IMPLEMENTATION OF A RECONFIGURABLE CONTROLLER FOR AN AC DRIVE CONTROL

Mária Imecs[1], P. Bikfalvi[2], S. Nedevschi[1], J. Vásárhelyi[2]

[1] Technical University of Cluj, Romania, [2] University of Miskolc, Hungary

INTRODUCTION

There are several different approaches to define the reconfigurable systems. “Reconfigurable Computing technology is the ability to modify a computer system’s hardware architecture in real time” as mentioned by VCC (12). Reconfigurable computing is also often called “Custom” or “Adaptive”. There is demonstrated in Hauck (2), Mangione-Smith (6), Mangione-Smith and Hutchings (7), Villasenor and Mangione-Smith (13), and Vuillemin et all. (14) the significant potential for the acceleration of computing for several, general-purpose applications.

Reconfigurable systems are those computing platforms whose architecture is modified by the software to suit the application at hand. This means that within the application program a software routine exists, that downloads a digital design directly into the reconfigurable space of the system. Most of Reconfigurable Computing Systems are plug-in boards made for standard computers and they act as a Co-processor attached to the main micro-processing unit. The computing research community defined the reconfigurable computing as one of very popular subjects.

Maciejowski (5) gave another definition to the reconfigurable systems. According to his opinion, the reconfigurable (control) systems are important when a major failure occurs. In the event of a failure at least three inter-related questions arise

1. Is it possible to control the plant to continue its evolution in safety conditions?
2. Is it possible to control the plant with reducing the original specification parameters?
3. Is it possible to cancel the process without incurring a disaster?

These questions occur primarily in safety critical systems/plants. Reconfiguration is also required if no failure occurs, but the changes in system parameter demand much more effective control law and no adaptive control facilities are implemented.

This paper combines both approaches of reconfiguration trying to give a solution on how to implement a reconfigurable embedded controller for the vector control of an AC drive. The following sections present the background of re-configuration, what make suitable the Triscend’s Configurable System on Chip to be used in embedded control, then some particularities of vector control will be presented and finally implementation of control blocks will be outlined.

BACKGROUND

Field Programmable Gate Arrays (FPGAs) were invented as an alternative to mask-programmable gate arrays. Mainly, these devices are used for implementing of high-volume digital applications, for which no standard off-the-shelf solution exists.

Most of FPGA chips can be re-configured through the configuration memory space. The configuration bits are loaded from an external storage device during the configuration process. The configuration procedure can be part of the power-on process of the FPGA or can be started externally by an external device (configuration manager) at any time during the system evolution.

The FPGA and its several configurations stored in an external memory could be used as multifunctional hardware, with the on-chip changing functionality in reaction to the current demands. Most of applications on reconfigurable hardware focus on the following areas: custom computers, reconfigurable coprocessor boards and reconfigurable processors.

All above-mentioned boards need an external micro-computing element to control and complete the re-configuration. This seems to be their main disadvantage. Considering the mentioned disadvantage, Hauck (2, 3) outlined a Configurable System on a Chip (CSoC) structure that contains all the elements for configurable applications. The possible elements of a Configurable System on a Chip are: microprocessor core, FPGA, DSP resources, RAM, special purpose interface logic and field programmable analogue arrays.

In 1998 Triscend announced the first registered CSoC. This chip was made for general purposes and contains most of embedded system’s logic. The basic elements included within this Configurable System on a Chip are shown in Figure 1.

They are:

- Silicon-efficient, industry-standard processor core, that provides the processing resources.
- On-chip memory, that is capable to store both system code and data. A small, bootstrap ROM supports the system boot-up and re-configuration.
- Programmable logic resources, that support custom defined peripherals, functions and implementing of fast algorithms.
- Test and control logic that supports the system reconfiguration (JTAG).
- Dedicated high performance internal bus to provide flexible connections between system modules.

