System on Chip Cores for Protection and Commutation of a Matrix Converter

Enekoitz Ormaetxea*, Jon Andreu*, Iñigo Kortabarria*, Iñigo Martínez de Alegría*, Edorta Ibarra*, Ekaitz Olaguenaga†
*Univ. of the Basque Country, Dep. of Electronics and Telecomm., Bilbao, Spain, e-mail: *jon.andreu@ehu.es*
†TECNALIA, Energy Unit, Parque Tecnológico Zamudio, Spain, e-mail: *pedro@robotiker.es*

***Abstract*—This article deals with the implementation of the commutation and protection of the matrix converter (*MC*) via some *hardware* blocks (*cores*) implemented in an *FPGA*. Thanks to the computational capacity of the cores, processing time is reduced. This allows a response in real time improving the safety of the *MC*.**

***Index Terms*—Matrix Converter, System on Chip (SoC), Field Programmable Gate Array (FPGA), Core.**

I. Introduction

Matrix Converter [1] (Fig. 1) is a very promising technology due to: it is an *all-silicon AC/AC* converter, with no dc-link requirement, low volume and compact design [2]; it presents bidirectional power flow [3]; it is possible to achieve sinusoidal currents in the grid and sinusoidal voltages at the load [4], with a unity power factor [5], having a low harmonic distortion [6]; and, lastly, it has a long lifetime with high-temperature and high and low pressure surroundings due to the absence of the electrolytic capacitors.

The *MC* is very attractive, but its industrial acceptance has been held back due to the fact that there are a number of challenges which must be overcome: voltage transfer ratio limited to 86.6%, lack of natural bidirectional switches, higher sensibility to input disturbances, protecting the converter is a complex task and the modulation and control techniques are very complex.

In the *MC*, the great majority of scientific studies are being focused on improving the control techniques [3], [5], waveform quality [6], stability analysis [7], modulation [8] and semisoft commutation [9].

Traditionally, *DSPs* (Digital Signal Processor) have been widely used to execute electronic power converters control algorithms [10], while, in this type of application, reconfigurable circuits with high data processing capacity, such as *FPGAs*, are infrautilised. Normally, these have been used to perform secondary tasks to complement the functions of the *DSPs* [10].

This circumstance represents a waste of the potential offered by this technology [11], which can be a solution to respond with greater efficacy and efficiency to the needs of the *MC*.

This paper summarises some of the most important blocks of the matrix converter control in a *SoC* (*System on Chip*) based on an *FPGA*.

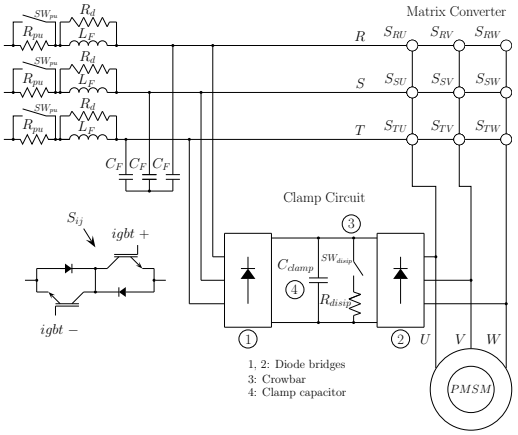


Fig. 1. Matrix Converter, input filter and clamp circuit driving a *PMSM*.

II. MC Control and Protection Requirements

A. Double-Sided Space Vector Modulation (DS SVM)

In the *DS SVM*, the control references are typically the output voltage vector $\overrightarrow{v}_{out}$ and input current vector $\overrightarrow{i}_{in}$ (Fig. 2). In order to calculate the duty cycles of the active and zero vectors it is necessary to transform the reference input phase currents and the reference output phase voltages to the $\alpha\beta$-axes, using the amplitude-invariant Clarke transformation.

The *DS SVM* technique establishes 14 vectors (Fig. 3), which must be activated throughout the modulation period T_{sw}. In this way, the synthesis of the low-frequency component of the converter's reference vectors will be accomplished.

