FPGA Based Monitoring and Protection System for Industrial drive Application

Shekhar Bhawal, Banda J Rajendranath, Vamshi Krishna M, Jose Titus, Kamalesh Hatua, Krishna Vasudevan
Dept. of Electrical Engg, IIT Madras, Chennai, Tamilnadu, India.

A Koteswara Rao, Shoubhik Mukherjee, G Balawanth Reddy
S Eswara Rao, Bishnu P. Muni
Bharat Heavy Electricals Ltd., Hyderabad.

Abstract—This paper describes an FPGA based industrial drive monitoring system. The FPGA system monitors different drive parameters and provide RMS based back up protection in an industrial grade Variable Frequency Drive (VFD). The designed FPGA system establishes all the necessary communications with a Human Machine Interface (HMI) for displaying drive parameters and fault status. Necessary hardware results are illustrated in this paper.

Index Terms—FPGA, Monitoring, Protection, communication , VFD, HMI

I. INTRODUCTION

The high speed Digital Signal Processor (DSP) and Field Programmable Gate Array (FPGA) are generally used as a controller for a Variable Frequency Drive (VFD). In high power drives, it is desirable to implement control and monitoring system in two separate controller. A DSP and FPGA based hybrid controller is developed to drive a 20 KW induction motor. The benefits of DSP and FPGA controllers needs to be utilized to its full extent by sharing jobs appropriately between them. In this work, drive control algorithm is implemented in DSP while communication protocols, monitoring and software protection schemes are implemented in FPGA. FPGA is very fast and capable of carrying out parallel computations. In this paper, FPGA based monitoring and protection system are described elaborately.

Fig 1. shows functional block diagram of the drive system. The designed FPGA system monitors different parameters of the drive system and implements software based protection with fault latching and identification. The FPGA system establishes serial communications with a Human Machine Interface (HMI) for displaying parameters and fault status of the drive in real time. The design of the FPGA system is based on NIOS II soft processor core and block diagram logic. The efficient performance of the system depends on the synchronous operation between NIOS II soft processor core and block diagram logic.

Section II describes the monitoring and protection system implemented in a 20 KW induction motor drive system. Section III describes the FPGA system
architecture, synchronization of the NIOS II processor with block diagram logic and flow chart of the implemented algorithm. Section IV presents the hardware results. Section V concludes the paper.

II. CONDITION MONITORING AND PROTECTION

The FPGA system monitors different parameters of the drive and provides software protection. The FPGA system establishes RS-485 serial communication with Red Lion G306 HMI for displaying all parameters and fault status of the drive in real time.

A. Instantaneous Signal Monitoring

The FPGA system provides instantaneous signal monitoring of different parameters like rotor speed, heat sink temperature and DC bus voltage of inverter and displays the same on HMI in real time.

B. RMS Monitoring

The FPGA system calculates true RMS values of phase and line voltages and currents. The true RMS formula is given in the equation (1).

\[
RMS = \sqrt{\frac{\sum_{n=0}^{N-1} [x(n)^2]}{N}} \quad (1)
\]

where \( n \) corresponds to \( n^{th} \) sample number, \( x(n) \) corresponds to \( n^{th} \) sample and \( N \) is the total number of sample over a time period. The FPGA system displays the true RMS values on the HMI in real time.

C. Harmonics Monitoring

The designed FPGA system calculates fundamental and harmonic components (3rd, 5th, 7th and 11th harmonic) and Total Harmonic Distortion (THD) of phase and line voltages and currents. The Discrete Fourier Transform (DFT) algorithm is implemented for harmonic calculations. The DFT algorithm can be represented in the following manner:

\[
A_i = \sqrt{(a_i^2 + b_i^2)} \quad (2)
\]

where \( A_i \) corresponds to \( i^{th} \) harmonic magnitude and \( a_i \) and \( b_i \) are as follows:

\[
a_i = \frac{2}{N} \sum_{n=0}^{N-1} [x(n) \times \cos(\frac{2\pi}{N} \times n \times i)] \quad (3)
\]

\[
b_i = \frac{2}{N} \sum_{n=0}^{N-1} [x(n) \times \sin(\frac{2\pi}{N} \times n \times i)] \quad (4)
\]

where \( n \) corresponds to \( n^{th} \) sample number, \( x(n) \) corresponds to \( n^{th} \) sample, \( N \) is total number of sample and \( i=1,3,5,7 \) and 11. THD is given in the equation (5) after neglecting higher order harmonics.

\[
THD = \sqrt{\frac{(A_{rms}^2 - A_{1rms}^2)}{A_{1rms}^2}} \approx \sqrt{\frac{\sum(A_i^2)}{A_{1rms}^2}} \quad (5)
\]

where \( A_{rms} \) is the \( i^{th} \) harmonic RMS value of the phase or line voltages or current and \( A_{1rms} \) corresponds to fundamental RMS value of the phase or line voltages or current and \( i=3,5,7 \) and 11. The FPGA calculates all the harmonics and THD and displays the same on HMI in real time.

