Buletinul Stiintific al Universitatii "Politehnica" din Timisoara, ROMANIA
Seria AUTOMATICA si CALCULATOARE
PERIODICA POLITECHNICA, Transactions on AUTOMATIC CONTROL and COMPUTER SCIENCE
Vol.47 (61), 2002, ISSN 1224-600X

Implementation Characteristics of Library Modules for Vector Control System for Tandem Converter Fed AC Drive

József Vásárhelyi[*], Mária Imecs[**], Csaba Szabó[**], Ioan Incze[**], Tihmér Ádám[*]

[*] Department of Automation, University of Miskolc, Egyetemváros, H3515 Miskolc, Hungary
Phone: (36) 46 565-111, Fax: (36) 46 431-822, E-Mail: vajo@mazsola.iit.uni-miskolc.hu, elkadam@gold.uni-miskolc.hu

[**] Department of Electric Drives and Robots, Technical University of Cluj-Napoca, str. Daicoviciu 15, RO3400 Cluj, Romania
Phone: (40) 264 195-699-164, Fax: (40) 264 192-055, E-Mail: {imecs, Csaba.Szabo, ioan.incze}@edr.utcluj.ro

Abstract ***– The paper focuses on the implemented library modules for vector control systems of AC drive. The rapid prototyping and fast implementation of vector control systems become possible with the created module library. There are presented the performances and characteristics of the main modules. The created control structures were compared with other simulation results. The control system structures are implemented in configurable logic cells using Field Programmable Gate Array (FPGA). The analyses of the modules give information of the hardware resources needed to implement all the control system as a parallel computing structure.***

Keywords: ***reconfigurable computing, FPGA, rapid prototyping, vector control, tandem converter.***

I. INTRODUCTION

Most motor control applications concern with vector control for AC drives. Vector control systems for induction motors give the best dynamic behaviour. Analysing these systems some modularity can be observed, which help fast implementation of motor control applications in reconfigurable structures [3], [10], [13], [14].

Reconfigurable hardware was used in vector control in the last years for control system implementations [3], [11], [12], [14]. In vector control systems, the reconfigurability was introduced by Imecs et all in [1]. In this concept, each configuration is considered as a state of a logic state machine. When reconfiguration condition occurs, the system will start reconfiguration process in which it switches the current configuration to the next corresponding one. This type of configuration is the context switching and was developed by Sanders in [6]. While context switching is a reconfiguration technology for Field Programmable Gate Arrays (FPGA), the logic state machine with different control system structure in each state is a reconfiguration method for vector control systems.

In order to make the reconfiguration possible, there the known control structures were deeply analysed. Kelemen and Imecs in [5] presented most of the known control structures for AC drive. As a result of the analyses it was possible to create a module library. Each module of the library was analysed separately from the viewpoint of used hardware, and performances.

The functionality of the modules was tested for the tandem converter. The results of this implementation and analyses will be presented as follows.

II. VECTOR CONTROL FOR TANDEM CONVERTER

The "*tandem*" configuration was proposed as a new solution of the SFC for medium- and high-power AC drives [7], [8]. It is a hybrid SFC, which combines the advantages of two components DC-link converters, which are of different types and different power ranges, and they are working in parallel arrangement. The control of the tandem-converter-fed induction motor can be achieved using conventional vector-control structures. If one of the components SFCs fails, in order to continue the drive its mission, the structure of the motor control system should be change depending on the actual working component SFC. Reconfigurable structure allows adapting the control system - implemented on Field Programmable Gate Arrays (FPGA) - to the actual operating situations.

The larger one of the component SFCs (Fig. 1) contains a conventional Current-Source Inverter (CSI) operating with 120° current wave-forms controlled by Pulse-Amplitude Modulation (PAM) and it converts the most part of the motor feeding energy. The smaller component SFC involves a well-known Voltage-Source Inverter (VSI) controlled by Pulse-Width Modulation (PWM) and it supplies the reactive power required to improve the quality of the motor currents in order to compensate them in sine-wave form. Consequently, the current i_s in each stator phase (a, b or c) will be given by the two parallel working inverters, i.e. by the CSI and the VSI, as follows:

$$i_s = i_{CSI} + i_{VSI} \qquad (1)$$

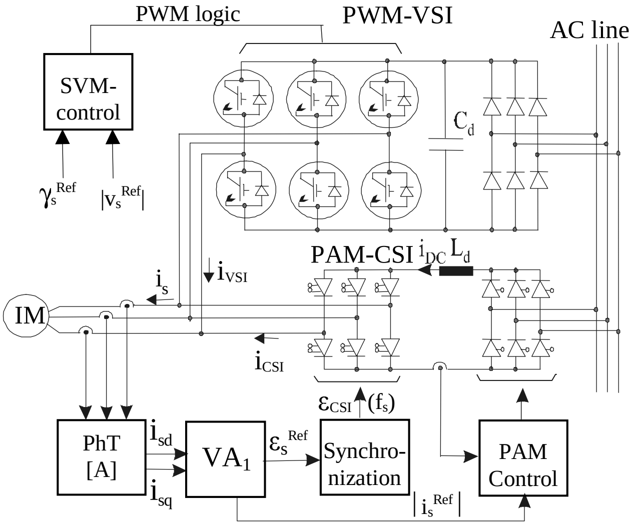


Fig. 1. Tandem Converter Topology.

In this way it is no more necessary to apply PWM procedure to control the whole energy, because a large value of it is transferred through the PAM-CSI, operating with reduced number of commutation. In comparison with an equivalent PWM-VSI, the tandem converter switching losses will be considerable reduced [2], [3].

III. ANALYSES OF VECTOR CONTROL SCHEMES

The analysis of the control schemes and especially the reconfigurable tandem converter from Fig. 2 was performed based on the following criteria:

- Given two vector control structures when common modules exist:
    - Which are the common modules in the same position with the same function?
    - Which are the common modules with different functionality?
    - Which are the particular modules of each reconfigurable structure?
- When reconfiguration condition occurs, is it possible the output variable value transfer for the modules on the same position or no output variable value transfer allowed?
- Is the output variable value transfer of the PI controllers of the different schemes possible?
- Is it possible to give a general mathematical form of all the modules?

Resulting from the analyses, the created module library should be universal for rapid prototyping of any vector control system and from the prototype the implementation should directly result.

One may observe in the module analysis, that the flux computation modules are common for both control schemes (in the example given by the structure in Fig. 2) and so they are one of the most used modules in the module library. As the three flux-computation modules can be computed in a single equation, they will represent a single module, with the following equations [5], [14]:

$$\Psi_{rd} = (1+\sigma_r)\int(u_{sd} - R_s i_{sd})dt - [(1+\sigma_r)L_{\sigma s} + L_{\sigma r}]i_{sd} \qquad (2.a)$$

$$\Psi_{rq} = (1+\sigma_r)\int(u_{sq} - R_s i_{sq})dt - [(1+\sigma_r)L_{\sigma s} + L_{\sigma r}]i_{sq} \qquad (2.b)$$

In this way in the module library for the flux computation there is one module, but when is needed it can easily create all the three flux computation modules ($\Psi_{sd,q}$, $\Psi_{md.q}$, $\Psi_{rd,q}$). In this way one will separate the stator field computation ($\Psi_{sd,q}$) and the flux compensation modules ($\Psi_{md,q}$, $\Psi_{rd,q}$). In addition, the library can handle both flux oriented control schemes, such as rotor flux oriented and stator flux oriented vector control system for any converter fed AC drive.

One of the most common modules (often-used modules) is the Vector Analyser (VA). It is used to compute different modulus of different parameters. Its equations given in the general forms as follows:

$$g = \sqrt{g_d^2 + g_q^2}; \quad sin\,\lambda = \frac{g_q}{g}; \quad cos\,\lambda = \frac{g_d}{g} \qquad (3.)$$

The other two modules, which are also common in many control structures, are the Coordinate Transformation Modules (CooT[D(λ)], CooT[D(-λ)]) with the general equations:

$$g_{sd\lambda r} = g_{sd}\,cos\,\lambda_r \pm g_{sq}\,sin\,\lambda_r; \qquad (4.a)$$

$$g_{sq\lambda r} = g_{sq}\,cos\,\lambda_r \mp g_{sd}\,sin\,\lambda_r; \qquad (4.b)$$

For all the mentioned modules when reconfiguration occurs there is no need for output variable transfer as they do all the computation for the actual sample values.