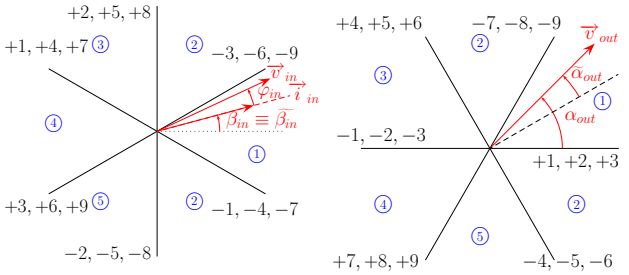
B. Safe Commutation

The transition from one vector to another dictated by the modulation algorithm must guarantee permanent compliance with the *two basic rules* of the *MC*:

1) *Input*: it is not possible to have more than one input phase connected to one output phase as this would cause a short-circuit between the input phases.
2) *Output*: all the switches of one output phase cannot be left opened. Otherwise, if the load is inductive, there will not be a path for the load currents, causing an overvoltage.

The *semisoft* [9] or four-step commutation technique (section IV-A) is the most widely-used commutation

978-1-4244-7855-2/10/$26.00 ©2010 IEEE



(a) Input current. (b) Output voltage.

Fig. 2. Active vector representations in $\alpha\beta$-axes.

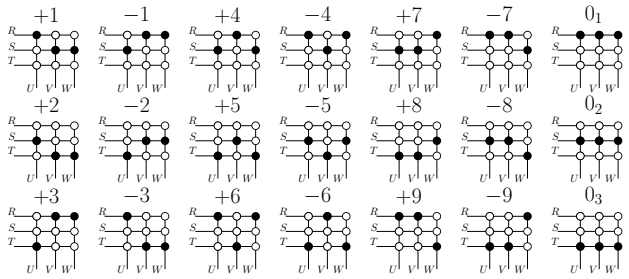


Fig. 3. Stationary active and zero vectors used in the *DS SVM*.

technique in the *MC* due to the robustness and reliability this provides. Another of its attractions resides in the fact that in *50%* of the transitions there are no commutation losses.

C. Protection

MC performs a direct *AC/AC* conversion, without any storage element. So, it is a low ride-through capability converter and it is very difficult to control and protect the *MC* during fault conditions. The reasons for overvoltages may be: input overvoltage due to distortion or perturbation on the grid. Moreover, there may be an overvoltage at the input when the *MC* is started up inadequately; output overvoltage due to an error on the current sign detection or sudden *MC* shut down.

On the other hand, the reasons for the overcurrent could be: *MC* supply faults, *MC* commutation faults; abnormal operation of the motor, overcurrents along *MC* input filter and clamp circuit (Fig. 1) and, finally, circulating current along *MC* switches.

Traditionally, the *MC* incorporates a clamp circuit and a series of power-up resistors (R_d) in the input filter (Fig. 1) to increase the robustness. Unfortunately, these passive elements cannot completely guarantee the reliability of the converter. For this reason, it is necessary to greatly increase the level of protection of the *MC*.

A control architecture based on *System on Chip*, which improves the features of the *MC*, both in computational and protection terms, is presented below.

III. System on Chip: SoC

Today, the density of transistors admissible in electronic devices is such that virtually complete digital systems can be integrated in a single integrated circuit. These systems are the so-called *System on Chips* or *SoCs* [12]. The *System on Chip* is a complete electronic system that integrates several complex blocks (cores) in a single device. These user-designed blocks may be both *hardware* blocks and *software* blocks.

The *SoCs* based on an *FPGAs* afford numerous advantages, such as: efficient use of resources, significant drop in consumption, overall reduction in the silicon surface, reduction in the cost of the product, possibility of performing both sequential executions and parallel executions, increase in the execution speeds, high data flow capacity and, finally, partial dynamic re-configuration [13].

Most *FPGA*-type devices share the same basic architecture. This consists, inter alia, of each of the following elements: *Configurable Logic Cells (CLB)*, *input and output blocks*, *interconnection matrix*, *ram memory blocks* and *clock synchronisation and distribution blocks*.

