Architecture for Electromechanical Model with Control Devices

Vladimir Karailiev[1] and Valentina Rankovska[2]
[1]Faculty of Electrical Engineering at Technical University of Sofia
8 Kl. Ohridski Blvd, Sofia 1000
Bulgaria
vkarailiev@gmail.com

[2]Faculty of Electrical Engineering and Electronics at Technical University of Gabrovo
4H. Dimitar str., Gabrovo 5300, Bulgaria
rankovska@tugab.bg

ABSTRACT: *We have developed a new architecture for an electromechanical model. We have explained the primary and auxiliary functions of the control device. We have given the appropriate components of efficient control devices. We have used the programmable logic devices to develop the architecture. Besides, we have presented the hardware and software part of the system in detail.*

Keywords: Electromechanical System (EMS), Electrical drives, Field-Programmable Gate Arrays (FPGA), Design, Simulation

Received: 19 September 2022, Revised 20 November 2022, Accepted 3 December 2022

DOI: 10.6025/jet/2023/14/1/1-8

1. Introduction

More than 90% of the electrical drives in the world include induction motors. But the necessity of a fluent velocity adjustment in a wide range and the requirements for a high starting torque and high energy indices have been an introduce holdback of the induction motors in the automotive and electrical transport. Only the recent twenty years thanks to the development of the power electronics and the microprocessor circuits the inductive electrical drives have become competitive to the asynchronous ones. A variant of a new approach for designing high efficiency control devices in the electrical drive systems is presented in the paper.

2. General Block Diagram of Electromechanical System

The examined Electromechanical Systems (EMS) are various kinds of electric drive systems, which convert the electrical energy into mechanical or vice versa. Generally these systems can be depicted by a simple block diagram, shown in Figure 1, where: IN is an Interface node; CD – Control Device; PC – Power Converter; M – Motor; MN – Mechanical Node; ON – Operating Node; S1, S2, ... , Sn – sensors for the adjustable coordinates; PS – Power Supply; PG – Power Group of the EMS; ICG – Information Control Group.

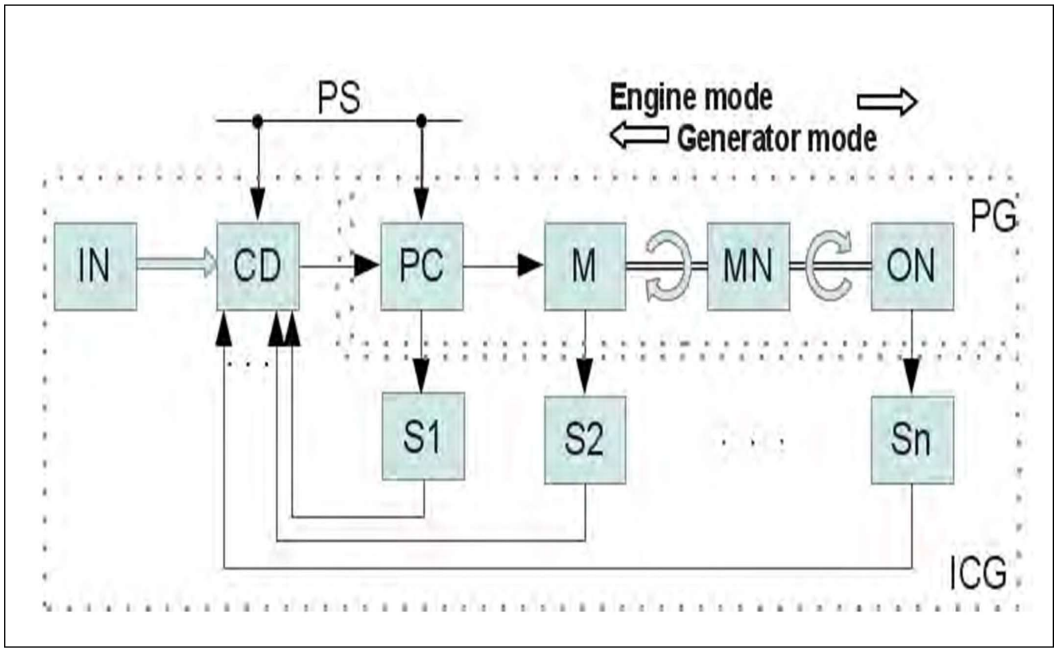

Figure 1. Block diagram of electromechanical system

3. Basic and Auxiliary Functions Implemented by the EMS Control and a Choice Of Suitable Components for High Efficiency Control Devices

Functions, performed by the EMS control with a closed operational loop
A control of the considered EMS means a process organization of the power conversion, supplying the necessary operation modes of the mechanical nodes. The control is automated, where the corresponding control signals could come from higher hierarchical level via wire or wireless interface (IN).

