the duty cycle, d , is calculated by the current controller and updated at the start of the next PWM-cycle.

ECP2 LFE2-12E-6T144I [6] with a speed-grade 6 from Lattice is used. Realization C2, to implement the highest possible switching frequency, uses the VIRTEX4 XC4-VLX25-10FF668 FPGA with a speed-grade 10 from Xilinx [7]. There are no analog to digital converters (ADCs) included in the FPGAs, which means that an external A/D-converter is required. The main limitation for the ADC is its delay time (time between sampling instant and availability of the sampled data), which is also influenced by the interface type of the ADC. Therefore, also two different ADCs with different interface types are used for the two realizations.

For the full rectifier system an output voltage and output voltage symmetry controller is needed. Since the timing requirements for these controllers are not so stringent a single DSP can be used to control these system parameters. The interface between the DSP (voltage controller) and the FPGA (current controller) can either be realized by a serial or a parallel interface. For the presented realizations a serial SPI interface is implemented, where the intrinsic DSP-functionality can be used.

In order to be able to implement a three-phase rectifier system with a switching frequency of over 1 MHz the whole signal chain, containing of current measurement, A/D conversion, digital controller and high speed DPWM-generation has to be considered (cf. **Fig. 2**). The realization of a current measurement with a bandwidth of [DC...>1 MHz] poses a challenge but will not be discussed in this work. However, some approaches can be found in [8] and [9].

III. SYSTEM DESIGN

As a first step, the timing requirements on the system-blocks have to be defined. A timing diagram of the sampling strategy (of one phase current) is shown in **Fig. 3**. Additional to the average current I_{avg} of the rectifier, a current ripple is present. As it is well known, the sample value $I[n]$ is equivalent to the average current value I_{avg} , if the sampling point is located in the middle of one PWM-period. Additional a large ringing occurs in the current signal near the switching instants. A possible influence of the current measurement through this ringing can also be minimized by this sampling strategy. Hence, synchronization between PWM-generation and sampling of the currents is essential. This synchronization can be

realized by a center aligned (symmetrical) PWM-generation. Additional the center aligned PWM-generation offers the advantage of reduced input current distortions in the intended VR application [4], therefore, a center aligned PWM is used.

After sampling the actual current value $I_{N,i}[n]$, the ADC requires some time to convert the sampled analog value into a digital word. This result is used by the current controller to calculate the new duty cycle of the PWM-signal. At sampling instant $n+1$ the PWM-value is updated with the new duty cycle. This sampling strategy is also known as uniformly sampled modulation [10]. Hence, according to **Fig. 3** the whole calculation, consisting of A/D conversion, calculation of current controller and update of the PWM, has to be done in one switching-cycle. This delay (deadtime) of one cycle has to be included in the controller design and reduces the phase margin of the controller loop considerably. For lower switching frequencies, a double update of the PWM is possible, which increases the stability of the current controller.

A. A/D Conversion

In a second step, an appropriate ADC has to be chosen. Due to the small amplitude of the ripple current a high dynamic range of the ADC is needed. Therefore a converter with a resolution of 12 bits or more may be required. However, the main selection criterion is its delay, because ADCs with high sampling rates normally have a relatively long delay between sampling and availability of the data. A/D-converters with sampling-frequencies of 3 MSa/s are commercially available with a simple SPI interface and such a converter is used for the controller realization C1. For higher sampling frequencies only ADCs with parallel- or LVDS-interface [11] are available. Consequently a (pipelined) ADC with a sampling-frequency of $f_s = 25$ MHz and a LVDS-interface is used for controller realization C2. The main specifications of the used A/D converters are listed in **TABLE I**. The LVDS interface in general is well suited for power electronic applications because it is based on current drivers and differential signaling. The physical effort of realizing a LVDS interface is much higher than a simple SPI-Interface. Indeed, the selected FPGAs are able to directly drive the LVDS signals without any additional components but impedance and length matched PCB tracks have to be realized. Additional, as discussed in section IV, the digital implementation of the LVDS-interface in the FPGA is much more complex. However, the serial LVDS interface is the only alternative to a parallel interface if very high switching frequencies of $f_s > 1$ MHz are to be realized.

TABLE I:
SPECIFICATIONS OF THE USED A/D CONVERTERS.

	Controller C1 AD7274	Controller C2 ADS5240
Sampling frequency	1 MSa/s	25 MSa/s
Resolution	12 bit	12 bit
Converters / Package	1	4
Interface	SPI	LVDS
Delay	448 ns	300 ns

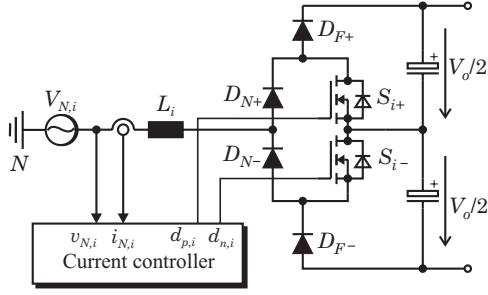


Fig. 5: One phase-leg of the three-level six-switch rectifier.

