RAPID PROTOTYPING OF DEDICATED SYSTEMS BASED ON SHARED MEMORY ARCHITECTURE: METHOD AND EXAMPLE

Grzegorz Rubin[1], Mirosław Omieljanowicz[2], Alexander Petrovsky[2]

[1] The State College of Computer Science and Business Administration Lomza, Poland

[2] Faculty of Computer Science, Bialystok University of Technology, Bialystok, Poland

Abstract: The aim of this paper is to present the method of rapid prototyping for reducing development cost of dedicated systems. In this paper the designing method for real-time embedded systems is proposed. At first the principle of specific universal balanced architecture is shown. Approach is based on modeling using modification of Petri nets called Hardware Petri Nets implemented in form of CAD software. Dedicated system is made using special computation architecture on FPGA. An example of designing processor for TVDFT is also given.

Keywords: rapid prototyping, shared memory architecture, FPGA

1. Introduction

The design and implementation of dedicated DSP (Digital Signal Processing) system is quite complex and time-consuming. Nowadays objective in times of high market pressure and ever decreasing time-to-market, automatization of the design and implementation are crucial. Finding an universal solution suited to wide range of DSP algorithms is permanently actual task. Rapid prototyping aims to reducing development cost[1]. A prototype is constructed prior to the system production version to gain information that guides analysis and design. This paper presents the method of designing real-time dedicated systems prepared for implementation in FPGA. In this approach a modification of Petri nets called Hardware Petri Nets (HPN)[2] is used. Implementation in FPGA chip using special computation architecture called dynamic reconfigurable computation architecture (DRCA) is also presented. Reconfiguration is made by altering the connections and control scheme of the universal computation

Zeszyty Naukowe Politechniki Białostockiej. Informatyka, vol. 9, pp. 105-118, 2012.

module (UCM). The main advantage of proposed method is that there is no need to change hardware even if all used algorithms must be changed. There is a possibility to make all design processes to be almost fully automated from modeling to working prototype. Proposed method could be described in the form of the following algorithm:

1. Specification of tasks including: functional requirements, real-time requirements, bandwidth, maximum power consumption, maximum cost, etc.
2. Functional modeling and testing .
   2.1. The choice of algorithms which conform functional requirements from the point 1.
   2.2. Functional modeling using available computer programs (i.e. Matlab, Simulink).
   2.3. Functional verification of algorithms and if it doesn't comply with requirements: change algorithms (back to the point 2.1) or (as a last resort) change the specification (back to the point 1).
3. Synthesis of a processor structure.
   3.1. Design of a processor architecture based on DRCA concept.
   3.2. Mapping (selected at the point 2) algorithm to the chosen (at the point 3.1) architecture.
   3.3. Modeling and testing using application based on proposed HPN and if it doesn't comply with functional requirements: change the architecture (back to the point 3.1).
4. Synthesis of the hardware.
   4.1. Modeling calculation timing and generation of the control microcode using application based on proposed HPN.
   4.2. Synthesis of the processor and testing according to the all requirements specified at the point 1. The non-compliance forced to modify or change the architecture (back to the point 3.1).
5. End.

The next part of the article shows cosiderations which lead to choice of the architecture for universal computational module and presents an example of using the proposed method to design a processor for TVDFT transformation .

2. Principle of universal computation module and dynamic reconfigurable internal architecture

At the architectural level the main interest is to plan the overall organization of the compound system using processing elements (PE), memories, communication channels and control elements. Finding a universal solution suited to wide range of DSP

algorithms is crucial task in terms of rapid prototyping . One of possible approaches is called shared-memory architecture (fig. 1 A).