D. RMS based back up protection

The FPGA system implements over voltage and over current protection that serves as a backup protection for the drive system. The RMS values of the currents and voltages are used for the protection system. The analog protection circuit based on instantaneous values are implemented separately. Analog protection system serves as primary protection unit of the drive system. The software protection system is shown in Fig 2. When one or more RMS values exceed beyond the specified limit, corresponding output of the comparator goes high, corresponding RS filp-flop latches the comparator output. The output of the corresponding RS flip-flop remains high even if the corresponding fault is cleared. All individual R-S latches are ORed to obtain the final trip signal which communicates with the slave controller (DSP). The outputs of the RS flip-flops can be cleared by external reset signal. The protection algorithm is implemented in the block diagram section, which provides very fast response. The FPGA system identifies the faults and communicates the same to HMI for displaying fault status.
E. Implementation of RS-485 Communication

The FPGA system displays all the parameters and fault status of the drive on a Red Lion G306 HMI in real time. The FPGA system establishes RS-485 serial communication with the HMI for transferring data in real time. The external level-shifting buffer has been used between the FPGA I/O pins and the external RS-485 connectors for complying with the RS-485 voltage specifications. The implemented RS-485 communication is shown in Fig 3. The 16 bit variable named as signal[15..0] represents the applied signal, the 32 bit variable named as rms[31..0] represents the calculated RMS value and serial_txd represent the transmitted RMS value to the HMI from FPGA. The one of the display pages of HMI is shown in Fig 4.

III. FPGA SYSTEM ARCHITECTURE AND IMPLEMENTATION

A generic DSP-FPGA hybrid board is developed for the drive system. Fig 5. shows the control architecture of the drive system. The hybrid board with HMI is shown in Fig 6. Altera Cyclone IV E FPGA (EP4CE30F23I7) is selected for master controller. DSP (TMS320F28335) is selected for slave controller. FPGA communicates with DSP by parallel communication. FPGA communicates with HMI by RS-485 serial communication. The designed FPGA system consists of the NIOS II soft processor core and block diagram logic. All the FPGA designs are carried out using Quartus II software (version 14.0). Altera provides Qsys system integration tool with Quartus II software (version14.0) which is used in the design.

NIOS II processor is constructed using Qsys design tool. NIOS II processor along with all associated IP cores are added, configured, inter connected and finally equivalent block diagram is generated using Qsys. Fig 7. shows FPGA system architecture. Central Processing Unit (CPU) of NIOS II processor communicates with all IP cores using Avalon Memory-Mapped (Avalon-MM) slave interface. NIOS II standard hardware system consists of CPU, On-chip memory (to store and run the software), JTAG link (for communication between
the host computer and target device), EPCS flash controller (for flashing software and hardware image file to the target device), interval timer (to generate periodic interrupt), System ID peripheral, UART core (establishes serial communication (RS-485) with HMI) and PIO ports (Parallel Input Output ports, provide interface between NIOS and Block diagram logic).

FPGA block diagram diagram section consists of ADC (Analog to Digital Converter), DAC (Digital to Analog Converter), counter (used as a frequency divider), PLL (used as a higher frequency clock).

A. NIOS II Processor Synchronization with Block Diagram Logic

The Block Diagram Logic Section of the FPGA is a parallel processor. On the other hand, NIOS II processor executes instructions serially. Therefore, it is important to synchronize the NIOS II processor with Block Diagram Section to avoid misinterpretation of data.

Fig. 8. Flowchart of the algorithm.
IV. RESULTS AND DISCUSSION

Calculation of true RMS of a sine wave (1000V peak, 50 Hz, 1 – φ) is shown in Fig 9. The variable named as sin[15..0] represents the signal. NIOS processor calculates true RMS and stores it into the variable named as dataout[31..0]. Then FPGA communicates with HMI to display the same in real time. The designed FPGA system also calculates fundamental, 3rd, 5th, 7th and 11th harmonic components and THD. After that, FPGA communicates with HMI to display the same in real time. Fig 10. shows the calculations of true RMS along with all the harmonics and THD of a square wave (1000V peak, 50 Hz, 1 – φ). The calculated RMS, harmonic components and THD values with actual values of the square wave signal is shown in Table I.
TABLE I
A COMPARISON OF CALCULATED VALUE TO TRUE VALUE

<table>
<thead>
<tr>
<th>Electrical Parameters</th>
<th>Actual value</th>
<th>Calculated value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMS</td>
<td>1000V</td>
<td>999V</td>
</tr>
<tr>
<td>Fundamental component</td>
<td>1273.23</td>
<td>1273</td>
</tr>
<tr>
<td>3rd harmonic</td>
<td>424.41</td>
<td>424</td>
</tr>
<tr>
<td>5th harmonic</td>
<td>254.64</td>
<td>255</td>
</tr>
<tr>
<td>7th harmonic</td>
<td>181.89</td>
<td>182</td>
</tr>
<tr>
<td>11th harmonic</td>
<td>115.75</td>
<td>117</td>
</tr>
<tr>
<td>THD</td>
<td>42.40</td>
<td>42</td>
</tr>
</tbody>
</table>

V. CONCLUSION

In this paper an FPGA based industrial drive monitoring and protection system are described. Overall FPGA design is carried out by concatenating block diagram files and programming the NIOS II processor. Parameters like RMS, all harmonic components, THD, DC bus voltage of the inverter, speed and supply frequency are monitored. The serial communication with HMI is established for real time display. The RMS value based software protection is also implemented as a back up protection for the drive system.

Fig. 11. The Experimental Setup.

REFERENCES