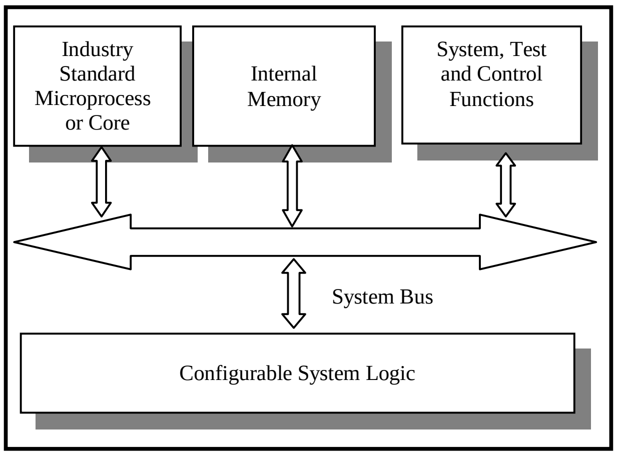

Industry Standard Microprocess or Core
Internal Memory
System, Test and Control Functions
System Bus
Configurable System Logic

Figure 1: Basic structure of the Triscend's CSoC

The Configurable System on a Chip solution with its flexible structure can be used to implement the desired reconfigurable controller described by Maciejowski (5), under the following conditions:

1. External memory is needed to store the several configurations (Configuration Store).
2. Either software or hardware has to be capable to start a reconfiguration on need.
3. The evolution of the system must be predictable in order to pre-compute the possible configurations.
4. The system control states have to be quantified and finite. This condition is due to the finite capacity of the available external memory.
5. The existence of 'high-fidelity' models and of very effective approximation-identification algorithms for multivariable systems.

Considering the application of the reconfigurable control theory for the vector control, one has to analyse if the CSoC corresponds to these demands.

IMPLEMENTATION ISSUES FOR THE VECTOR CONTROL

There are known several dedicated DSP processors for digital motor control. Some successful implementations of vector control are referred in Beierke (1). A DSP implementation of speed-sensor-less induction motor drive using artificial intelligence is presented in Vas et all. (11). Unfortunately, all these implementations and especially their hardware structures do not correspond to the reconfigurable system paradigm.
An example of a vector control scheme for an AC drive is presented in Figure 2 (after Kelemen and Imecs (4)).
This paper does not intend to compare the performances of different possible implementations. It concentrates only on the possibility of the reconfigurable vector control implementation issue.
For this purpose, there are taken under consideration the FPGA chips that are already used in reconfigurable systems and the presented CSoC chip, which is suited for embedded control.
The paper analyses the conditions under which the FPGA and the CSoC chips are able to implement a vector control, pointing out possible disadvantages.
The main problem of a vector control implementation is that of the real time computation. For this reason, usually, DSP chips are involved. Fixed point DSP chips are preferred for two reasons: firstly, because they cost much less than the floating point ones, and secondly, because in most of applications it suffices a dynamic range of 16 bits. Of course, the dynamic range can be increased when using a fixed-point processor by doing software floating-point calculations. Texas Instruments (8) related about a dedicated DSP implementation for vector control, where the control-loop presents a sampling period of 35 μs.

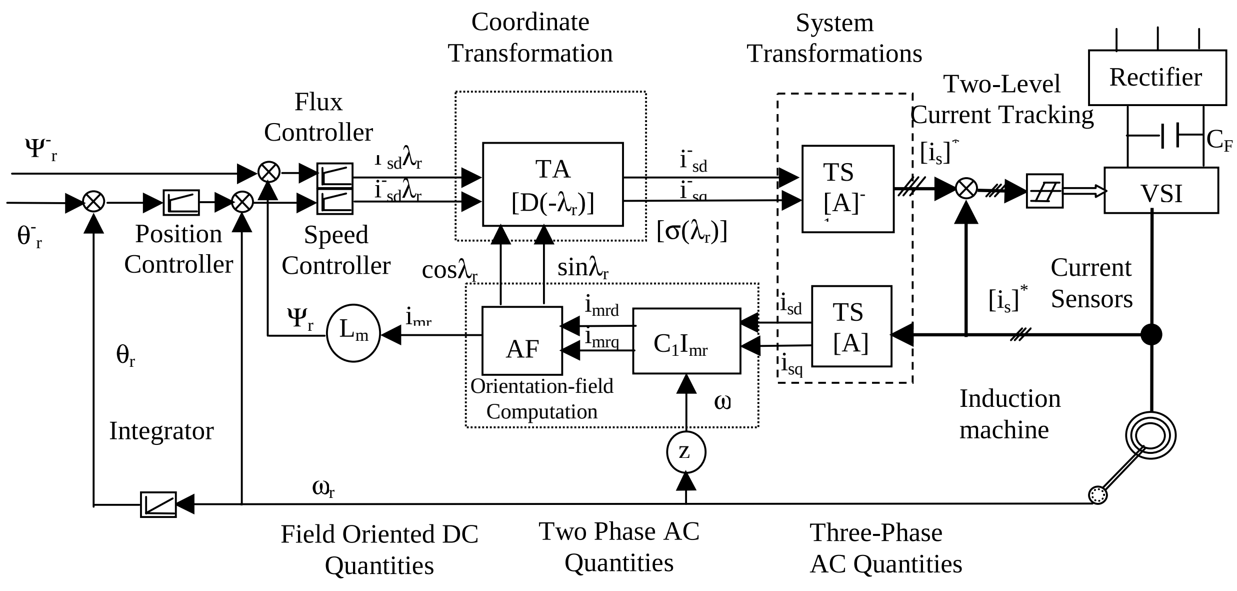


Figure 2. Example of vector control for an AC drive.