In addition to the resources of a generic *FPGA*, the *FPGA* used in this paper (*XC4VFX12* of the *Virtex-4*^{TM} family by *Xilinx*) [14] incorporates the following blocks:

1) Signal Processor (*SP*): the XtremeDSP^{TM} slice provides a high performance, low power and versatile arithmetic unit. It can implement a multiply accumulate unit and can be cascaded with other similar blocks to implement larger and critical design structures.
2) *I/O Bandwidth*: provides high-speed serial transceivers capable of *10.3125 Gb/s*.
3) Embedded Processor: *Virtex-4*^{TM} *FPGAs* offer a built-in, enhanced *PowerPC*^{TM} *405* core that delivers *680 MIPS* (Millions of Instructions Per Second) performance.

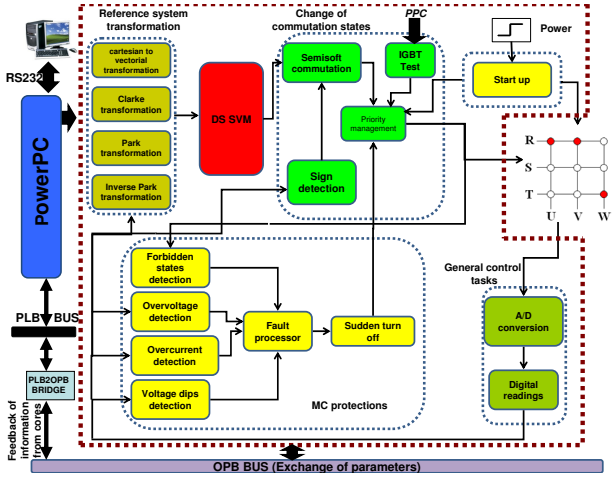
Bearing in mind that the *SoC* is based on an *FPGA*, the system designed makes up a *System on a Programmable Chip* (*SoPC*). The different blocks of the *SoPC* are interconnected by means of specific, standarised buses inside the *FPGA*. Two main buses are used to connect the on-chip peripherals: the Processor Local Bus (*PLB*) and the On chip Peripheral Bus (*OPB*).

The most outstanding functions of the cores developed in this work are the commutation management of the 18 semiconductors and the protection tasks of the *MC*. Fig. 4(a) shows the architecture used to develop the *MC* control application (Fig. 4(b)). In addition, the interaction between the different implemented *hardware* blocks or cores is showed. The implementation of these cores are described below.

IV. Commutation Cores

A. Semisoft commutation core

The general diagram of the cores related to the change of the switches commutation state are presented in Fig. 5. These blocks perform the transition of one vector of the *MC* to another to ensure compliance with the *two basic rules* of the converter (section II-B). This transition is carried out in four steps (Fig. 5 block 3) based on the sign of the output currents of the *MC* [9].



(a) Control platform block diagram.

(b) Prototype of the *MC* 7 kW which integrates the *Virtex-4*^{TM} with the *FPGA*.

Fig. 4. Control cores of the *MC* implemented in the *FPGA XC4VFX12*.