The main functions executed by the EMS control are the following:

• Starting, stopping and reversing the rotation direction;

• Keeping the defined coordinates values;

• Defined operation mode implementation by a control program;

• Watching arbitrary changing signals coming from the sensors.

Auxiliary functions, performed at the monitoring, the control and the protection of EMS Except the main control functions some auxiliary functions can be performed such as:

• Protection of the electric motors, converters and mechanical nodes against various overloads;

• Locks preventing from failures and abnormal modes;

• Diagnostics and signaling for the system blocks states and for the progress of the technological process, etc.

Selecting suitable components for the high efficiency control devices design

The last tendencies in the high efficiency control systems in the electrical drive systems, based on control methods like:

Magnetic field orientation; Vector control; Neural networks;

Searching based on fuzzy logic, use microcontrollers and DSP. However they have got many disadvantages, which do not allow them to achieve best results.

Drawbacks of the traditional and DSP microcontrollers [1], [7]:
• The lack of options for parallel data processing;

• Difficult interrupt request handling in complex system working in real time;

• Limited resources which make difficult the complex systems implementations;

• Lack of differential signaling inputs and outputs leading to low noise resistance;

• More difficult implementation of bit operations made entirely by software;

Advantages of the programmable logic devices
The FPGA and CPLD production and applying is one of the fastest growing areas in electronics, because of the number of their advantages, like: ([3], [6], [7])

• Possibilities for synthesizing hardware by software means;

• Possibilities for integrating various functions and several microcontrollers in one chip;

• Implementing complex algorithms for parallel processing;

• Possibilities for using various embedded features;

• Higher information power at data processing;

• Software tools with many features for synthesis, simulation and verification of the designs;

• Easy implementation of bit operations, which leads to increased effectiveness of the processed information.

4. A New Approach in Designing High Efficiency Control Devices in the Electrical Drive Systems Based on FPGA

The appearance of the programmable logic devices (PLD) is a new era in the digital devices and systems design flow development.

The configurable and reconfigurable logic devices are highly integrated flexible universal circuits, including power logic, memory and allowing In-system Programming (ISP).

Main features, special purpose blocks and PLD selection criteria
FPGA main features [8]
The system operation of FPGAs is defined by their features; their parameters are like of the other integrated circuits. An idea of the complexity of the FPGAs is derived from the number of the equivalent logic elements (LE), usually called System Gates (SG), and together with them the number of the Logic Cells (LC) is usually given in the technical data.

The ability of the FPGAs to communicate with other devices can be appreciated by the number of the parallel inputs and outputs and the highest possible speed. Other features are the number of embedded interface modules, the type of embedded memories and their volume. The memory organization is not fixed and can be configured during programming. The same is referred to the embedded FIFO memory blocks in some FPGAs often used as buffers. The main features of the DSPs, which are optional blocks too, are the length of the multiplied numbers and the maximum operating frequency.

Special purpose peripheral blocks in FPGAs

• Phased-Lock Loop (PLL) blocks

Cyclone FPGAs include PLL blocks and a global clock network. The PLL allow:

• Clock multiplication and division;

• Phase shift;

• Programmable duty cycle;

• External clock outputs

All the features allow the clock signals and edge delays control at system level.

• Embedded hardware multipliers

Most of the FPGAs of Xilinx and Altera include hardware multipliers. They could be used for digital signal processing, where many multiplications are implied, such as Finite Impulse Response (FIR) filters, quick Furrier transform, discrete cosine transform, etc.

• Digital Signal Processors (DSP)

Traditionally Digital Signal Processing/ Processors (DSPs) are used to process digital signals. They have a standard architecture, which advantage is that they are flexible and can be used for filtering and modulations. Only the software is changed for that purpose. On the other hand their flexibility limits their efficiency. Initially the DSPs include only one multiplier but the recent ones have to 8 multipliers. Iteration calculations consisting of 2 - 8 multiplications for the corresponding system clocks are usually used to calculate the result. That is why the DSPs are more suitable for signal processing in systems of medium to low productivity.

Programmable logic selection criteria

The FPGAs selection criteria depend on the application of the designed system. They can be as follows:

• An availability of embedded resources according to the application;

• Enough logic capacity, for the monitoring and control algorithms implementation;

• The features and the cost of the hardware and software development tools and configuration memories;

• The cost of the programmable logic devices;

• Availability of development tools, methodological and technical supplement, etc.

Design flow for high efficiency electrical drive control devices

There are two separate design stages which are carried out in parallel in time in order FPGA system to reach the market: design stage and debug and verification stage (Figure 2). The detailed system design flow is presented in [2], [4], [5] and [8].