The real-time computation problem is often a problem of competition between the computational speed and the method of how the algorithm is implemented.

The main reason why the CSoC chip was preferred against the FPGA chips is its ability to reconfigure itself. This ability means that there is no need for external configuration supervisor when the need for reconfiguration appears. A second reason was that the CSoC chip has a very flexible internal structure.

The Triscend Starter Kit's TE520S40 CSoC chip has the working frequency of 40 MHz, which allows a 10 MHz instruction rate. Considering for start the estimated sampling period of 35 μs, the control algorithm must have less then 350 instructions. So, for the real-time implementation of a vector control for AC drives, there is a need for efficient algorithms in computing of the sine and cosine functions, of the vector transform formulas, and of the matrix multiplication, just to only mention some critical parts of the vector control.

One way to cope with these problems is to implement parallel algorithms in the Configurable System Logic (CSL) blocks of the CSoC. The architecture of the CSL blocks is similar to that of Field Programmable Gate Arrays and these latter already proved their usefulness in implementing several DSP algorithms. It is preferable also to implement parallel algorithms with short time response, i.e. which last one or two machine cycles.

The CSoC chip structure depends directly on the processor core that is implemented inside. Nowadays, the worst performance for speed is obtained for the 8051 micro-controller core. However, this performance can be improved by changing the chip to another with different (better) core.

Some implementation aspects are also to be taken into account. First, the time consuming parts of the control software are analysed and then it is presented a way of their hardware implementation in the CSL.

One critical part of the control structure refers to the calculation of the sine and the cosine functions. These functions can be implemented in two ways. One way is to form a look-up table with their pre-computed values. Another way is to calculate them at the beginning of the initialisation phase using Taylor series expansion, and to store the results in the on chip RAM memory in a same form of a look-up table. Then, this table can be accessed very fast from the CSL by using the DMA technique.

The algorithm for computing the value of $\sin(\alpha)$ is:

```
IF ($\alpha$>=0.0)
    $\sin(\alpha) = (((0.0372*\alpha - 0.2338)*\alpha + 0.0544)*\alpha + 0.9826)*\alpha + 0.0013$
ELSE                                                                    (1)
    $\sin(\alpha) = (((0.0372*\alpha + 0.2338)*\alpha + 0.0544)*\alpha - 0.9826)*\alpha + 0.0013$
```

where the angle α is considered in radians.

The values of the sine and the cosine functions are calculated as an unsigned 16 bit data corresponding to the allocation given in the Table 1.

TABLE 1. Unsigned 16 bit data representation for sine and cosine functions

Angle (radians)	Assigned value (Hexadecimal)
0	0000
π/2	4000
π	8000
3π/2	C000

Another critical part in the real-time implementation is related to the co-ordinate system transformations. The direct transformation equations are given by:

$$\begin{bmatrix} g_d \\ g_q \\ g_0 \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} * \begin{bmatrix} g_a \\ g_b \\ g_c \end{bmatrix} \tag{2}$$

The reverse transformation equations are:

$$\begin{bmatrix} g_a \\ g_b \\ g_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 1 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 1 \end{bmatrix} * \begin{bmatrix} g_d \\ g_q \\ g_0 \end{bmatrix} \tag{3}$$

g_a, g_b, g_c are the three phase system variables (currents or voltages), g_d, g_q are the two-phase system variables and g_0 is the zero component of the three-phase system reflected in the two-phase co-ordinate system. It is given by the equation:

$$g_0 = \frac{g_a + g_b + g_c}{3} \tag{4}$$

One can obtain the further co-ordinate transformations for the transformed quantities with the relations:

$$\begin{bmatrix} g_{sd} \\ g_{sq} \end{bmatrix} = \begin{bmatrix} \cos(\lambda) & \sin(\lambda) \\ -\sin(\lambda) & \cos(\lambda) \end{bmatrix} * \begin{bmatrix} g_d \\ g_q \end{bmatrix} \tag{5}$$

λ represents the instantaneous value of the phase shift between the considered and the real reference frames.