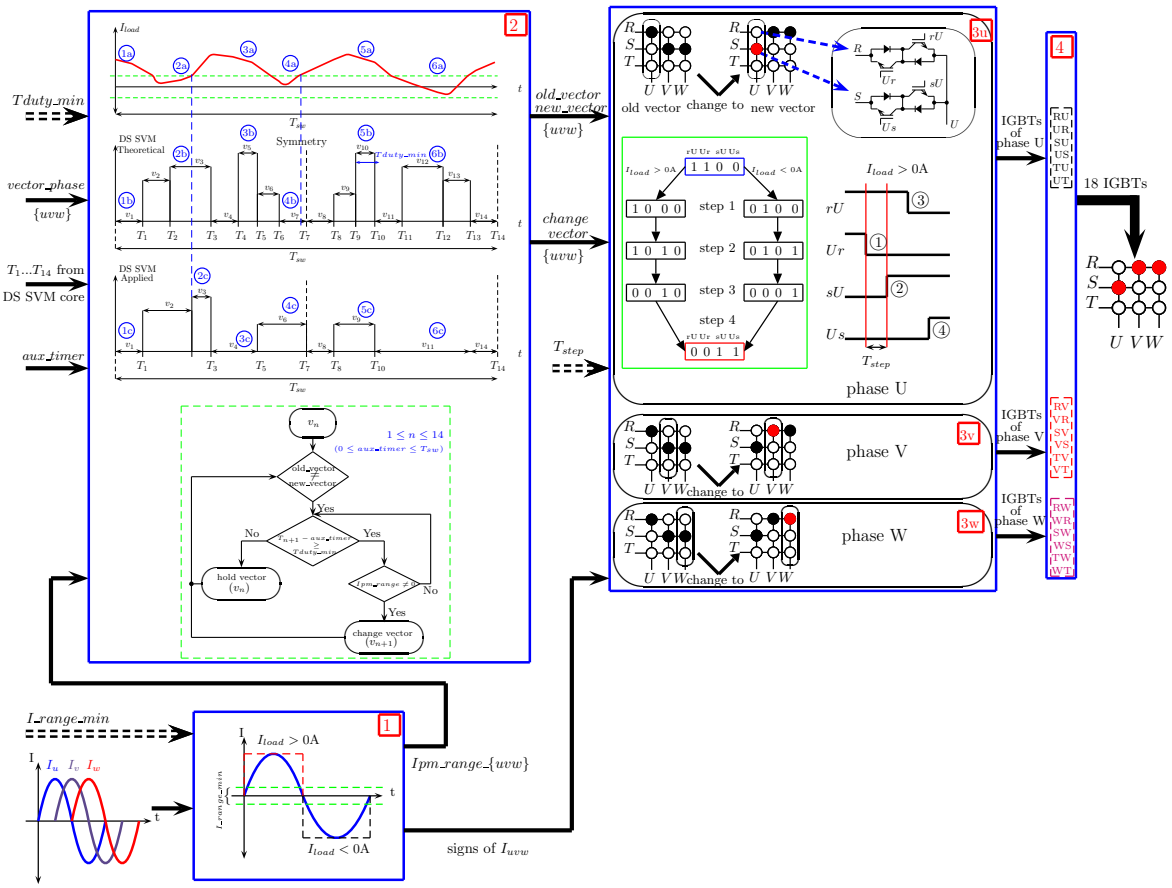


Fig. 5. General diagram of the implementation of the "semisoft commutation".

On the one hand, it must be remembered that the modulation *DS SVM* dictates series of sequences that cause there to be only one phase change between the vectors applied. However, in order to be able to perform a more general transition, the implemented four-step commutation core allows the change from one state to another to be made simultaneously for each one of the output phases U, V and W of the *MC* (i.e., from vector $+1$ to -1, Fig. 5: 3u, 3v and 3w blocks).

On the other hand, this core allows the transitions of vectors to be performed provided that the following two conditions are met:

a) The current level of the corresponding phase must exceed the value of the parameter I_{range_min} (Fig.

5: blocks 1 and 2). In this way, performing an incorrect commutation sequence (Fig. 5 block 3) at low current levels (due, for example, to the offset error of the current sensors) is avoided.

b) A specific vector is applied if its activation time (t_i) exceeds T_{duty_min} (Fig. 5: block 2). This last parameter is determined by the number of steps in the commutation sequence (in this case, 4) and the response time of optocouplers and *IGBTs* of the *MC*.

Should the aforementioned conditions not be met, the core varies the times of the corresponding work cycles, prolonging the duration of the current vector and readjusting the next (Fig. 5, see sequence ②a → ②c), ceasing to apply any vector that fails to comply with the condition (Fig. 5, see ③a → ③c and ④a → ④c), etc. Fig. 5 shows all the possible combinations via the symbols ①a to ⑥c. Fig. 6 shows an experimental result of a four-step sequence in which it has been possible to reach a step (T_{step}) of *160 ns*.

B. Current sign detection core

This core (Fig. 5: block 1) determines, on the one hand, whether the sign of the current of each one of the output phases is positive or negative, and therefore allows the establishment of a specific sequence of four steps or other (Fig. 5 block 3) in the *semisoft* commutation core. On the other hand, this block determines whether the corresponding current exceeds I_{range_min} for each output phase. This information is used by the *semisoft* commutation core.

C. IGBT activation priorities management core

Under normal conditions, the *IGBTs* of the *MC* are governed by the *DS SVM* modulator unit and four-step commutation block. However, there are other blocks, besides the aforementioned one, capable of governing the *18* gates of the *MC*. These are as follows: power-up cycle of the *MC* and sudden shut-down of the *MC*. Bearing in mind that there are some blocks that can govern *IGBTs*, a core (Fig. 4(a)) has been designed to manage the priorities of each one of these modules, so that there is a single core sending the power-up and shut-down orders to the *IGBTs* at all time.