The modules where one has to consider the output variable transfer is for example the so called control strategy block, represented in this case by the *PI controllers of speed, current and torque*. These modules are called together "control strategy block", as they can be realised in many ways. The control strategy can be implemented using fuzzy logic, neural networks, or other intelligent control methods. The most critical part of the reconfiguration is the output variable value transfer of the PI controllers. In the case when (as is shown in Fig. 2) the output variables of the controllers are different in each state (in one case this is the current reference $i_{sd,q\lambda r}^{Ref}$, and on the other case is the voltage reference $v_{sd,q\lambda r}^{Ref}$), the output variable value transfer cannot be solved.

The modularity is important when the implementation target is reconfigurable hardware such as Field Programmable Gate Arrays (FPGAs) [10].

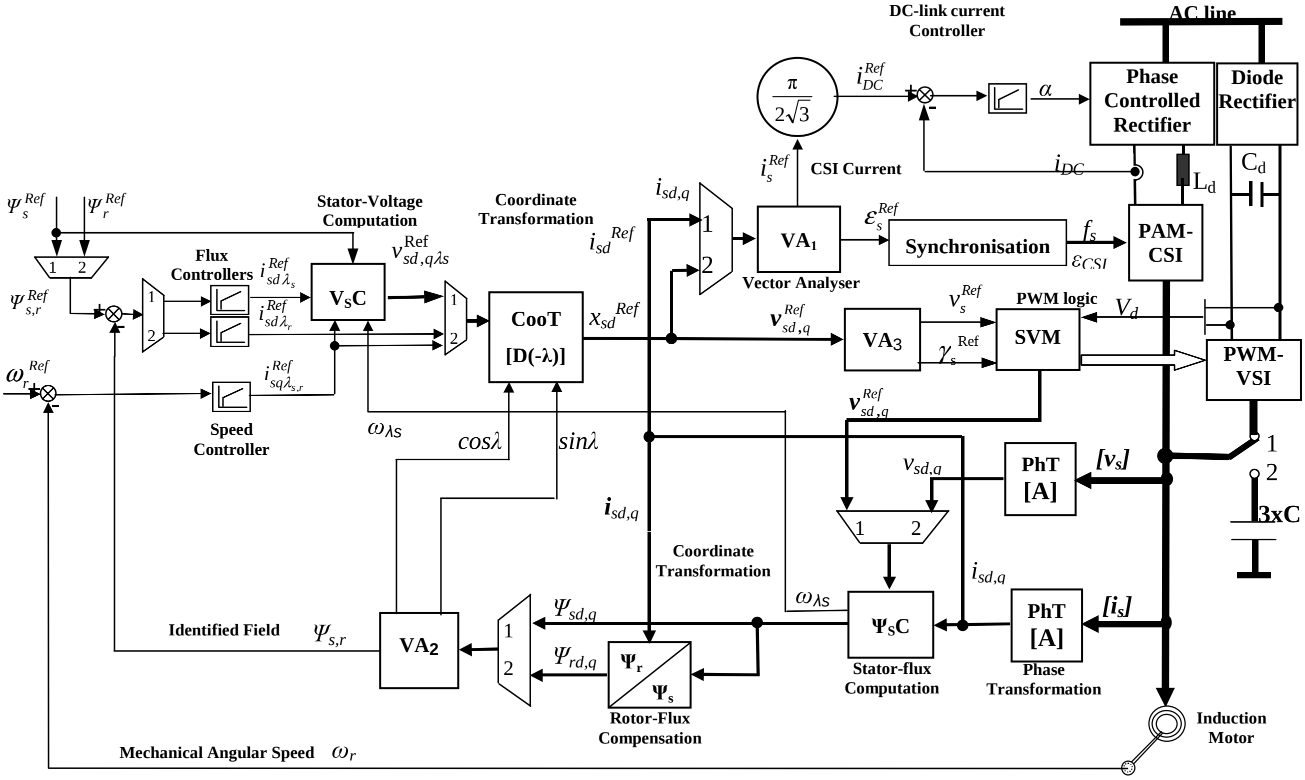


Fig. 2. Reconfigurable control system for the tandem converter-fed induction motor.

IV. LIBRARY MODULES PERFORMANCES

The control of the actuators, represented by the AC motor together with the SFC and the electrical and mechanical sensors assembly, impose real-time performance of the control system algorithm. Also one may conclude that decreasing the sampling period the control system can perform better. On the other hand, decreasing the sampling period, this can impose very short reconfiguration time if reconfiguration is needed, as it is the case of the tandem converter presented in Fig. 2. The performances of the control system directly depend on the performances of each module, when parallel computation is applied.