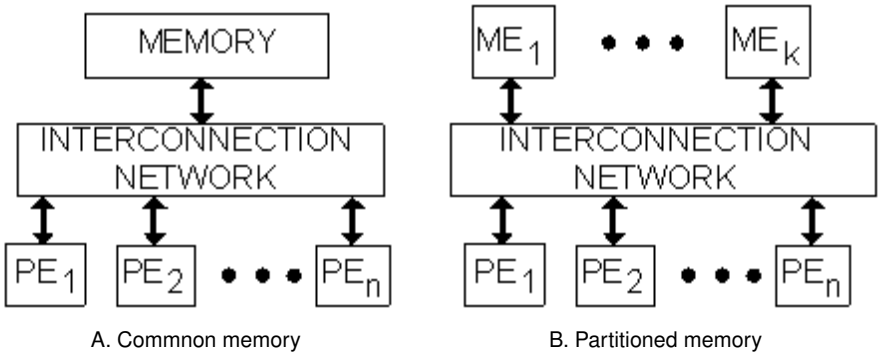


Fig. 1. Shared memory architecture

The idea is very simple. In order to provide simultaneously PE's with input data, the shared memory is partitioned into blocks. Using a rotating access scheme each processor gets access to the memories once per N (number of PE's) cycles. During this time processor either writes or reads data from memory. All PE's have the same duration time slot to access to the memories and access conflict is completely avoided. Shared-memory approach for DSP application is detailed in [3]. The well-known disadvantage of shared-memory architecture is memory bandwidth bottleneck. In order to avoid bandwidth bottleneck and simultaneously provide the processors with several (K) input data, the shared-memory is partitioned into K memories (fig.1 B).

In this paper a special instance of that architecture called universal computation module (UCM) is presented. The main target is to find balance between complexity of interconnection network, type of computation model of PE's (serial vs. parallel), number of PE's and memory size. Chosen compromise should fulfill following factors: required performance, minimal power consumption and cost in terms of chip area. Another important requirement is to create flexible, easy reconfigurable architecture suited to wide range of DSP algorithms.

Processing elements (PE's) usually perform simple memory less mapping of the input values to a single output values. The PE's can be in parallel or serial form. Memory elements (ME) comparing to PE's are slow. It is obvious that there is a need for additional register to fulfill performance requirements. By bit parallel PE high

speed register will play a trivial (one word) cache memory role. By bit serial PE there must be a shift register. The RAM addressing requires only cyclic work. Number of RAM words should be enough to store all variables accordingly to realized algorithm. Interconnection network (ICN)should provide the communication channel needed to supply PE's with proper data and to store results in the proper memories. The data movement should be kept simple, regular and uniform. Major design issues involve the topology of the communication network and its bandwidth.

There is a variant of shared-memory architecture that seems be well-matched to wide range of DSP algorithms and can be very well-balanced in term of processing capacity and communication bandwidth. This modification splits interconnections

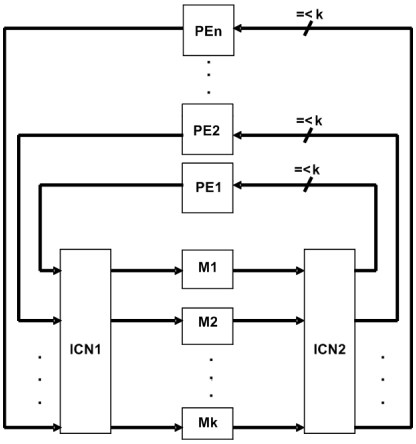


Fig. 2. Shared memory architecture with split intercommunication network [3]

into two parts (fig.2). In this architecture to obtain balance [3] there should be:

$$T_{PE} >= 2 * N * \frac{T_M}{W_M} \tag{1}$$

where: $T_(PE)$ - operation time of PEs, N - number of PEs, T_M - memory access time, W_M - word width in memory. Then number of PEs should be:

$$N <= (T_{PE}/T_M) * W_M/2 \tag{2}$$

Typically $T_{PE} = \frac{T_M}{4}$ and data word length is 16 to 24 bits. Thus, balanced shared-memory module will have 2 or 3 PE's. After that a trade-off between parallel and serial form of PE's and ICN was made.