The table I shows the resources required as well as the execution times of the cores involved in the change of commutation status of the *MC*.

V. Protection Cores

The simultaneous execution capacity of the *FPGA* [14], together with its short response time, determine two of the greatest benefits for the control of the *MC* provided by this programmable logic device. Taking these advantages into consideration, in this paper a series of cores has been designed in order to increase the robustness of the *MC*. In this context, it can be said that these cores offer a certain number of advantages over the algorithms implemented in a *DSP*:

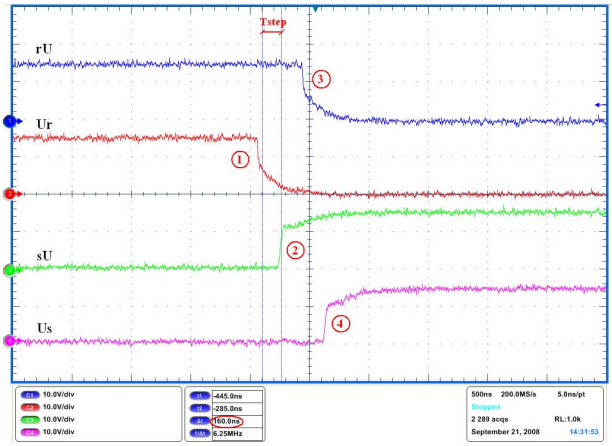


Fig. 6. Four-step commutation sequence (T_{step} of *160 ns*)

- ❒ They are independent protections, each one is executed separately and simultaneously with the others. As a result, the response time of each core does not delay the other protection blocks.
- ❒ The reaction time is drastically shorter than any algorithm based on the execution of *software* instructions.

Each of the cores performed in the *FPGA* for the protection of the *MC* we have built (Fig. 4(b)) is detailed below (the table I shows the number of *hardware* resources).

A. Start-up core

The aim of this core is to ensure the correct power-up of the *MC*, so as to mitigate the overcurrents and overvoltages that might arise during this period. To do this, this core (Fig. 4(a)) detects that the *MC* has started and immediately afterwards performs the following actions simultaneously:

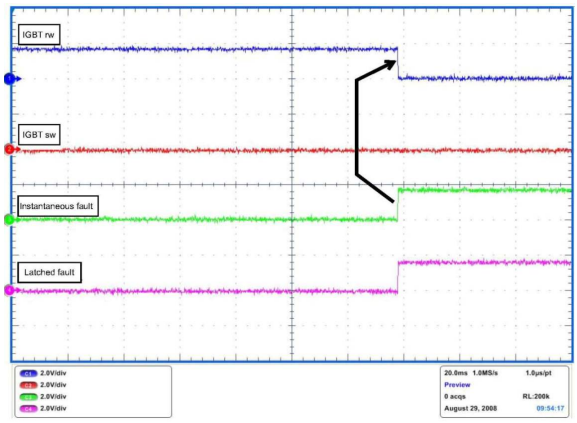
- ❒ Deactivates the *18 IGBTs* of the *MC*.
- ❒ Disables the supply of the optocouplers of the *IGBTs* (Fig. 7(c)).
- ❒ Activates the power-up resistors (R_{pu}) of the filter (Fig. 1) and, once the clamp circuit is loaded, short-circuits these. This sequence reduces the voltage and current spikes generated in the *MC* during the converter power-up stage [15].

The actions carried out by this core do not involve a high computational load. However, the reason behind the implementation of this functionality in a *hardware* circuit is that this must respond extremely quickly (a start-up algorithm executed via *software* does not ensure a correct initialisation, as the latter responds with a certain latency) during the power-up of the *MC* [15].