All these equations can be implemented more or less easily in the CSL space of the CSoC.

RECONFIGURATION ASPECTS

In the present case study, the process to be controlled is a voltage-source inverter fed AC motor drive. The CSoC is the hardware support for the controller. The necessity of reconfiguration is based upon the practical observation that the performance of a vector control drive depends on the flux identification method, on the load characteristics (dynamic and/or static) and on the range of the speed. Concerning these aspects, there have been developed several versions for vector control. One may find these control schemes in the corresponding literature (Vas (10), Kelemen and Imecs (4)).

The rotor-flux oriented vector control is apparently simpler to implement and related by Vas (9) as widely used. One drawback of this method is the low efficiency at lower ranges of speed. Another problem is related to the need for adaptation of the control algorithm to the rotor resistance change during operation due to the heating. For lower speed range, the stator-flux oriented vector control is preferred. The above-mentioned control schemes have different structures, but in principle, each of them can be implemented in a CSoC.
This paper intends to introduce in one more idea. This consists of the application of the reconfigurable controller concept for implementing different control structures for the AC drives. In fact, the CSoC that implements at one moment one controller structure, can be used not only to implement, but also to switch to another control scheme. In this way, the disadvantage of using adaptive control can be avoided.
Each control structure can be seen as a distinct state of a state machine. In fact, each state represents a different hardware configuration of the CSoC.
Figure 3 presents a possible implementation for two controller structures into two different configuration states (S1, S2) of the CSoC. The transition from one state to the other can be determined by the state variables of the controlled system. If a transition condition occurs (i.e. the motor speed reference transits a limit value) the need for reconfiguration is fulfilled. The controller will start automatically a reconfiguration process and will change its configuration.
As example, the state machine switches between two control schemes. These could be: the rotor-flux oriented vector control allocated to state 1 and the stator-flux oriented vector control allocated to state 2. In principle, the state machine can be extended to implement other states, respectively other control schemes, as well.

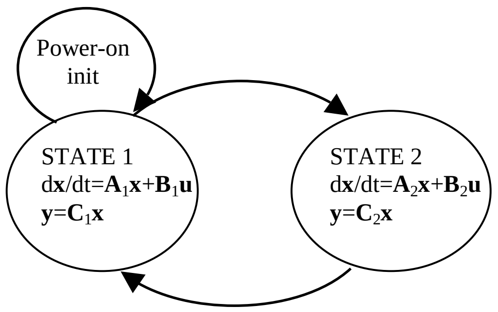

Power-on init
STATE 1
$d\mathbf{x}/dt=\mathbf{A}_1\mathbf{x}+\mathbf{B}_1\mathbf{u}$
$\mathbf{y}=\mathbf{C}_1\mathbf{x}$
STATE 2
$d\mathbf{x}/dt=\mathbf{A}_2\mathbf{x}+\mathbf{B}_2\mathbf{u}$
$\mathbf{y}=\mathbf{C}_2\mathbf{x}$

Figure 3. The state transition graph of the reconfigurable controller

The first attempt of realisation showed that the implementations of the two vector control strategies are hard resource consuming. Therefore this problem needs further research. As example, the matrix multiplication of general form presented in Equation (3) consumes 33% of the CSL resources from the TE520S40 Triscend CSoC. Figure 4 presents this result.
Figure 5 presents the situation, when supplementary to the matrix multiplication, the co-ordinate transformation of Equation (5) and the sine/cosine functions are included. It shows a consumption of 75% of resources.