B. Controller Design

For the design of a digital controller two different strategies are possible. One strategy is to design an appropriate analog controller and convert it into the digital domain. The other strategy, known as direct digital design, is to directly design the controller in the z-domain. The direct digital design method, results in a slightly better controller performance with regard to phase margin and achievable bandwidth [9]. On the other hand, digital redesign offers the capability to use the well known design methods of the continuous-time implementations. Hence, a digital redesign of an analog controller will be used in this work. For that purpose an analog model of the rectifier has to be built. In this work average mode control will be used, which means that all signals are averaged over one switching period. According to Fig. 5 each switch (of the two switches per phase) is only involved for half a period of the mains voltage. Using the duty-cycle d_i the inductor voltage balance results in

$$v_{N,i}(t) - \frac{v_o(t)}{2}(1 - |d_i(t)|) = L \frac{di_{N,i}(t)}{dt}. \quad (1)$$

The output voltage $v_o(t)$ of the rectifier is assumed to be constant $v_o(t) = V_o/2$, which is accomplished by the output voltage controller. Using the feed-forward function

$$d_{res,i} = d_{ff,i} + d_i, \quad d_{ff,i} = 1 - \frac{v_{N,i}(t)}{V_o/2} \quad (2)$$

and by application of the Laplace transform on (1) results in the simple model

$$G(s) = \frac{i_{N,i}(s)}{d_i(s)} = \frac{V_o}{2Ls} \quad (3)$$

for the rectifier. In this model some details, such as the impedance of the mains, the characteristics of the EMI-filter, the delay times of the switches or the characteristic right hand zero of the boost circuit are not considered for sake of simplicity. Previous work on a 400 kHz VR system [5]

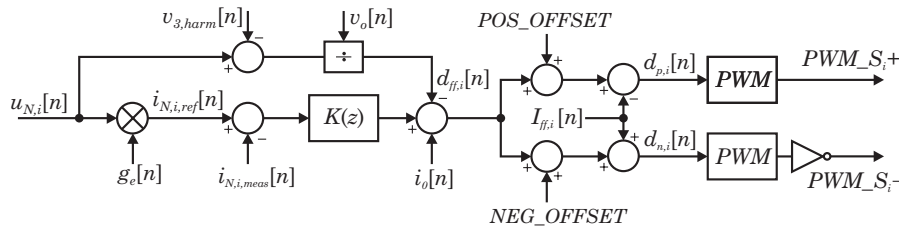


Fig. 6: Structure of the digital current controller for one phase.

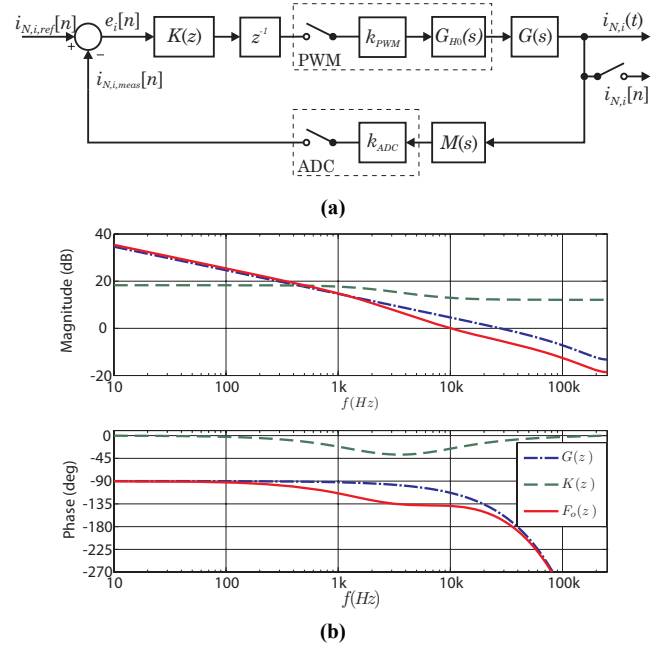


Fig. 4: Design of the digital current controller; (a) control loop and (b) bode plot of the digital current controller. The designed controller has a phase margin of 40° .

showed that a simple P+Lag controller

$$K(s) = K_p \frac{1 + sT_D}{1 + sT_1} \quad (4)$$

in conjunction with the voltage feed-forward function of (2) is a good solution. For the design of the analog controller the delay of the symmetrical PWM $G_{PWM}(s)$ [13] and the delay caused by the controller calculation $G_{calc}(s)$, which can be modelled as

$$G_{PWM}(s) = e^{-sT_s/2}, \quad G_{calc}(s) = e^{-sT_s}, \quad (5)$$

where T_s is one switching period, have to be considered. For discretization of the analog controller given in (4), the bilinear (Tustin) transformation is used, which results in

$$K(z) = K \frac{1 - k_1 z^{-1}}{1 - k_2 z^{-1}}. \quad (6)$$

The behaviour of the designed controller can be verified in the z-domain. A block diagram of the control loop including the digital controller is given in Fig. 4(a). According to [14] the symmetrical PWM is modelled by a sample and hold element $G_{H0}(s)$. Additionally the bandwidth limitation of the current sensor is included in the model by $M(s)$. In Fig. 4(b) a Bode plot of the resulting control loop is depicted, where a phase margin of 40° can be found.