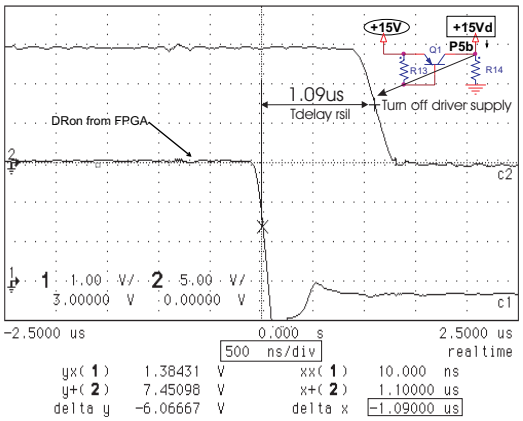
B. Sudden Shut-down core

The function of this *hardware* block is to open, instantaneously, all the *IGBTs* of the *MC*, whenever a fault has been detected in the converter (Fig. 7).

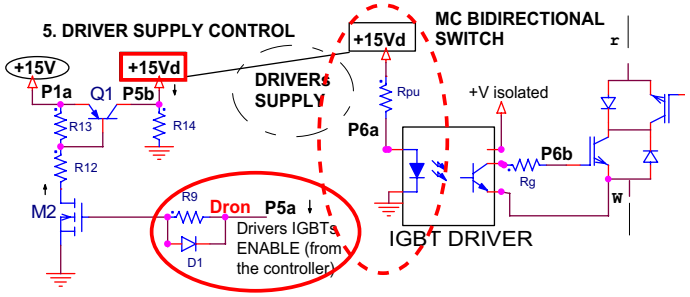
In order to protect the semiconductors, the gates of the *IGBTs* are disactivated (Fig. 7(a)) and the supply of the



(a) Shut-down of the *IGBTs*.



(b) Disabling ("DRon") from the core and shut-down of the optocouplers in 1.09 μs.



(c) Control circuit of the power supply of the optocouplers of the *IGBTs*.

Fig. 7. Protection of the *IGBTs* from the *Sudden Shut-down* core.

optocouplers of the 18 semiconductors of the *MC* (Fig. 7(c) and 7(b)) are disabled, constituting, as in the case of power-up, a redundant system.

Attention should be drawn to the fact that the time the *FPGA* needs to process the operations associated with this core is only 10 *ns* (table I).

C. Forbidden Status Detection core

The aim of this core is to monitor whether the different blocks that have the capacity to govern the *IGBTs* of the *MC* (*semisoft* commutation, *MC* power-up, etc.), Fig. 4(a), issue a permitted combination of signals. These combinations include the *18* active vectors, the *3* zero vectors (Fig. 3) and the complete opening of the *MC*. In this way, the activation of those *IGBTs* that do not comply with the "*two basic rules*" of the *MC* is inhibited.

Should a forbidden status be detected, the corresponding fault signal is activated (Fig. 8) and the "Sudden Shut-down" core is activated (Fig. 7). In this way, the core constitutes a block that prevents the existence of overvoltages and overcurrents in the converter. It should be noted that the time the *FPGA* requires to process the operations associated with this core is 70 *ns* (table I).

D. Fault Detection cores

Three cores have been implemented with the aim of:

- ❒ Overvoltage detection: this *hardware* monitors whether the three-phase voltage of the grid exceeds certain thresholds considered to be dangerous. The sampled voltages and the threshold value are compared to determine dangerous states by means of "if" and "else" sentences. Fig. 10(a) shows an experimental result obtained in the implemented *MC*.
- ❒ Overcurrent detection: this core carries out a function equivalent to the overvoltage detection. However, the variables to be monitored in this case being the output currents (I_U, I_V and I_W) of the *MC*.
- ❒ Dip detection: this module (Fig. 9) monitors whether there are sharp voltage changes at the *MC* input. Fig. 10(b) shows the detection of different disturbances.

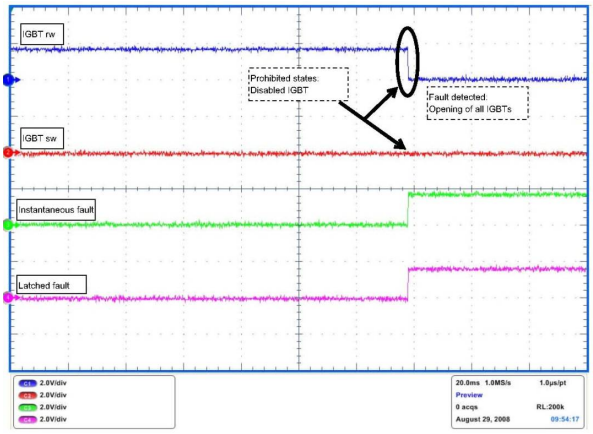


Fig. 8. Detection of forbidden statuses.