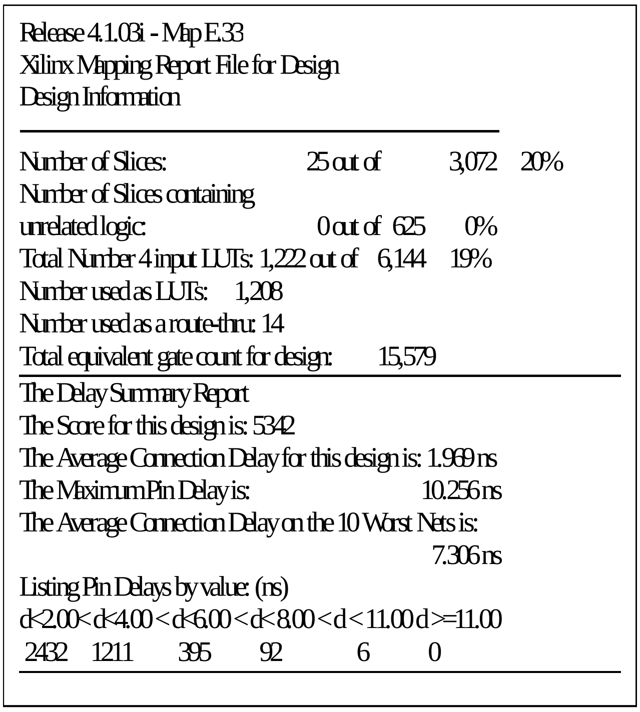
When the performances of each module were analysed one considers two criterions:

1. The hardware resources used by the module in the FPGA, i.e. how many configurable logic blocks (CLB) are needed to implement the function?
2. Which is the worst-case time delay? (i.e. the maximum speed of the module, depending on the FPGA type used.)

There are some module implementation presented in the following:

A. Coordinate Transformation CooT[D(+/-λ)]

Equations (4) represent the general form of the *CooT[D(λ)]* block. The implementation results are presented in Fig. 3. As observed the hardware resources consumed by the coordinate transformations are significant. The equations (4) are implemented in four input 1222 look up tables (LUT), which correspond also to 15579 equivalent gates.



```
Release 4.1.03i - Map E.33
Xilinx Mapping Report File for Design
Design Information

Number of Slices:                    25 out of      3,072   20%
Number of Slices containing
unrelated logic:                      0 out of  625     0%
Total Number 4 input LUTs: 1,222 out of   6,144   19%
Number used as LUTs:     1,208
Number used as a route-thru: 14
Total equivalent gate count for design:      15,579
The Delay Summary Report
The Score for this design is: 5342
The Average Connection Delay for this design is: 1.969 ns
The Maximum Pin Delay is:                          10.256 ns
The Average Connection Delay on the 10 Worst Nets is:
                                                    7.306 ns
Listing Pin Delays by value: (ns)
d<2.00<d<4.00 < d<6.00 < d<8.00 < d < 11.00 d >=11.00
 2432   1211      395      92         6       0
```

Fig. 3. Hardware Resources Consumed And Time Delay Introduced By The Module Coot[D(-λ)]

The high gate count may be a disadvantage of the developed module library, while the time delay introduced by the module is a positive result, which have to be considered when computation speed is important. The maximum pin delay is 10.256 ns.

The quantisation error for both components *d* and *q* is presented in Fig.4. It can be observed that for the data structure used (16 bit 4.12 signed data format) the quantisation error can be accepted.