CONCLUSIONS

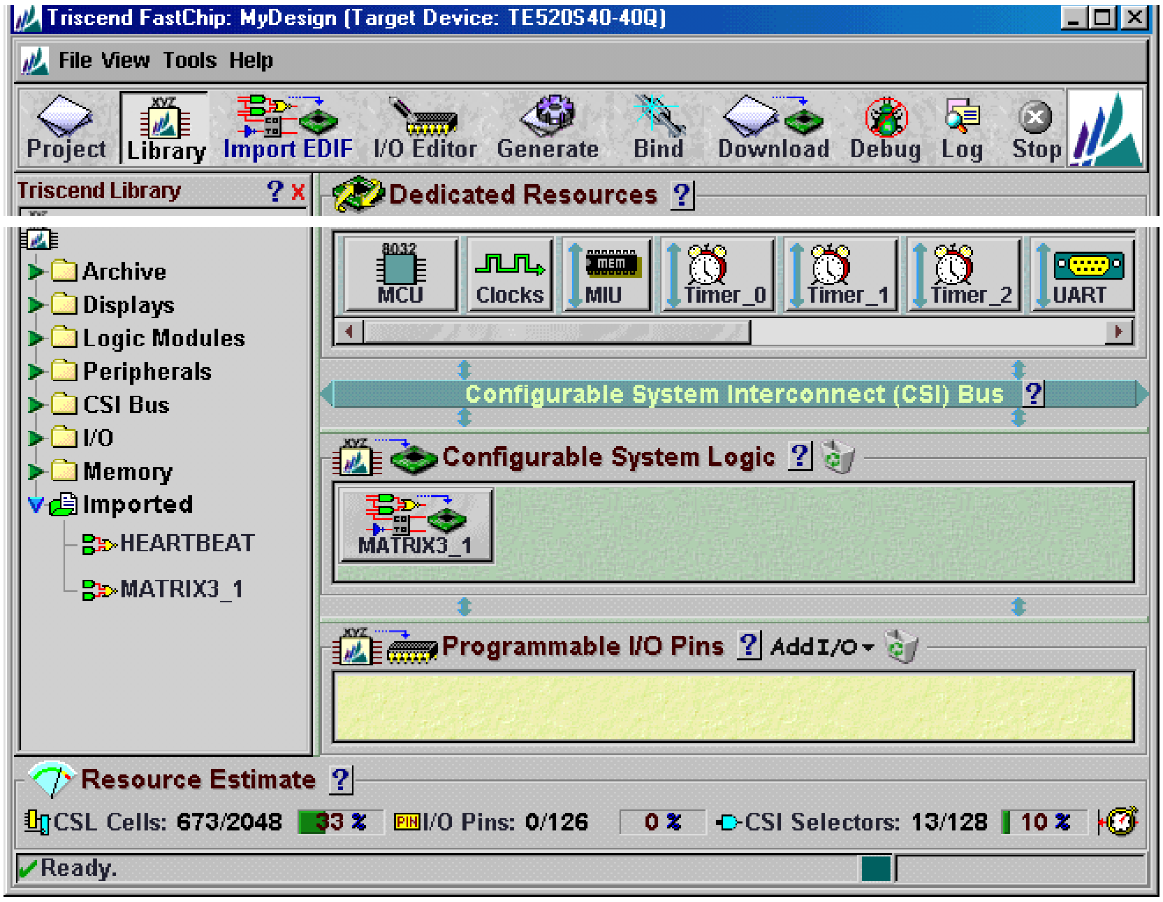
The paper presented a possible implementation of the vector control for an AC drive using the Triscend CSoC. The idea of possible reconfiguration for control was also introduced in a form of a suitable configured and implemented state machine that can realise this task. Sharing of the available resources represents a problem of further investigation.
The changes in the control law by reconfiguration may improve the performance of the controlled system and can avoid the more sophisticated adaptive control.
Further research work needs to investigate the effects of the reconfiguration transition process, too. The main problem that seems to appear consists of how can be managed the drive during the reconfiguration process. Even if this event needs very little time, the problem of loosing the control may appear.

ACKNOWLEDGEMENT

The authors would like to thank Chris Balough and Triscend Inc. for their support and help. Triscend supported our work with the FastChip Starterkit and the FastChip 1999 software.