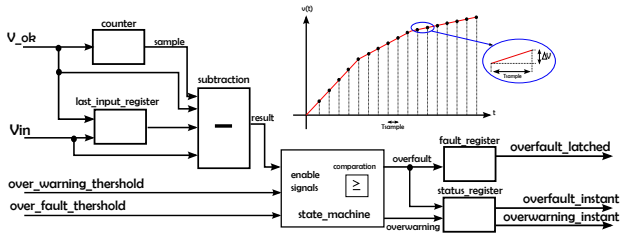
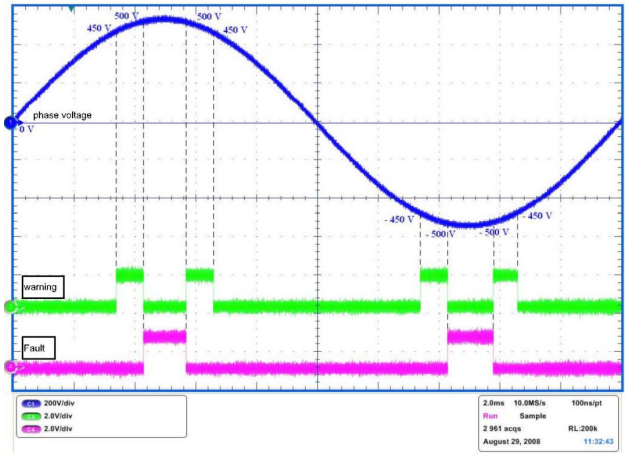
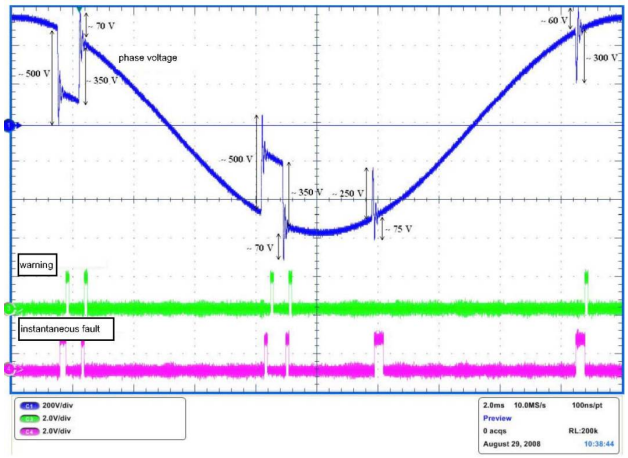


Fig. 9. Internal architecture of dips detection core.



(a) Overvoltage detection.



(b) Dip detection.

Fig. 10. Voltage disturbances at the *MC* input detected by the *FPGA*.

Should these limits be exceeded, these cores activate the corresponding warning or fault signal. The latter is processed by the "Sudden shut-down" core, guaranteeing the safety of the *IGBTs*.

The processing time of this fault detection hardware fluctuates between 30 *ns* and 40 *ns* of the "Dip Detection" core (table I). This demonstrates the high response capacity of the protection system implemented, which eliminates latency problems (critical in certain cases) present in *software*-implemented protection algorithms (via instruction in a μP or DSP).

Table I
RESOURCES AND EXECUTION TIMES OF THE CHANGE OF COMMUTATION STATUS AND *MC* PROTECTION CORES.

Core	FFs	LUTs	SP	Execution time (ns)
Semisoft Commutation	181	371	0	160
Current Sign Detection	0	24	0	10
IGBTs Priorities Management	0	55	0	20
Subtotal	181	450	0	
MC start-up	31	65	0	temporised
Sudden shut-down	2	1	0	10
Forbidden Statuses Detection	2	63	0	70
Overvoltage Detection	14	85	0	30
Overcurrent Detection	14	85	0	30
Dip Detection	49	215	0	40
Subtotal	112	514	0	

VI. CONCLUSIONS

The *MC* presents a number of important advantages. However, this device is characterised for the complexity of its control and the need for protections.