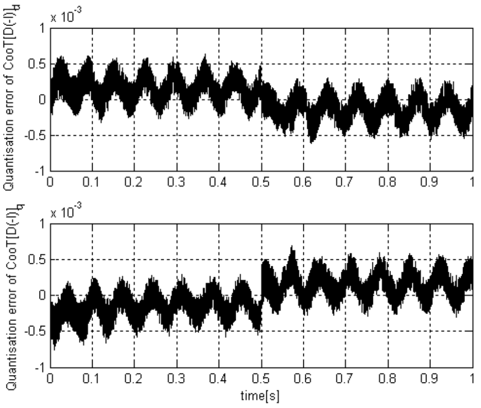
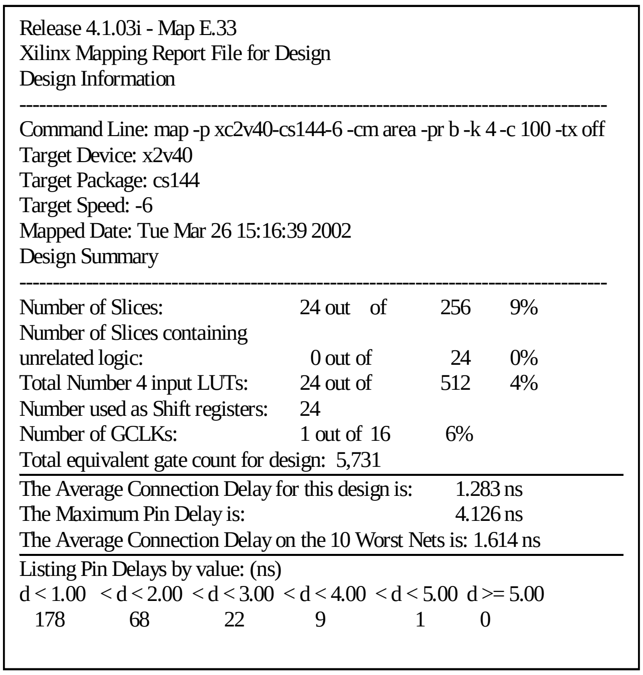


Fig.4. Quantisation error of block CooT[D(-λ)]

The implementation was analysed for Virtex FPGA chips with the constraints to minimise for speed.

B. Flux Controller

One of the modules, which have different structure from the universal computation module [15] is the PI controller. The implementation result of one of the PI controller (flux controller) is shown in Fig. 5



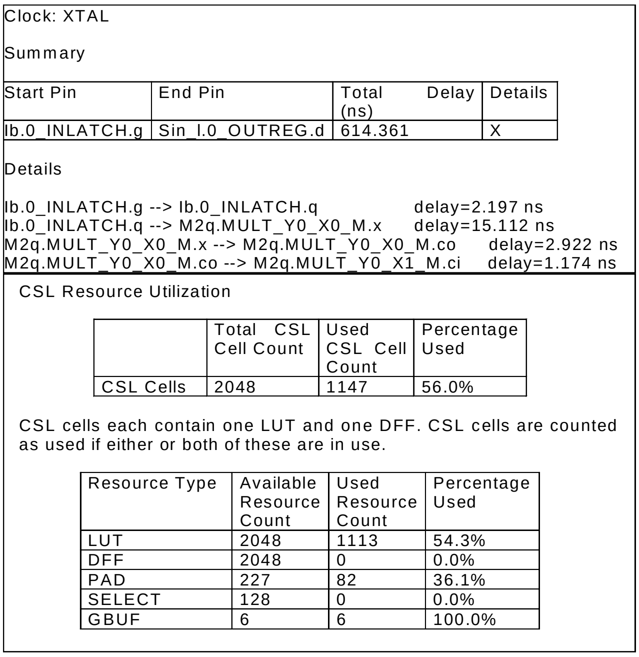
```
Release 4.1.03i - Map E.33
Xilinx Mapping Report File for Design
Design Information
--------------------------------------------------------------------------------------
Command Line: map -p xc2v40-cs144-6 -cm area -pr b -k 4 -c 100 -tx off
Target Device: x2v40
Target Package: cs144
Target Speed: -6
Mapped Date: Tue Mar 26 15:16:39 2002
Design Summary
--------------------------------------------------------------------------------------
Number of Slices:                  24 out   of       256      9%
Number of Slices containing
unrelated logic:                    0 out of          24      0%
Total Number 4 input LUTs:         24 out of         512      4%
Number used as Shift registers:    24
Number of GCLKs:                    1 out of  16      6%
Total equivalent gate count for design:  5,731
The Average Connection Delay for this design is:      1.283 ns
The Maximum Pin Delay is:                             4.126 ns
The Average Connection Delay on the 10 Worst Nets is: 1.614 ns
Listing Pin Delays by value: (ns)
d < 1.00   < d < 2.00   < d < 3.00   < d < 4.00   < d < 5.00   d >= 5.00
   178          68           22            9            1          0
```

Fig. 5. *Hardware resources consumed and time deley introduced by the module flux controller*

Fig. 5 shows the characteristics of the flux controller. Only 9% of the total slices were used to implement the module, which is equivalent to 24 four input LUT and 5731 equivalent gate. The maximum pin delay is 4.126 ns.