It is obviously that bit-parallel version of PE's have several times higher performance than bit-serial one at the same clock. However, taking into account whole module with PE's, input and output registers, memory and interconnection network, advantage of parallel form is not so clear. Including power consumption and chip area, serial form could be more convenient. Generally smaller chip area and smaller clock lead to smaller power consumption. The requirements of the PE is that it completes its operation within the specified time limit. Self explanatory chip area of single serial PE is much smaller then parallel PE, but to get the same performance needs faster clock. Parallel PE's leads to more connections lines, consistently more chip area and power. Parallel PE looks to have several times bigger computational throughput (than serial by the same clock), however when consider more than one PE in shared memory architecture it could be impossible to use high speed clock because of noise in signal propagation on parallel buses. Otherwise control part of whole system in serial PE version may be in micro program fashion, where implemented algorithm will be changed by the way of changing control memory contest. Using parallel PE control part must be significantly changed due to change type of computational task.

In universal architecture on fig. 2 it is convenient to use serial bit PE's. Only single wire is required for each signal, hence interconnections are simple and consume small chip area. The memory has been split into K logical memories that K values can be accessed simultaneously. Hence every PE can have K inputs and memory conflicts are avoided. However to have nice working architecture there is a need to change the data format from serial to parallel and parallel to serial without influence of interconnections. It is easy to accomplish using shift register as a part of memory block as is shown on fig. 3.

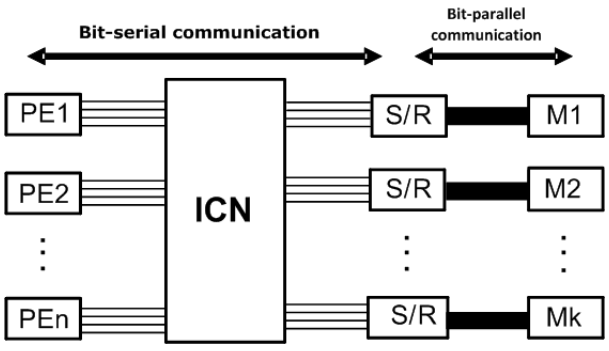


Fig. 3. Shared memory with bit-serial and bit- parallel communication [3]

Above brief considerations shows that shared-memory architecture with 2-3 serial PE's can be easier to implement and well suited to wide range of DSP algorithms (dynamic reconfiguration made by microcode change). Processing elements used in that type of shared memory architecture should have three functions: bit-serial full addition (inc. carry), bit-serial multiplication, negation and two serial inputs and one output. Other arithmetic operations will be done as a sequence of additions. Block diagram of proposed UCM with 3 PE's is shown on fig.4. Because of three PE's, UCM is equipped with six shift registers and two independent memories. Those registers are used as single word cache memory and as serial to parallel and parallel to serial translators on communication path to RAM memory. Hence interconnection network is very simple. This leads to small chip area and possibility of using high speed clock. To achieve flexibility in organization of computation process, connection network should enable access to any RAM for every PE. Thus more than 3 PE's give more connections lines and more chip area (only for wires) so the balance between chip area for logic and connections will be loosed.

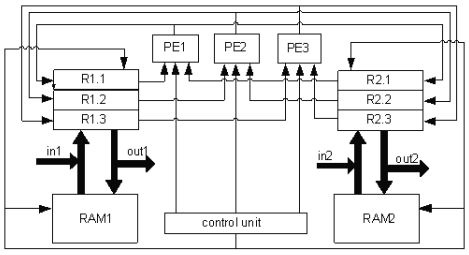


Fig. 4. Proposed universal computation module - UCM

Proposed kind of shared-memory architecture in form of UCM offers good balance in terms of chip area, power consumption, computational throughput and flexibility but offers relatively small performance to run advanced DSP algorithms. To fulfill requirements of complex DSP task more computation power is needed. It can be achieved by connecting some number of UCM's. Changing number of UCM's and way in which they are connected and controlled we get dynamic reconfigurable computation architecture - DRCA. In such a manner almost every required performance could be achieved.

To be able to use proposed UCM in rapid protoyping process it is necessary to have specific modeling method (taking into account parallel processing). That can be done using Petri Nets theory. There are few known examples of that approach [2,

5, 6]. For proposed approach a special modification of Petri Nets called Hardware Petri Nets (HPN)was developed Proposed HPN allows for deadlock prevention [7] in parallel processes. Every transition corresponds to enable signal in physical device. Moreover there is possibility to design hierarchical structure of the net. Every transition and place can be designed as the subnet. Simulation of hierarchical structure is also possible.