For the final controller implementation several other parts have to be added to the current controller. The resulting

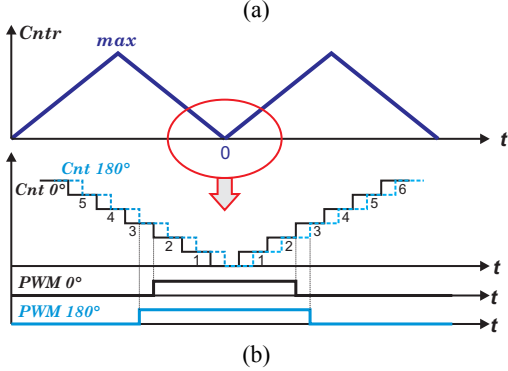
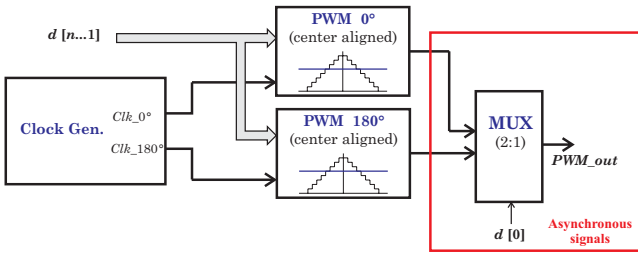


Fig. 7: Proposed digital PWM-concept for a center aligned PWM modulator; (a) Structure of the concept and (b) generation of the PWM-patterns for a comparator value of $d=3$.

structure of the digital current controller is shown in **Fig. 6**. In order to increase the modulation range of the rectifier a 3rd harmonic signal v_{3harm} is added [15] and a zero sequence component i_0 , generated from the voltage symmetry controller, is used to balance the output voltages [16]. CoolMOS devices are used for realization of the switches. Unfortunately these devices exhibit, apart from their excellent conduction behaviour, a relative large turn-off delay at lower current levels [17]. This results in increased current zero-crossing distortions, especially at the intended high switching frequencies [18]. In order to attenuate this effect a current feed-forward signal $i_{ff,i}$ is used to compensate the delay. There are two PWM units necessary to generate the gate signals for the two switches, where the PWM-signal of the switch S_i has to be inverted. The POS_OFFSET and NEG_OFFSET signals are used to realize the required voltage feed-forward signal of (2).

C. PWM-Generator

The generation of the symmetrical high speed PWM-signals also has to be implemented inside the FPGA since there is no external PWM modulator readily available for such high frequencies. There are several possibilities known in literature to implement a digital high-speed PWM using an FPGA. The performance of a “classical” counter/comparator approach is limited due to the very high clock-frequencies required to implement a high frequency PWM. In [19] a FPGA implementation using a hybrid delay-line/counter scheme is described. However, this implementation is only applicable to trailing edge modulators, shows a strong temperature dependence and is not easy to implement. In [20] it was shown, that the resolution of a counter/comparator approach can be increased by application of phase shifted clocks. However, this realization is also limited to trailing edge-modulators. In this work a novel PWM-modulator using two phase-shifted clocks is used (cf. **Fig. 7**). The modulator consists of

TABLE II
COMPARATOR VALUES d FOR THE PROPOSED PWM-MODULATOR.

	PWM 0° = 1	PWM 180° = 1
Count up	$Cntr < d$	$Cntr < d$
Count down	$Cntr < d$	$Cntr < d+1$

two different counters/comparators running with a phase shift of 180°. The least significant bit (LSB) selects, if either the PWM signal with 180° phase shift or the PWM signal without phase shift is used. A combination of the two signals is used to increase the resolution of the PWM-modulator for 1 bit. This is done by changing the comparator value d of the 180° phase shifted modulator at the two instants, where the counter direction has to be turned ($Cntr=0$, $Cntr=max$). The necessary comparator values d are summarized in **TABLE II** and the resulting PWM-patterns for $d=3$ are plotted in **Fig. 7** (b). The two phase shifted signals have to be treated as asynchronous signals in the FPGA and therefore timing and routing constraints have to be used in the design implementation.

IV. IMPLEMENTATION OF THE SYSTEMS

In this section the two implementations of the current controllers using the two FPGAs are discussed. Even if the implementations maybe seem to be trimmed to fit the selected FPGAs, nearly all other FPGA-vendors offer FPGAs with similar functionality.

An overview of the controller implementation, listing all implemented blocks, is shown in **Fig. 8**. All three current controllers (one for each phase) are processed in parallel. Additional system parts, like the interface to the DSP, system management and clock generation for the digital PWM are also shown. The implementation of the whole system is fully written in VHDL. To synthesize and fit the implemented controller, the free tools from the vendors are used (ispLEVER Starter for implementation C1 and Xilinx ISE Webpack for the implementation C2). Both implementations are running with an internal system clock of 125 MHz. The data from the superimposed voltage controller g_e and voltage symmetry controller i_0 are sent from the DSP by a serial SPI-interface. This interface is not discussed further in this work.

D. Implementation of the ADC-interface

According to section II the sampling instants have to be synchronized to the PWM-generation. Therefore the PWM-module generates the start of conversion signal for the ADC.