• System design stage

The main design stages are input, simulation, implementation and programming in FPGA. At this stage the debug also begins using simulation tools.

• Debug and verification stage

The severe problems not seen at the simulation process should be found out at the debug stage.

Integrated development environment for system design

The modern integrated development environments include program modules with wide range of features. They include:

• Tools used to input/ edit, compile, simulate, program the project;

• (Intellectual Property) cores, ready to use, implementing various functions – general purpose processor and DSP, components implementing various interfaces and communication protocols, modulators and demodulators, coders, memories, etc.;

• Software tools allowing creating projects including DSP modules; various tools for functional or timing design simulation;

• Various design and test modules for complex systems at block level;

• Software tools for design and simulation of applications based on 16- or 32-bit software CPU core including peripheral blocks, etc.

Design approaches and stages

There are two design approaches: Bottom-Up and Top- Down [4]. At the Top-Down approach a leading designer creates and optimizes the top level of the project on the whole.

The Bottom-Up approach allows creating the project top level, which includes any number of projects from a lower level as partitions of it. Therefore the designer (or the designers in the team) can design and optimize every partition as a separate project.

The FPGA based devices and systems design consequence is nearly identical for the various producers and consists of the following stages, shown in Figure 2.

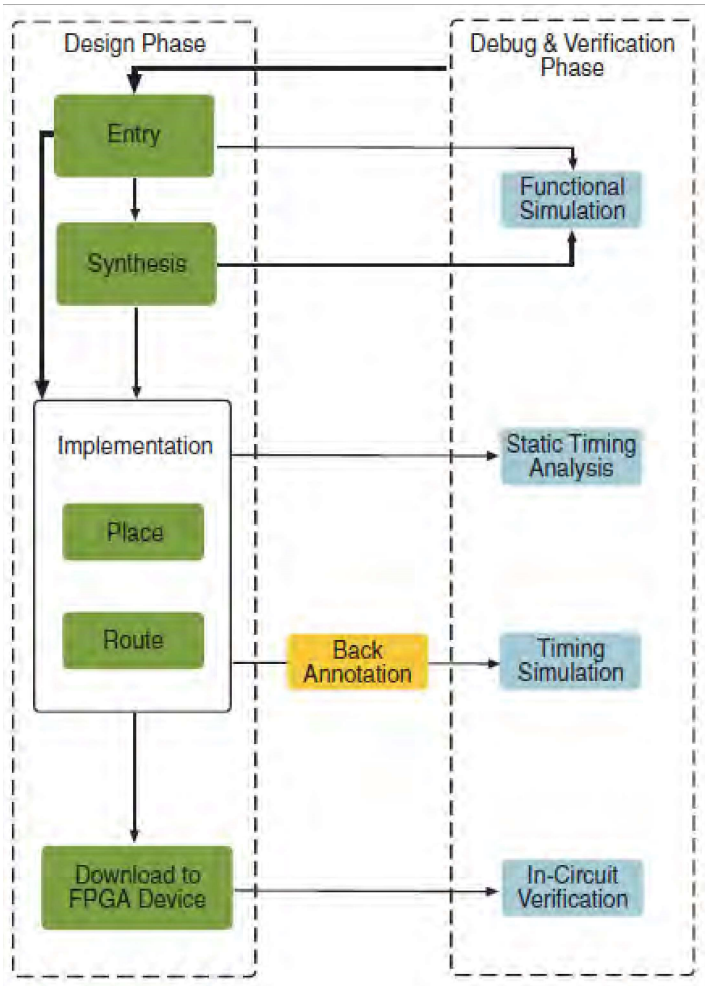

Figure 2. Design flow for FPGA based systems

0) It is possible to input a device or a system design as a *behavior or structure model with* various design automation software tools in the following ways:

- By a program written on some of the following hardware description languages - AHDL, VHDL, Verilog, etc.
- Usage of ready library components;
- Creation of a memory initialization file using a special memory editor;
- As a block diagram using a block editor;
- A combination of the above approaches.

Defining project assignments and integrated environment settings
The preliminary assignments of various requirements to the project and software environments settings allow controlling their features and options in order to increase its effectiveness. Conditions for project optimization can be also defined.

Project compilation
Two main approaches for design compilation exist – flat and incremental compilation. In the flat compilation the compiler uses several modules to manage the project, creating one or more files for programming.

In the incremental compilation it is possible to compile project partitions independently. It reduces the necessary memory and time for the compilation process, allowing recompiling only the modified parts of the design. The incremental compilation is used with large designs and when changes have to be made only in some parts.

The compilation stage consists of several sub-stages: Analysis & Synthesis, Place & Route, Assembling and Timing Analysis.