REFERENCES

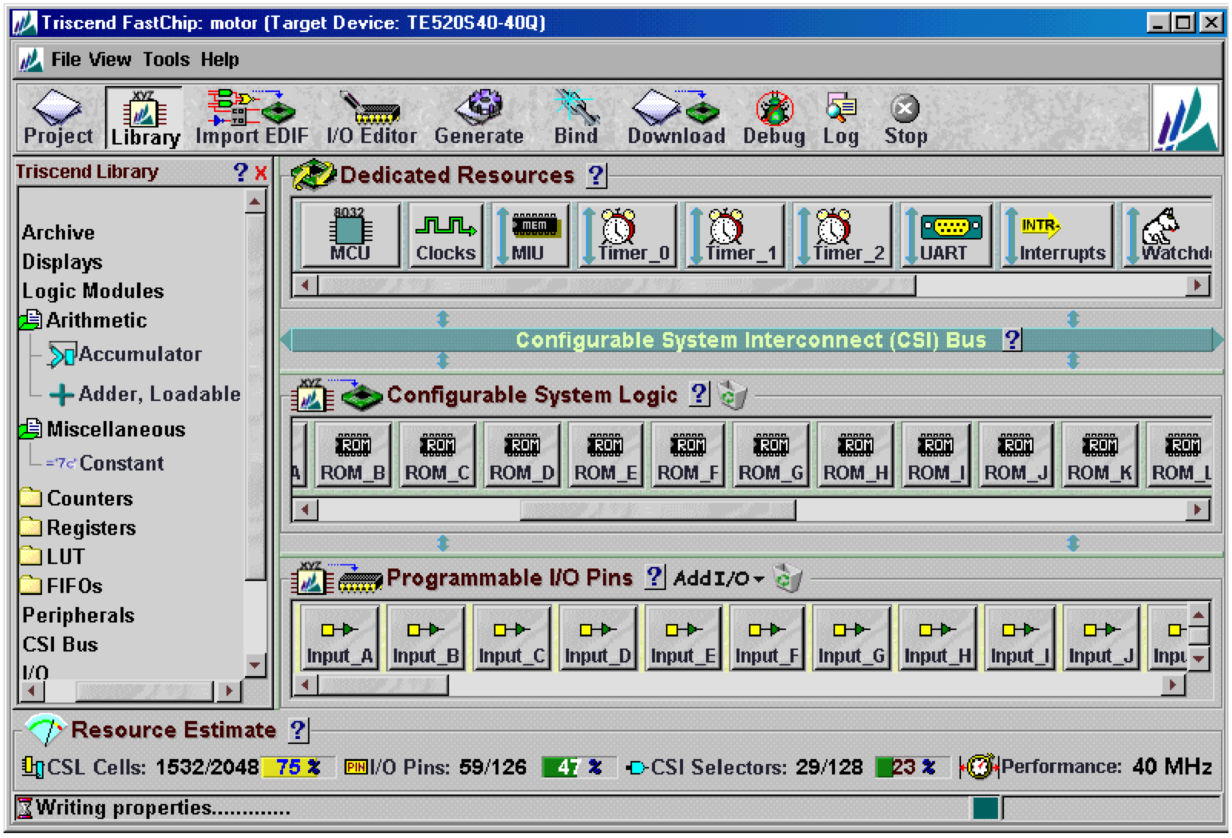
1. Beierke, S., 1994, EPE Symp. on El. Drive Design and Appl., 361-365.
2. Hauck, S., 1998, Proc. of the IEEE, 86, 4, 615-639
3. Hauck, S., 1998, 5th Canadian Conf. on FPD
4. Kelemen, A., Imecs M., 1991, "Vector control of AC Drives", OMIKK Publisher, Budapest, HU
5. Maciejowski, J.M., 1997, ECC97, 107-130.
6. Mangione-Smith, V.B., Hutchings, L., 1997, Workshop on Reconfigurable Architectures, 81-96
7. Mangione-Smith, V.B., Hutchings, L., Andrews, D., DeHon, A., Ebeling, C., Hartenstein, R., Mencer, O., Morris, J., Palem, K., Prasanna, V.K., Spaanenburg, H.A.E., 1997, Computer, 30, 12, 38-43
8. Texas Instruments, 1999, TMS320C24X DSP Sol. CD-ROM
9. Vas, P., 1990, "Vector Control of AC Machines", Oxford University Press, UK
10. Vas, P., 1996, "Electrical Machines and Drives. A Space Theory Approach", Oxford University, UK
11. Vas, P., Stronach, A.F., Rashed, M., Neuroth, M., 1999, Proc. Intelligent Motion, 113-117
12. VCC Inc., 1997, http://www.vcc.com
13. Villasenor, J., Mangione-Smith, V.H., 1997, Scientific American, 66-71
14. Vuillemin, J., Bertin, P., Roncin, D., Shand, M., Touati, H., Boucard, P., 1996, IEEE Trans. on VLSI Systems, 4, 1, 56-69



CSL Cells: 673/2048 33 % I/O Pins: 0/126 0 % CSI Selectors: 13/128 10 %

Ready.

Figure 4. Resources used when implementing matrix multiplication



CSL Cells: 1532/2048 75 % I/O Pins: 59/126 47 % CSI Selectors: 29/128 23 % Performance: 40 MHz

Writing properties............

Figure 5. Resources used when implementing matrix multiplication, sine/cosine functions, coordinate transformation