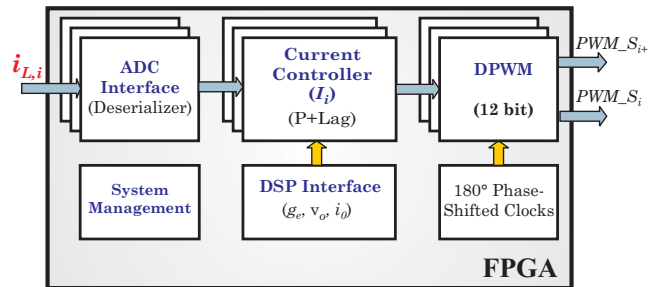


Fig. 8: Overview of the controller implementation in the FPGA.

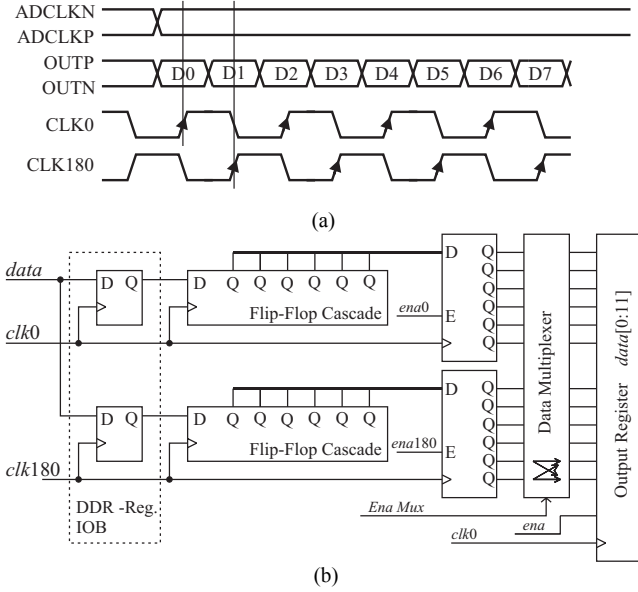


Fig. 9: LVDS-interface to the ADCs according to [21]; (a) Timing of the interface. The data is sampled at the rising and falling edge of the clock signal and (b) overview of the implementation in the FPGA.

Implementation C1:

The realization of a SPI-Interface in an FPGA is relatively simple. The serial data-stream has to be deserialized, which can easily be done by a simple shift register. The ADC-clock of 31.25 MHz is generated in the FPGA by application of a PLL. In general there are two possibilities – either the (asynchronous) ADC-clock is used as clock-signal for the ADC interface in the FPGA or the ADC-signals (clock and data) are synchronized to the internal system clock. In general multiple clock domains should be avoided in FPGAs whenever this is possible, so synchronization is the better way. However, several fast clock-nets are available in the used FPGA and so the external ADC-clock is used in the realization which results in a very easy implementation. Of course, the ADC-data has to be synchronized to the system clock after deserialization.

Implementation C2:

In contrast to the very easy SPI-Interface of realization C1, the LVDS-interface implementation of realization C2 is much more complex. The ADC-interface is implemented according to an application note of Xilinx [21] and it seemed to be straight forward but several issues occurred, which will be discussed now. The used Xilinx FPGA offers the capability to drive LVDS signals without any external components. The sampled data is transferred at the rising and falling edge of the differential clock signal (cf. **Fig. 9(a)**). To deserialize this serial data stream, the internal DDR-input registers (clock signal on positive and negative edges) of the FPGA are used. In this realization the ADC ADS5240 runs with a sampling frequency of 25 MHz. This high sampling frequency in conjunction with a resolution of 12 bit results in a serial data rate of 300 Mbit/s. To handle such high data rates inside of the FPGA care has to be taken with the signal timing. Therefore, defined timing and routing constraints are needed for a successful implementation, which requires a detailed knowledge of the

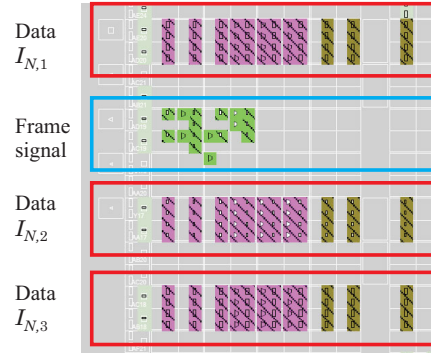


Fig. 10: Resulting placement of the ADC-Interface (current inputs) in the Xilinx-FPGA.

TABLE III
SUMMARY OF THE ADC-INTERFACE IMPLEMENTATION IN THE FPGAS.

	Slices	$f_{clk,max}$
C1 (ECP2-LFE2-12E)	43	180 MHz
C2 (XC4VLX25)	47	248.16 MHz

FPGA-Slices. Special attention has to be laid on the generation of the *Ena_Mux* signal since there, from the FPGAs point of view, asynchronous signals are matched together to one data word. The resulting routing of the interface is shown in

Fig. 10. In an FPGA data and clock inputs show slightly different delays and therefore the phase of the clock signal has to be adjusted dynamically. Special clock management blocks are available inside of the FPGA for that purpose. But unfortunately this clock management blocks seem not to work well in the desired frequency range although the frequencies still fulfill the specs. Therefore this phase adjustment is not implemented and as a result the maximal sampling frequency is limited to 25 MSa/s (although the maximal sampling frequency of the ADC is 40 MSa/s).