C. Vector Analyser (VA)

One of the most resource-consuming module is the Vector Analyser (VA), which computes the equations (3). To show this we present the implementation on Configurable System on a Chip (CSOC). The implementation of this module unfortunately consumes 56% of the CSOC resources, as is shown in Fig. 6. The VA implementation in the CSOC occupied 56% of the available 2048 CSL cells. For this reason the parallel implementation in the CSOC was abandoned and a sequential method will be developed.



Clock: XTAL

Summary

Start Pin	End Pin	Total Delay (ns)	Details
Ib.0_INLATCH.g	Sin_I.0_OUTREG.d	614.361	X

Details

```
Ib.0_INLATCH.g --> Ib.0_INLATCH.q                  delay=2.197 ns
Ib.0_INLATCH.q --> M2q.MULT_Y0_X0_M.x      delay=15.112 ns
M2q.MULT_Y0_X0_M.x --> M2q.MULT_Y0_X0_M.co      delay=2.922 ns
M2q.MULT_Y0_X0_M.co --> M2q.MULT_Y0_X1_M.ci     delay=1.174 ns
```

CSL Resource Utilization

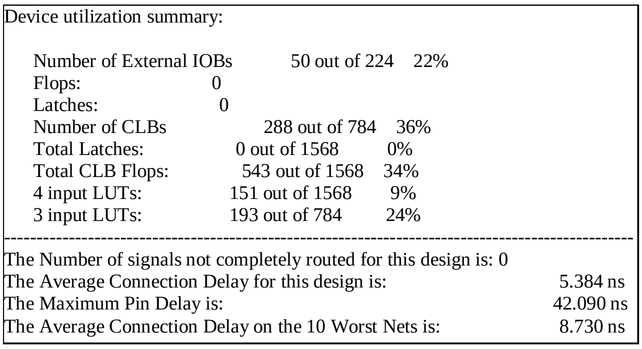
	Total CSL Cell Count	Used CSL Cell Count	Percentage Used
CSL Cells	2048	1147	56.0%

CSL cells each contain one LUT and one DFF. CSL cells are counted as used if either or both of these are in use.

Resource Type	Available Resource Count	Used Resource Count	Percentage Used
LUT	2048	1113	54.3%
DFF	2048	0	0.0%
PAD	227	82	36.1%
SELECT	128	0	0.0%
GBUF	6	6	100.0%

Fig. 6. CSL implementation of Vector Analyser module

In the VA module there is the most resource consuming operation of all the control system. This operation is the square root (sqrt) operation (Fig. 7). The most often used method to implement the sqrt is the LUT method.



```
Device utilization summary:

   Number of External IOBs          50 out of 224    22%
   Flops:                     0
   Latches:                    0
   Number of CLBs                  288 out of 784    36%
   Total Latches:               0 out of 1568        0%
   Total CLB Flops:           543 out of 1568    34%
   4 input LUTs:              151 out of 1568       9%
   3 input LUTs:              193 out of 784       24%
--------------------------------------------------------------------------------------------
The Number of signals not completely routed for this design is: 0
The Average Connection Delay for this design is:                         5.384 ns
The Maximum Pin Delay is:                                               42.090 ns
The Average Connection Delay on the 10 Worst Nets is:                    8.730 ns
```

Fig. 7. FPGA resources used in the sqrt function implementation and time delays.

The implementation method chosen in this case was not the LUT method, as the sum of the two components (d^2+q^2) results from real time computation.

The obtained results for this operation are 9 % of the total resources of the FPGA, which is equivalent to 288 CLBs. The maximum pin delay is 42 ns, which is the worst time delay in the above presented modules.

VI. SIMULATION RESULTS

The Simulation of the CSI-fed vector control system for AC drive was simulated using MATLAB-Simulink® environment. The simulation structure used the module library created and presented in [14]. The induction motor data are: 5.5 *kW*, 50 *Hz*, 220 V_{rms}, 14 A_{rms}, cosφ = 0.735 and 720 *rpm* (4 pole-pairs).

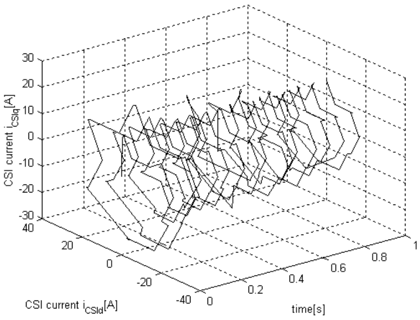


Fig.8. CSI output-current space phasor.