Developed HPN concept was implemented as CAD-type application. This allows automatation of the designing process (important feature in rapid prototyping method). Produced software environment has a graphical model creation tool and allows simulation of algorithms step by step. A sample graph corresponding to UCM is presented on fig.5. Formal analysis, hierarchical designing and simulation are possible too. Each part of designed algorithms can be independently simulated. That is useful for error correction. As the result of software simulation we have control

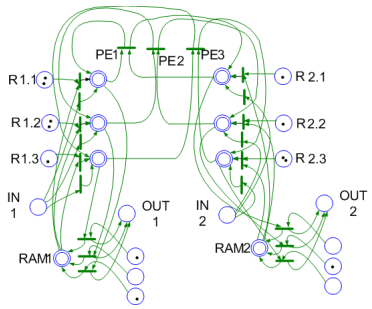


Fig. 5. Hardware Petri Net model for universal computation block

vectors for FPGA device. These vectors can be written as a text file, then loaded to Xilinx FPGA device simulator. Such approach allows for better scheduling, because of running every elements of architecture as fast as it is possible, when proper data are ready for processing. These operations ending protopyting process.

In the next part of this article the example of realization of TVDFT transformation based on proposed approach is presented.

3. Rapid prototyping of TVDFT

A wide variety of signal processing functions can be hosted on the UCM, including complete subsystems that encompass multiple algorithms. The TVDFT transforma-

tion [8] will be used as the example. TVDFT is given by equation:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\varphi(n,k)} w(n), k = 0..K \tag{3}$$

where: $X(k)$ - spectral component corresponding to k- th harmonic, N - length of analysis frame, $x(n)$ - input signal, $w(n)$ - time window, K - number of orders in input signal. Function $\varphi(n,k)$ is given as

$$\varphi(0,k) = 0; \quad \varphi(n,k) = \sum_{i=0}^{n} \frac{2\pi k(f_0(i) - f_0(i-1))}{2F_s} \tag{4}$$

where $f_0(i)$ is fundamental frequency at time specified by i, F_s is sampling frequency. In case of linear change of fundamental frequency formula (4) can be written as follows:

$$\varphi(n,k) = \frac{2\pi nk}{F_s}(f_0 + \frac{2\Delta f}{2N}) \tag{5}$$

where: f_0 -fundamental frequency at the beginning of analysis frame, Δf is fundamental frequency change within analysis frame. Hence, TVDFT formula (3) can be written as follows:

$$Re\, X(k) = \sum_{0}^{N-1} x_w(n) \cos(\frac{2\pi nk}{F_s}(f_0 + \frac{2\Delta f}{2N})) \tag{6}$$

$$Im\, X(k) = \sum_{0}^{N-1} x_w(n) \sin(\frac{2\pi nk}{F_s}(f_0 + \frac{2\Delta f}{2N})) \tag{7}$$

where: $x_w(n) = x(n)w(n)$.

Formulas (6),(7) show, that for practical realization of TVDFT, two sine wave generators with linear change of frequency [4] and one multiplication block must be used. Block diagram of computations is show on fig.6. According to this at the first step we must calculate value of sine and cosine function (generators blocks) then at the second step we must multiply that value by input signal (transform block). The sine and cosine function algorithm [4, 9] was based on formula $(-1)^k(\bar{A_n} - kM)[M - (\bar{A_n} - kM)]$, where n - number of current sample, M - amount of samples per half period, $\bar{A_n}$ - value of n'th sample, k - integral part from division n/M. This equation leads to computation block diagram presented on fig.7.

