FLEXIBLE FPGA INTERFACE FOR
THREE-PHASE POWER MODULES

R. GARCÍA-RETEGUI †, S. A. GONZÁLEZ, and M. FUNES†∗

Dto. de Electrónica, Universidad Nacional de Mar del Plata, Argentina.
† CONICET, Argentina.
rugarcia@fi.mdp.edu.ar

Abstract— This article proposes the development of a flexible interface for the control of Power Inverters. A FPGA-designed system enables an implementation adaptable to any Power Module, with no need to modify the existing hardware. The interface identifies and indicates the type of faults encountered in these modules, generates switching signals on the basis of dead-time and sends a protection signal that inhibits its operation.

Keywords— FPGA, Intelligent Power Modules, Power Semiconductors, VSI.

I. INTRODUCTION

Three-phase inverters synthesize any random waveform from a direct current (DC) source (Kazmierkowski and Malesani, 1998). Their field of application comprises active filters, uninterruptible power sources, power factor correction and motor drivers, among others. (Duan et al., 1999; Ko et al., 2006). Fig. 1 shows schematically the basic inverter and the control system.

The bridge configuration of Fig. 1 has two power switches per leg. Each leg corresponds to a different phase whose output is the common connection point.

The controller defines the switching actions \((R_{out}, R_{out}, S_{out}, T_{out}, T_{out})\) by sensing the inverter output current \((I_{filR}, I_{filS}, I_{filT})\), and the reference current \((I_{refR}, I_{refS}, I_{refT})\). These actions are conditioned to achieve a safe switching. In order to safeguard an optimum operation of the inverter, some protections should be taken into account:

• A time delay or dead-time (DT) addition between same-leg switches turning-off and -on, to avoid short circuits in the DC bus (Leggate and Kerkman, 1997; Trivedi and Shenai, 1998; Lai and Shyu, 2004; Summer and Betz, 2004).

• A current control in the switches to avoid that maximum ratings specified by the manufacturer are not exceeded.

• Temperature of the switches monitoring to avoid reaching maximum devices junction temperature.

• Under voltage lockout for all power supply levels. This prevents the switches activation in the lineal area by voltage supply drops in the control signal.

The power semiconductor technology has allowed manufacturers to integrate semiconductor devices, control electronics and some protection systems in a single chip (Tanaka et al., 2003). These devices, known as Power Modules (PM), provide similar specifications. PM characteristics are available in (Powerex, 1994; International Rectifier, 2005). However, the lack of standardized manufacturing renders them noninterchangeable (Alaith and Grant, 2000). Due to this, all the external electronics mandatory to condition trigger signals, fault type determination and controller operation inhibition are
specific for each PM. Therefore, it is crucial to have an interface (between the controller and the power module) capable of being adapted to any PM.

In this work, a flexible interface that can work across many PM manufacturers is developed. Also, the interface must be flexible on the controller side. In this sense, the interface must be capable of operating with microcontrollers, general purpose Digital Signal Processors (DSPs), analog controllers and other systems. The use of a field programmable logic array (FPGA) for the design of the interface is highly suitable. Moreover, the same FPGA can also be used as an intelligent power controller.

II. PROPOSED SYSTEM

The proposed system consists of the development of a configurable interface. The use of FPGA reprogrammable technology allows an implementation that can be adapted to any given PM, with no need to modify the existing hardware. A reliable and fast execution is achieved, when compared to that carried out in software with microprocessors or DSP.

The interface identifies and indicates the type of fault found in a PM, generates the switching signal with dead-time and sends out a protection signal that inhibits its operation. DT configurability constitutes another feature that confers further flexibility to the system. Fig. 2 provides a scheme of the proposed system.

The interface development is divided into the following three blocks: dead-time generation, determination and indication of fault types, and init control. Fig. 3 depicts the block diagram of the proposed interface. In the following sections, a description of each of the blocks is provided.

A. Dead-time Generation (DTG)

A dead-time addition between switches turning-on and -off in a leg avoids short circuits in the DC bus from switches simultaneous conduction. Simultaneous conductivity results from the switches being non-ideal, since they involve a minimum settling time when turning-on and off. Fig. 4 details the time diagram of the control signals, and the current waveforms in the switches. The way in which the current flows from one switch to another after a given time can thus be observed.