Compared to the simple SPI interface, the LVDS implementation requires much more detailed knowledge of the used FPGA but offers very high data rates and a very small delay between sampling and availability of the data. A summary of the used logic cells and achieved timing is given in **TABLE III**.

E. Controller Implementation

The controller given in (1) yields to the control algorithm

$$\begin{aligned}
 e[n] &= i_{ref}[n] - i_{meas}[n] \\
 u[n] &= K(e[n] - k_1 e[n-1]) + k_2 u[n-1], \quad (7) \\
 d[n] &= u[n] + d_{ff}[n] + i_{ff}[n]
 \end{aligned}$$

where $d_{ff}[n]$ and $i_{ff}[n]$ are the voltage and current feed-forward parts. There are several possibilities to realize this control algorithm inside of an FPGA. A realization of a 12x12 bit multiplication using normal logic units of the FPGA is possible, but requires pretty much logic cells and shows a limited timing. Alternatively the multiplications can be avoided by application of shift operators, but then the possibilities of choosing K , k_1 and k_2 are limited. To overcome this limitation modern FPGAs offer so-called DSP-blocks that include hardware multipliers. These

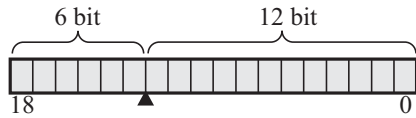


Fig. 11: Fractional number representation of K, k_1 , and k_2 in the FPGA

multipliers are able to process an 18x18 bit (signed) multiplication in one cycle and offer additional functionality such as multiplication and addition or multiplication and accumulation. These blocks are typically able to run with clock-frequencies of over 300 MHz, if all multiplier stages are pipelined. For the controller implementation these blocks are used. The DSP-blocks can be targeted in a number of ways, where the most promising way are upcoming tools, which allow to design the system in Matlab Simulink® and convert it into HDL-code. Unfortunately, these tools are currently only applicable for low clock-frequencies and therefore the DSP-blocks are directly inferred by VHDL-code.

In a first step, after getting the data from the ADC, the data has to be transferred from binary offset to two's complement number representation. This can be done by a simple inversion of the most significant bit of the ADC data word. An 18 bit signed number representation is used in the FPGA because of the required 18 bit numbers for the DSP-blocks, but the 12 bit range is not extended to 18 bits. On one hand this has the advantage that no overflows can occur, on the other hand accuracy is given away. For the controller constants K, k_1 and k_2 a fractional number representation as given in Fig. 11 is used. After the multiplication the remainder is thrown away.

The implementation of the controller in an FPGA differs from an implementation in a DSP, because all calculations are done in parallel in the FPGA. But this does not mean that enable signals have to be generated for every block. As an example the implementation of the P+Lag current-controller of (7) using the HW multipliers is shown in Fig. 12. For a high-frequency implementation it is very important that all stages are pipelined. According to Fig. 12 the result $u[n]$ is valid after 5 clock cycles and it only has to be ensured that the result is not used for PWM-generation before these 5 clock cycles are completed.

Since the controller implementation is more or less the same in the two FPGAs no comparison will be made for this part. The delay of the full current controller is 128 ns for both implementations.

F. PWM-Generation

For implementation of the digital PWM of Section III a clock frequency of 250 MHz is used. A higher clock frequency is not possible even with the high-speed FPGA of implementation C2.

Implementation C1:

For this implementation the counter/comparator approach without phase shift is used. Unfortunately this results in a PWM-resolution of only 7 bits at a switching frequency of 1 MHz. For the implementation of the required 7 bit high speed counters the VHDL data-type *std_logic_vector* has to be used, because it is much faster than the VHDL data type

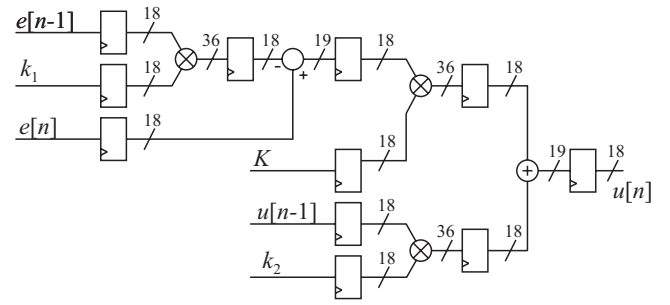


Fig. 12: Implementation of the P+lag current controller for one phase in the FPGAs using the HW-multiplier of the FPGA. All the multipliers are pipelined by registers for highest possible throughput.

integer. For each modulator an own counter is used which further increases the speed. It has to be said, that a detailed statement about maximal clock frequency of the implementation can only be given if the whole system is fixed by routing constraints. Otherwise the timing is influenced by other system parts. For the realization at hand only timing constraints are used. The results of the implementation are although given in TABLE IV.