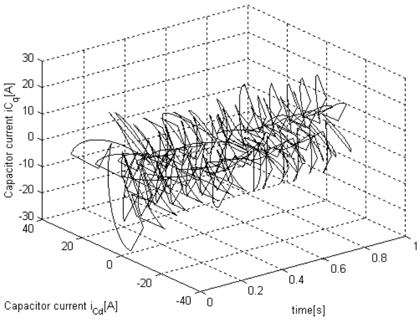


Fig. 9. Capacitor-current space phasor.

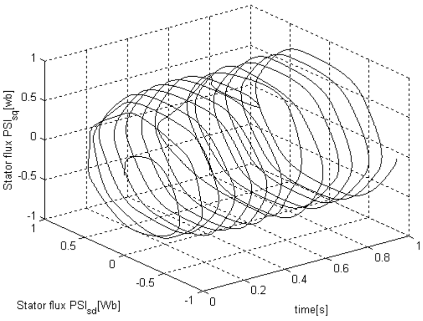


Fig. 10. Stator-flux space phasor.

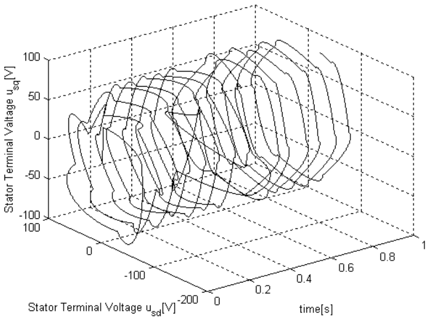


Fig. 11. *Stator-terminal-voltage space.*

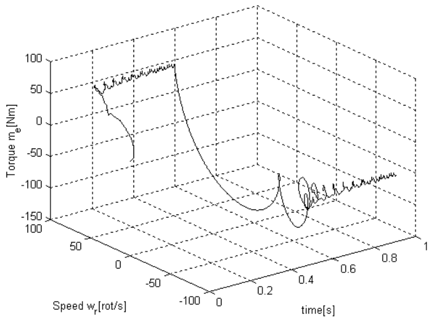


Fig. 12. Dynamic speed-torque mechanical diagram.

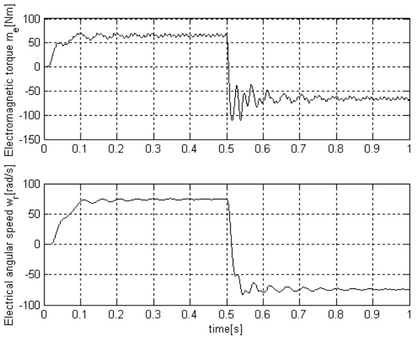


Fig. 13. Electromagnetic torque and electric angular speed.

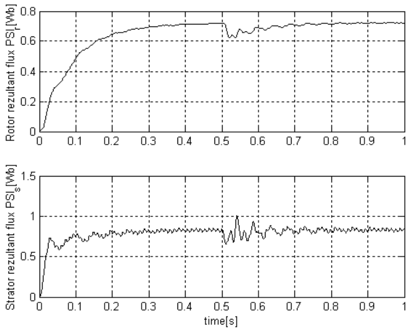


Fig. 14. Rotor and stator resultant flux.

The simulation was performed for the reference value of the electrical angular speed of *+94.2rad/s* and at time 0.5s a speed inversion was made to reference *–94.2 rad/s*. The results are presented at the above figures.

VI. CONCLUSIONS

The performance of the implemented modules and the consumed hardware resources are different, depending on the type of the chip (FPGA or CSOC). As the implementation process uses the method called hardware-software co-design, there were implemented in both type of chips the same modules, in order to decide, which performs better and which occupies less hardware resources since the structure of the CSOC Configurable System Logic (CSL) is slightly different from the FPGA CLB.

From the implementation resulted that choosing FPGAs, may be implemented in parallel all the control structures in a high performance and high-density chip, while choosing CSOC there is need a sequential implementation.

ACKNOWLEDGMENTS

A research project in the subject of the "*tandem inverter*" was realized at the Institute of Energy Technology, Aalborg University, Denmark. Special thanks to Prof. A. Trzynadlowski from Nevada University, Reno, USA for the collaboration in this theme, to Prof. F. Blaabjerg from the Aalborg University and to the Danfoss Drive A/S, Denmark for their generous support.

The authors are grateful to Triscend Inc. and Xilinx Inc. for donations, which made possible the research on some aspects of reconfigurable vector control framework.