The design database is created at the Analysis & Synthesis process. A logical synthesis is made with minimizing the logic used and verifying the logical completeness and for syntax errors. An optimization is made considering the volume of the resources used.

Fitting (place and route) the design logic blocks is performed after that according to the available chip capacity.

At the same time the defined logical and timing requirements for the project are checked.

During the assembly the design processing is finished by creating device programming files and data for the consumed power.

The time analysis is a method for analyzing, debugging and verification the correct operation of the project. The all signal path times in the project are measured and the project operation and efficiency is verified.

Design Simulation
The integrated environment simulator is a tool used to test and debug the design logical operation and the internal timings. It allows the verification of the design before its hardware implementation. The simulator can reduce considerably the time for initial design transmutation into a working circuit.

According to the type of the necessary information, functional or timing simulation could be carried out, to test the design logical operation.

a) Simulator settings
A vector waveform file consisting of the input vectors has to be created before simulation. The simulator uses it to simulate the input signals which the programmed circuit would generate at the same conditions. Break points could be also added at which the simulation will stop allowing to the designer to correct the errors.

Other user settings could be also defined, such as simulation period, setting time and delay errors, detecting glitches in the

transients, etc.

b) Options of the Vector Waveform Editor
The Vector Waveform Editor allows inputting simulation vectors and examining the simulation results in a graphical form. It can be used to create Vector Waveform Files (.vwf) or Compressed Vector Waveform Files (.cvwf), which contain the input simulation vectors. The waveform file can be created by defining the input logical levels as a graphical time diagrams.

c) Creating a Waveform file for design simulation
The waveform file is an ASCII file depicting the simulating input and output vectors in a graphical form. It is possible to input and edit the file, to add new nodes, to delete nodes, to change their order, to define the numbering system radix, to rename nodes and to change their type. An existing file also could be used. If the vector waveform file is used, it defines the following:

- Input logic levels, which control the pins and define the internal logic levels of the design.
- The nodes we want to watch at the times when the applied vectors start, stop and their duration.
- The nodes, which have to be grouped in the simulation process.
- The numbering system radix used at the interpretation of the logic levels.
- Bidirectional line simulation. They are represented in the .vwf or .cvwf file by two ports: one input and one output.

d) Design simulation and results report
The simulation process is carried out until its completion, till break point reached, till stopping or cancelling. The eventual errors and warnings could be examined in the Messages window.

Hardware tools for FPGA and CPLD based system design

Altera produces several types of development tools according to the FPGA family an application area of the selected circuit [8]:

- For digital signal processing – variants of DSP Development Kits for various FPGA families;
- High-Speed Development Kits including various high speed interfaces;
- PCI/PCI-X - PCI Development Kits and PCI High-Speed Development Kits – including various type and quantity of memory, interfaces on a standard PCI card.
- Nios II Development Kits, including the software processor Nios II, the necessary software and an access to many Intellectual Property (IP) cores and example projects.
- General purpose design tools. All they use in-circuit serial programming.
- Autonomous Programming Units (APU) (with USB interface) including hardware and software for programming all the FPGA families of Altera.

5. Conclusion

A new design approach for high efficiency control devices in the electrical drive systems is suggested in the paper. The main design and verification stages and hardware and software tools are examined. The approach allows reducing the design time and money for electrical drive systems operating at many times higher frequencies and applying new control methods with higher efficiency and better reliability.

Acknowledgement

The present work is partially supported by the Science Research Fund at the Ministry of Education, Youth and Science under contract No E1301/2013.

References

[1] Bazzi, A. & Krein, P. (2010). A Complete Implementation Procedure for State Estimation in Induction Machines on the eZdsp F2812. University of Illinois at Urbana-Champaign.

[2] Rankovska, V. & Karailiev, H. (2011). (in Bulgarian). *Digital Devices and Systems Design Using Field-Programmable Gate Arrays*, Vol. 1–2, pp. 8–15.

[3] Woods, R. et al. (2008). *FPGA-Based Implementation of Signal Processing Systems*. John Wiley & Sons: New York, USA.

[4] Quartus, I.I. (2012). *Design & Synthesis*, version 12.1 [Handbook], Vol. 1. Altera, Corp.

[5] Cofer, R.C., and and Ben Harding. Rapid System Prototyping with FPGAs: *Accelerating the Design Process* (2006). Elsevier: Oxford.

[6] Kilts, S. (2007). *Advanced FPGA Design: Architecture, Implementation, and Optimization*. John Wiley & Sons: New York, USA.

[7] Implementation of a Speed Field Orientated Control of Three Phase AC Induction Motor Using TMS320F240 (1998). Texas Instruments.

[8] www.altera.com.