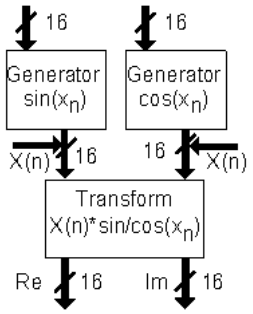


Fig. 6. Block computation diagram for TVDFT

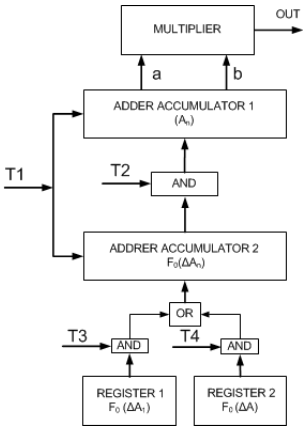


Fig. 7. Generator with linear time-varying frequency [9]

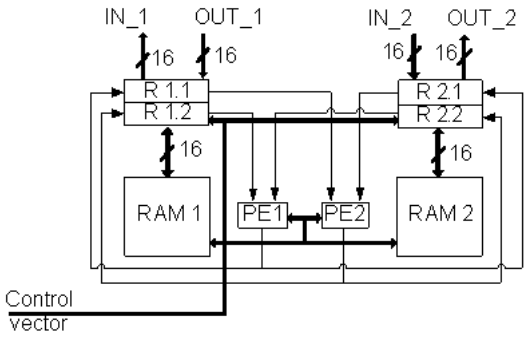


Fig. 8. UCM for sine/cosine generators

The balanced computation module for such algorithm needs two serial PE's, so there are only four shift registers for communication between RAM's and PE's. Thus it gives UCM in form shown on fig.8. Size of memories is 3 of 16- bit words (RAM1[0..2] and RAM2[0..2]). According to block diagram of the TVDFT to accomplish whole task we need 3 computation modules. For improving performance it is possible to use parallel or cascade connection of computation units. On fig.9 two UCM's calculate sine and cosine values in parallel form and then the results get into the inputs of additional UCM unit, where, input signal $x(n)$ is multiplied by sine value. The result (according formula (6),(7)) accumulates in memory .

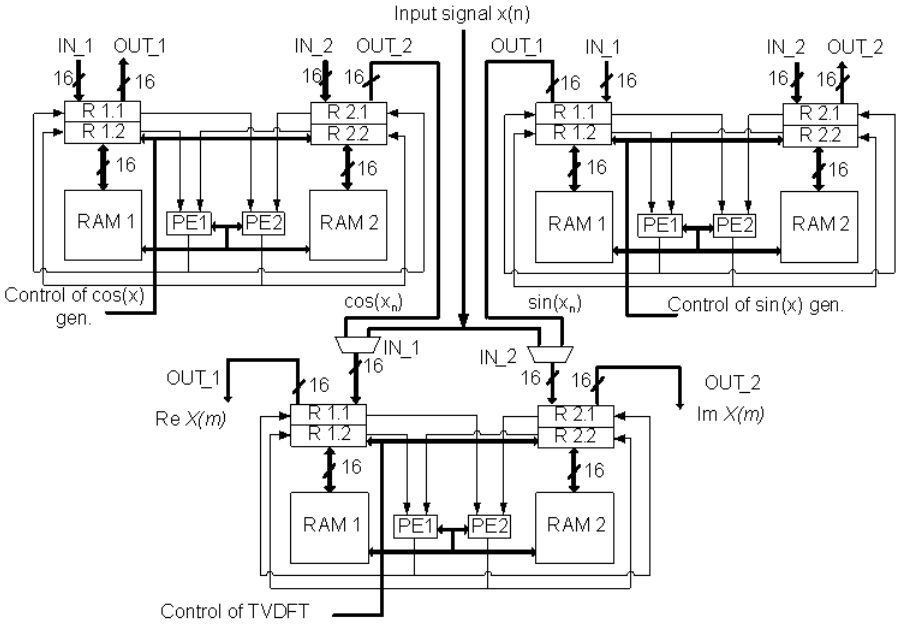


Fig. 9. Three UCM's for TVDFT realization

The next prototyping phase is to make computation model and to create all necessary control signals. First stage was for generator block and second one for transform block. Set of operations for generator block mapped on HPN model is given bellow (operations from 1 to 8 repeats for every N - value of sine/cosine function):
1) Load data into input registers.
R1.1<- IN1, R2.1<- RAM2[0]