Implementation C2:

For C2 the resolution is increased to 8 bits by implementation of the proposed PWM-concept. The 180° phase shifted 250 MHz-clock is generated by the digital clock management module (DCM) of the FPGA. Also here, the two phase shifted modulators have to be treated as asynchronous signals in the FPGA so that adding timing and routing constraints to the FPGA design are essential. Special attention has to be laid on the connection of the modulators output to the multiplexer and further on to the output pin, because the multiplexer has no register. This means, that different delays of these signal paths directly result in a phase difference of the two PWM-signals. Unfortunately, there exists no timing constraint to define the same routing delay for several signal paths. Therefore manually placing of the output-register and the multiplexer has to be used instead. With that, delay-differences under 50 ps could be achieved. A drawback of the proposed PWM-concept is that glitches of the multiplexer can occur. But in the desired application several “slow” components (optocoupler, gate-driver) would absorb these glitches.

For lower switching frequencies (i.e. $f_s = 500$ kHz) the duty cycle can be updated twice a period to minimize the delay of the PWM modulator.

G. Summary of the Implementation

The two designs have been successfully fitted into the two FPGAs. A summary of the report is given in TABLE V. The utilization of the DSP-blocks (one DSP-block contains several multipliers) of 68% and 87% means in practice nearly a full usage, because there are some mapping limitations of the multipliers into the DSP blocks which

TABLE IV
SUMMARY OF THE PWM- IMPLEMENTATION IN THE FPGAS.

	Slices	$f_{clk,max}$
C1 (ECP2-LFE2-12E)	232	253.16 MHz
C2 (XC4VLX25)	712	255.36 MHz

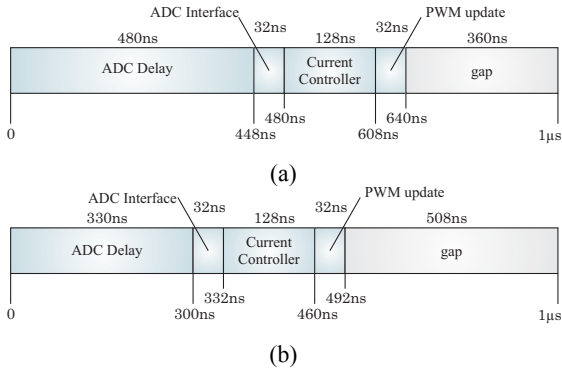


Fig. 15: Resulting timings of (a) current controller implementation C1 and (b) current controller implementation C2 for a switching frequency of 1 MHz.

TABLE V
REPORT SUMMARY OF THE CONTROLLER IMPLEMENTATIONS.

	C1 (ECP2-LFE2-12E)	C2 (XC4VLX25)
DSP-block utilization	87 %	68 %
Logic utilization	25 %	10 %
Clock management	50 %	40 %
I/O Pins	41 %	8 %

inhibit a usage of 100%. In contrast to the utilization of the DSP-blocks only a small amount of the available logic cells and I/O Pins are used for the controller implementation. This clearly illustrates that the speed of the FPGA is needed and not it's size. The resulting timings of the implementations for a switching frequency of 1 MHz are depicted in Fig. 15. The biggest part of the processing time is the ADC-delay, even if a pipelined ADC with a sampling frequency of tens of MHz is used. The second largest part of delay is introduced by the controller implementation caused by pipelining of all the multiplier stages. According to Fig. 15(b) the controller implementation C2 is able to handle switching frequencies of 2MHz.

V. EXPERIMENTAL RESULTS

To verify the function of the implemented digital current controller a laboratory prototype of a VR according to the following key specifications is constructed (cf. Fig. 13):

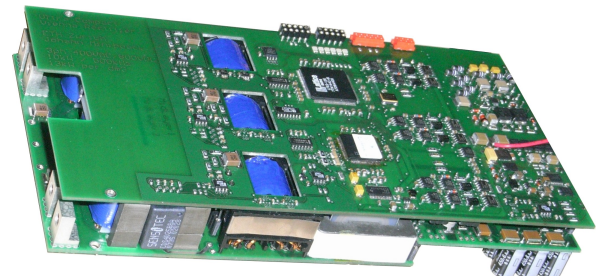
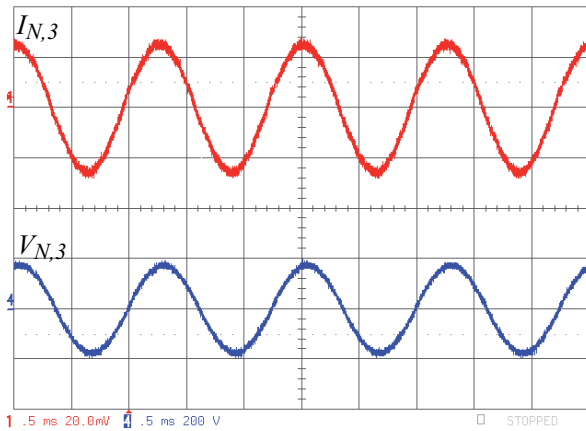


Fig. 13: Picture of the realized Vienna Rectifier prototype designed for a power level of 10kW with controller implementation C1.