This work proposes the implementation of the switch triggering sequence via a state machine. In doing so, all possible transitions are considered, ensuring no prohibited switching occurs. Unlike the software implemented in a controller, this method is more reliable and does not require further modifications to be adapted to an inverter. Besides,
the state machine contemplates the inhibition of switching pulses lasting less than dead-time (short pulses). These pulses must not generate switchings between power switches, since their turning-off and -on is not guaranteed.

B. Detection and Indication of Fault Type

Since both the nature and number of reported faults by PMs have not been standardized by manufacturers, the electronic circuits developed for a particular module cannot be employed by others. For this reason, the suggested system involves the development of a module adaptable to different PMs, with sufficient inputs to receive the different fault signals on an individual basis.

The module design was divided into two blocks. In the first block, called decoder, the type of fault is determined. In the second, and based upon the detected faults, an external indication is completed and, if appropriate, the controller is inhibited. Fig. 5 shows a block diagram of this module. It is important to note that if the PM were changed, there would be no need to change but the relevant VHDL code of the fault decoder.

C. Interface State Control

This stage (init control in Fig. 3) defines the system state relying on an external signal. Possible states are: disable-reset and start.

In the first state, the system is disabled; i.e., any trigger action generated in the inputs $R_{in}$, $S_{in}$ and $T_{in}$ has no effect on the power switches. Control outputs as well as fault indications remain off.

In the second state, the system is activated; therefore, any change in the $R_{in}$, $S_{in}$ or $T_{in}$ inputs triggers the switches.
Table 1: Time-coded fault signal description

<table>
<thead>
<tr>
<th>Type of fault</th>
<th>Time of pulse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short circuit</td>
<td>2 ms</td>
</tr>
<tr>
<td>Over current</td>
<td>2 ms</td>
</tr>
<tr>
<td>Over temperature</td>
<td>&gt; 100 ms</td>
</tr>
<tr>
<td>Under voltage</td>
<td>&gt; 2 ms</td>
</tr>
</tbody>
</table>

The type of fault is determined by the pulse measurement of the FAULT line. Fig. 5 shows the pulse measurement in order to decode the type of fault and indicate it accordingly.

IV. EXPERIMENTAL RESULTS

The developed system was tested using a 1 hp-three phase inverter. The inverter’s power module is PM25RSB120, commanded by an interface composed by a FPGA Xilinx XC2s50e-6TQ144. Control signals \(R, S, T\) are provided by a DSP (Analog Devices ADCM 401 BST)

The developed interface has optocoupler outputs to isolate control signals from the power stage. Dead-time is determined via a DIP SWITCH 8, with a range of 0 up to 25.6 \(\mu\)s in 100 ns steps. The board is provided with fault indicators and an operation mode selector.

The FPGA design was assessed in terms of logical and temporization resource usage. The use of the area obtained was of 12%. This reduced area level allows the incorporation of new features to the system. System temporization is restricted in terms of fulfillment of switches dead-time. By generating restriction during the implementation, it could be seen that the maximum time delay between switches command and driving was below 7 ns, which means 7% of DT resolution step.

Tests were carried out in two stages. The first one assessed dead-time generation with the power section disabled. A 2, 4 \(\mu\)s-DT was programmed and a PWM signal was examined. Fig. 7 shows PWM signals of one leg and the obtained dead-time.

In addition to DT measuring, dead-time generator operation was tested when the pulses lasting less than DT appeared. Fig. 8 shows how a short pulse (2.2 \(\mu\)s) in PWM signal generates just one power switch commutation.

V. CONCLUSIONS

This article introduces the development of an interface to control Power Inverters. This interface not only generates dead-times but also indicates and protects detected faults in the inverter control.

The aim of this work is to provide an interface adaptable to both, an integrated power module and a discrete switch command, with individual fault detection. Therefore, the use of a FPGA is suitable for this application since it allows enough inputs to adapt its design to different alternatives.

The FPGA-system design is flexible as it can be adapted to any PM, with no need to modify the existing hardware. In case PM is changed, only the code VHDL corresponding to the faults decoder should be modified.

The purpose of using the proposed interface to control the inverter device is to exempt the task controller from generating dead-time and faults management. Besides, the implementation with programmable logics renders the system more reliable and fast when compared to that performed with a software.

By means of restrictions generation in the implementation, it was proved that the maximum time delay between switches command and driving was below 7 ns. This renders it an accurate system of higher resolution when it comes to generating dead-time.
REFERENCES


Received: April 14, 2006.
Accepted: September 8, 2006.
Recommended by Special Issue Editors Hilda Larrondo, Gustavo Sutter.