REFERENCES

[1] Mária Imecs, P. Bikfalvi, S. Nedevschi, J. Vásárhelyi "Implementation of a Configurable Controller for an AC Drive Control a Case Study", Proceedings of IEEE Symposium on FPGAs for Custom Computing Machines FCCM 2000, Napa, California, USA, 2000, pp. 323-324.

[2] Mária Imecs, J. J. Incze, J. Vásárhelyi, Cs. Szabó "Vector Control Of Tandem Converter Fed Induction Motor Drive Using Configurable System On A Chip", INES 2001 IEEE International Conference on Intelligent Engineering Systems, September 16-18, 2001, Helsinki-Stockholm-Helsinki, Finland-Sweden, pp. 489-495.

[3] Mária Imecs, J. Vásárhelyi, J. J. Incze, Cs. Szabó, "Tandem Converter Fed Induction Motor Drive Controlled With Re-Configurable Vector Control System", Power Electronics Intelligent Motion Power Quality Conference PCIM 2001, Nürnberg, Germany, pp. 341-346

[4] Mária Imecs "Open-Loop Voltage-Controlled PWM Procedures", ELECTROMOTION'99, Patras, Greece, Vol. I, pp. 285-290

[5] Á. Kelemen., Mária Imecs "Vector control of AC Drives", OMIKK publisher Budapest, ISBN 963-593-140-9, Budapest, 1991, pp.304.

[6] A. Sanders "The Design and Implementation of Context Switching FPGA", IEEE Symposium on FPGAs for Custom Computing Machines FCCM 1998, Los Alamitos California, USA, April 15-17, 1998, pp. 78-85.

[7] A. M. Trzynadlowski, F. Blaabjerg, J. K. Pedersen, Niculina Patriciu "The Tandem Inverter: Combining the Advantages of Voltage-Source and Current-Source Inverters", Applied Power Electronics Conference, APEC'98, Anaheim, USA, pp.315-320.

[8] A. M. Trzynadlowski, Mária Imecs, Niculina Patriciu "Modelling and Simulation of inverter Topologies Used in AC Drives: Comparison and Validation of Models", ELECTRIMACS'99, Volume I/3, Lisboa, Portugal, 1999, pp. 47-52.

[9] J. Vásárhelyi, Mária Imecs, J.J. Incze "Run-time Reconfiguration of Tandem Inverter used in Induction Motor Drives", Proceedings of Symposium on Intelligent Systems in Control and Measurement, Veszprém, Hungary, 2000, pp. 138-143.

[10] J. Vásárhelyi "Run-Time Reconfiguration of AC Drive Controllers", Dagstuhl Seminar 0261 on Dynamically Reconfigurable Architectures, June, 2000, Germany, http://www.ee.qub.ac.uk/dsp/HsD/fpl/.

[11] Aubépart F., Poure P., Braun F., Contribution to System-on-Chip in motion control: VLSI design of a digital controller for an induction machine, PCIM 2001 Power Electronics Intelligent Motion Power Quality, June 19-21, 2001 Nuremberg, Germany, pp. 211-216

[12] Cirstea M., Aounis A., McCormick M., Rapid Prototyping of Induction Motor Vector Control System Based on Reusable VHDL Digital Architectures and FPGA Implementation, PCIM 2002 Power Electronics Intelligent Motion Power Quality, May 14-16, 2002 Nuremberg, Germany, pp. 199-203.

[13] Belmimoun M. H., Monmasson E., Sambuis E., Modularity in Code Development for DTSFC Algorithms Implementation on a Fixed-Point DSP, PCIM 2002 Power Electronics Intelligent Motion Power Quality, May 12-16, 2002 Nuremberg, Germany, pp.129-135

[14] Vásárhelyi J., Imecs Maria, Incze J.J., Szabó Cs: *Module Library for Rapid Prototyping and Hardware Implementation of Vector Control Systems*, INES 2002 IEEE International Conference on Intelligent Engineering Systems, May 26-28, 2002, Opatija, Croatia, ISBN 953-6071-17-7, pp. 447-452.

[15] Vásárhelyi J.**,** Imecs MARIA, Szabó CS, Incze J.J., Ádám T., Performances Of Module Library For Vector Control Using Reconfigurable Hardware, IEEE: International Workshop "Trends And Recent Achievements In Information Technology", 16-18 May 2002, Cluj-Napoca, Romania, ISBN 973-8335-49-3, pp.128-133.