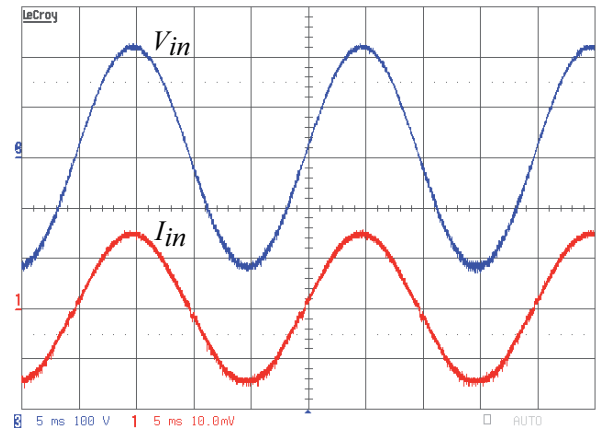
- input voltage: $v_{N,i} = 230 \text{ VRMS}$
- input frequency: $f_{in} = 50/60 \text{ Hz}, 360 - 800 \text{ Hz}$
(different controller gains required)
- output voltage: $V_o = 800 \text{ VDC}$
- output power: $P_o = 10 \text{ kW}$
- switching frequency: $f_s = 500 \text{ kHz} / 1 \text{ MHz}$
- boost inductance: $L = 40 \mu\text{H}$

Controller implementation C1 is used for current control of the rectifier. The system is operated at a switching frequency of 500 kHz but double update of the PWM is implemented and therefore the controller has to deal with a control sampling frequency of 1 MHz. A measurement result of the input current $I_{N,3}$ and input voltage $V_{N,3}$, taken from the VR-prototype at a line frequency of 800 Hz, is shown in Fig. 14(a). As can be seen the input current has a sinusoidal shape and is in phase with the input voltage. The THD of the phase-current at a power level of 5 kW is 1.2% and confirms the very good performance of the implemented current controller. The controller implementation C2 has been tested on a single phase PFC prototype with a switching frequency of 150 kHz by application of a Xilinx evaluation board [22]. For that purpose the controller-structure has not been changed, only the internal clock-frequencies have been reduced to handle the lower operating frequency. The measurement result, which also confirms the good performance, is given in Fig. 14 (b).

The behavior of the proposed DPWM is tested using the evaluation board (cf. Fig. 16). The PWM has a frequency of



(a)



(b)

Fig. 14: Measurement result of the two controller implementations. (a) Measurements taken from the rectifier prototype employing the controller implementation C1 operating at an input frequency of $f_{in} = 800 \text{ Hz}$ and a power level of 3 kW. Ch1: 10 A/Div, Ch2: 200 V/Div; timebase: 0.5 ms/Div. (b) Measurements taken from the single phase PFC employing the controller implementation C2. Ch1: I_{in} 5 A/Div, Ch2: V_{in} 100 V/Div; timebase: 5 ms/Div

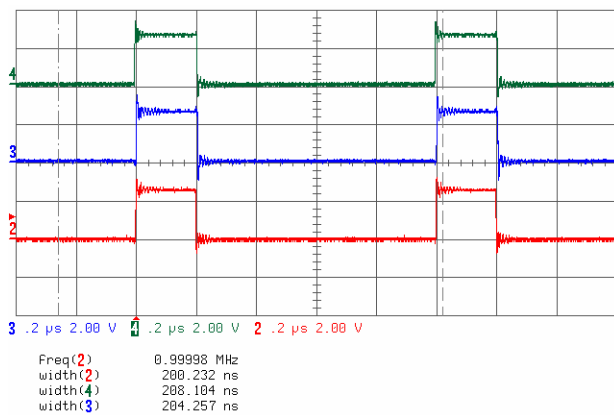


Fig. 16: Measurement result of the proposed PWM-concept with a PWM-frequency of 1 MHz and the PWM-values: Ch2 (2 V/Div): $d=50$ (200 ns); Ch3 (2 V/Div): $d=51$ (204 ns) and Ch4 (2 V/Div): $d=52$ (208 ns); timebase: 200 ns/Div.

1 MHz and all three PWM-channels differ only in 1 bit. According to Fig. 16, a PWM-value of $d=50$ results in a pulse-width of 200 ns for an 8 bit PWM with a maximum counter-value of 125, and a PWM-value of $d=51$ results in a pulse-width of 204 ns.

VI. CONCLUSION

Two digital current controllers for a 3-phase rectifier operating at 1MHz have been successfully implemented by application of modern high-speed FPGAs. The entire signal chain, starting with the current measurement and ending with the PWM-generation has been considered. The way to minimize the delay introduced by the A/D-converter has been shown by application of two different high speed A/D-converter implementations. A control model of the three phase rectifier has been developed, which was used to design a digital current controller. All necessary steps to implement the controller in the FPGA have been discussed. For the implementation of the current controllers the hardware multipliers, of the FPGAs have been used. An increased PWM resolution has been achieved by application of an enhanced PWM-generation, based on the classical counter comparator approach. The good performance of the current controller is verified by measurements taken from a 10kW rectifier prototype.