The element that traditionally controls the *MC* is a *DSP*, in association with an *FPGA* to perform the safe commutation sequence. This involves a waste of resources.

In this paper, a series of *hardware* modules in an *FPGA* have been used and the aim of which is to perform: the safe commutation of the converter, the protection of the *MC*, etc.

The designed *hardware* modules operate without latencies, independently and simultaneously, which, together with the computational capacity of the *FPGA*, increases the level of robustness and the dynamical characteristic of the *MC*.

In order to implement the set of cores described in this paper, in total, the following have been required: *293 FFs* and *964 LUTs*, while the processing times are reduced to the order of nanoseconds, which ensures a rapid response of the control and a more efficient protection of the converter. The different developed cores have been validated experimentally in a *MC* of 7 kW.

ACKNOWLEDGMENT

The present paper has been financed by "Ministerio de Educación y Ciencia" and "FEDER" within the research project ENE 2004-07881-C03-01/ALT and by the University of the Basque Country.

REFERENCES

[1] P. Wheeler, J. Clare, L. Empringham, and M. Bland, "Matrix converters: the technology and potential for exploitation," in *The Drives and Controls Power Electronics Conference*, vol. 5, 2001.
[2] K. Shinohara and K. Yamamoto, "Technical trends of direct AC/AC converters," *IEEJ Transactions on Industry Applications*, vol. 126, no. 9, pp. 1161 – 1170, 2006.
[3] A. Rebsdorf and L. Helle, "Variable speed wind turbine having a matrix converter," United States patent 6856038, February 2005.
[4] L. Neft and C. D. Shauder, "Theory and design of a 30 HP matrix converter," *IEEE Transactions on Industry Applications*, vol. 28, no. 3, pp. 546 – 551, May / June 1992.

[5] S. Kwak, "Indirect matrix converter drives for unity displacement factor and minimum switching losses," *Electric Power Systems Research*, vol. 77, no. 5-6, pp. 447 – 454, April 2007.

[6] P. Wheeler, J. Clare, M. Apap, and K. Bradley, "Harmonic loss due to operation of induction machines from matrix converters," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 2, pp. 809 – 816, February 2008.

[7] D. Casadei, J. Clare, L. Empringham, G. Serra, A. Tani, A. Trentin, P. Wheeler, and L. Zarri, "Large-signal model for the stability analysis of matrix converters," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 2, pp. 939 – 950, April 2007.

[8] D. Casadei, G. Serra, A. Tani, and Z. Luca, "Matrix converter modulation strategies: A new general approach based on space-vector representation of the switch state," *IEEE Transactions on Industrial Electronics*, vol. 49, no. 2, pp. 370 – 381, April 2002.

[9] N. Burany, "Safe control of 4-quadrant switches," in *Proc. of IEEE Industry Applications Society (IAS)*, vol. 2, October 1989, pp. 1190 – 1194.

[10] F. Luo and Z. Pan, "Sub-envelope modulation method to reduce total harmonic distortion of AC/AC matrix converters," *IEE Proceedings-Electric Power Applications*, vol. 153, no. 6, pp. 856 – 863, November 2006.

[11] R. Joost and R. Salomon, "Advantages of FPGA-based multiprocessor systems in industrial applications," in *Proc. of Industrial Electronics Society Conference (IECON)*, November 2005, p. 6 pp.

[12] Z. Zalewski, "Winning with SoC designs in FPGAs," *Embedded System Engineering*, vol. 14, no. 8, 2006.

[13] A. Astarloa, A. Zuloaga, U. Bidarte, J. L. Martín, J. Jiménez, and J. Lázaro, "Tornado: A self-reconfiguration control system for core-based multiprocessor CSoPCs," *Journal of Systems Architecture*, vol. 53, no. 9, pp. 629 – 643, September 2007.

[14] Xilinx, "Virtex-4 multi-platform FPGA," http://www.xilinx.com/products/silicon_solutions/fpgas/virtex/virtex4/index.htm.

[15] J. Andreu, J. M. de Diego, I. M. de Alegría, I. Kortabarria, J. L. Martín, and S. Ceballos, "New protection circuit for high speed switching and start-up of a practical matrix converter," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 8, pp. 3100 – 3114, 2008.