REFERENCES

- [1] J.W. Kolar and F.C. Zach, "A novel three-phase utility interface minimizing line current harmonics of high-power telecommunications rectifier modules," *IEEE Trans. on Industrial Applications*, Vol. 44, No. 4, Aug. 1997, pp. 456-467.
- [2] G. Gong, M.L. Heldwein, U. Drogenik, J. Miniboeck, K. Mino, J.W. Kolar, "Comparative evaluation of three-phase high-power-factor AC-DC converter concepts for application in future More Electric Aircraft," *IEEE Transactions on Industrial Electronics*, vol.52, no.3, June 2005, pp. 727-737.
- [3] J. Miniboeck, J.W. Kolar, "Novel concept for mains voltage proportional input current shaping of a VIENNA rectifier eliminating controller multipliers," *IEEE Transactions on Industrial Electronics*, Vol.52, No.1, Feb. 2005, pp. 162-170.
- [4] U. Drogenik, J.W. Kolar, "Comparison of not synchronized sawtooth carrier and synchronized triangular carrier phase current control for the VIENNA rectifier I," *Proc. of the IEEE International Symposium on Industrial Electronics (ISIE '99)*, Bled, Slovenia, Vol.1, 1999, pp.13-19.
- [5] P. Karutz, S.D. Round, M.L. Heldwein, J.W. Kolar, "Ultra Compact Three-phase PWM Rectifier," *Proc. of the 22nd Annual IEEE Applied Power Electronics Conference, (APEC 2007)*, Feb. 25 - March 1, pp.816-822.
- [6] Lattice "Lattice ECP2/M Family Datasheet," www.latticesemi.com (2008).
- [7] Xilinx, "Virtex IV - User Guide," www.xilinx.com (2008).
- [8] L. Dalessandro, N. Karrer, J.W. Kolar, "High-Performance Planar Isolated Current Sensor for Power Electronics Applications," *IEEE Transactions on Power Electronics*, Vol.22, No.5, Sept. 2007, pp.1682-1692.
- [9] G. Laimer, J.W. Kolar, "Design and experimental analysis of a DC to 1 MHz closed loop magnetoresistive current sensor," *20th Annual IEEE Applied Power Electronics Conference and Exposition, (APEC 2005)*, Vol.2, March 2005, pp.1288-1292.
- [10] R.D. Middlebrook, "Predicting modulator phase lag in PWM converter feedback loops," *Powercon 8*, Paper H-4, Apr. 27-30, 1981.
- [11] National Semiconductor, "LVDS Owner's manual, 4th edition (2008)," www.national.com/lvds (2008).
- [12] Y. Duan, H. Jin, "Digital controller design for switchmode power converters," *14th Annual Applied Power Electronics Conference and Exposition, (APEC '99)*, Vol.2, 14-18 Mar 1999, pp.967-973.
- [13] D.M. Van de Sype, K. De Gussemme, A.P. Van den Bossche, J.A. Melkebeek, "Small-signal Laplace-domain analysis of uniformly-sampled pulse-width modulators," *Proc. of the 35th Annual IEEE Power Electronics Specialists Conference (PESC 04)*, 20-25 June 2004, vol.6, pp. 4292-4298.
- [14] D.M. Van de Sype, K. De Gussemme, A.R. Van den Bossche, J.A. Melkebeek, "Small-signal z-domain analysis of digitally controlled converters," *Proc. of the 35th Annual IEEE Power Electronics Specialists Conference (PESC 04)*, 20-25 June 2004, vol.6, pp. 4299-4305.
- [15] F. Stogerer, J. Miniboeck, J.W. Kolar, "Implementation of a novel control concept for reliable operation of a VIENNA rectifier under heavily unbalanced mains voltage conditions," *Proc. of the 32nd Annual IEEE Power Electronics Specialists Conference (PESC 2000)*, vol.3, pp.1333-1338.
- [16] J.W. Kolar, U. Drogenik, F.C. Zach, "Current handling capability of the neutral point of a three-phase/switch/level boost-type PWM (VIENNA) rectifier," *Proc. of the 27th Annual IEEE Power Electronics Specialists Conference, (PESC '96)*, 23-27 Jun 1996, vol.2, pp.1329-1336.
- [17] L. Lorenz, G. Deboy, I. Zverev, "Matched pair of CoolMOS transistor with SiC-Schottky diode - advantages in application," *IEEE Transactions on Industry Applications*, Sept.-Oct. 2004, vol.40, no.5, pp. 1265-1272.
- [18] S.D. Round, P. Karutz, M.L. Heldwein, J.W. Kolar, "Towards a 30 kW/liter, Three-Phase Unity Power Factor Rectifier," *Proc. of the Power Conversion Conference - Nagoya, (PCC '07)*, 2-5 April 2007, pp.1251-1259.
- [19] R.F. Foley, R.C. Kavanagh, W.P. Marnane, M.G. Egan, "An area-efficient digital pulsewidth modulation architecture suitable for FPGA implementation," *Proc. of the 20th Annual Applied Power Electronics Conference and Exposition, (APEC 2005)*, 6-10 March 2005, vol.3, pp. 1412-1418.
- [20] A. de Castro, G. Sutter, S.C. Huerta, J.A. Cobos, "High Resolution Pulse Width Modulators in FPGA," *3rd Southern Conference on Programmable Logic, (SPL '07)*, 28-26 Feb. 2007, pp.137-142.
- [21] Xilinx, "A Virtex-4 and Virtex-5 FPGA Interface for Texas Instruments Analog-to-Digital Converters," Xilinx Application Note XAPP866, July 10, 2007.
- [22] Xilinx, "ML401/ML402/ML403 Evaluation Platform, User Guide," www.xilinx.com (2008).