R1.2<- RAM1[2], R2.2<- RAM2[1]
2) Calculations.
PE1<- R1.1+R1.2, PE2<- R1.2+not(R2.2)
3) Writing the results into output registers.
R2.1<- PE1, R1.1<- PE1, R2.2<- PE2
4) Writing the result from output registers to memories.
RAM2[2]<- R2.2, RAM1[0]<- R1.1, RAM2[1]<- R2.2,
5) Load data into input registers.
R1.1<- RAM1[0], R2.1<- RAM2[1]
R1.2<- RAM1[2], R2.2<- RAM2[2]
6) Calculations.
PE1<- R1.1+R1.2, PE2<- R1.2(R2.2)*
7) Writing the results into output registers.
R1.1<- PE1, R2.1<- PE1, R2.2<- PE2
8) Writing the result from output register to output.
Out1<- R2.2
where: *IN1* - at first step starting frequency value and then relative value of frequency changing per step (in case of the generator with time-varying frequency[4]), *Out* - value of sine/cosine function.

According to presented set of operation and prototyping method we built HPN model (fig.10) and made appropriate simulation. It allows for mapping and scheduling (see results on fig.11) for first stage - given sinus generators algorithm. It the same way the second stage was made - modeling of transform block (fig.6) using HPN.

As the end of prototyping process an FPGA implementation of TVDFT computational unit was made on XILINX VIRTEX-II family. Obtained result was compared to prototype created in traditional way, i.e using HDL language and libraries in Xilinx ISE Design Suite. Summary of results are shown in Table 1. Using proposed method shorter time to make prototype and better hardware in terms of cost, power, bandwith were achieved as well.

4. Summary

In this paper the method of rapid prototyping to reduce development cost of dedicated real-time systems was presented . Proposed method is based on developed DRCA architecture (with UCM as a main building block) and Hardware Petri Nets as automated modeling tool. The balance between processing elements (PE's), count of memories and interconnection network was achieved. Processing element based on bit- serial arithmetic (multiplication and addition) was also given. As an example

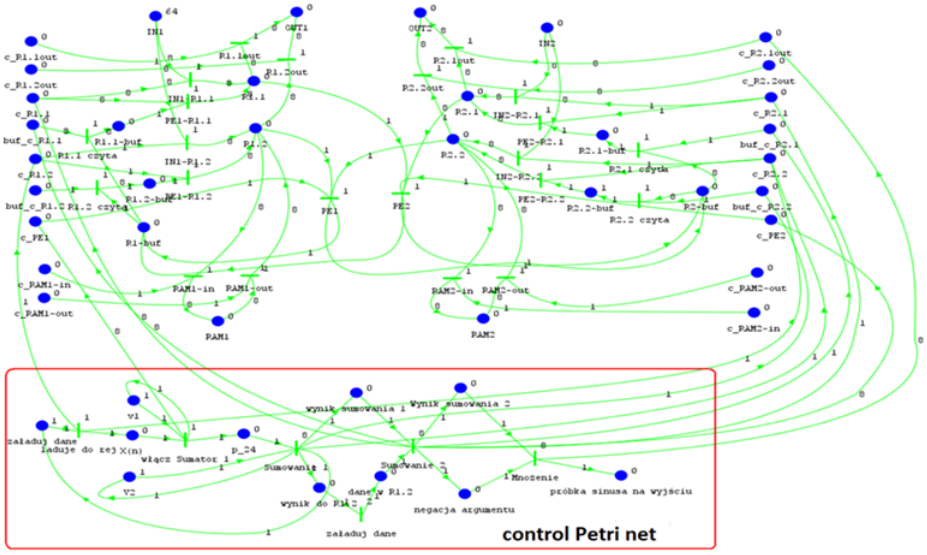
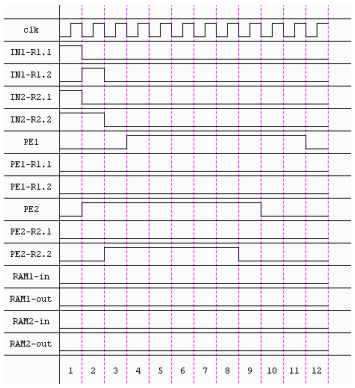


Fig. 10. Mapped algorithm of sin/cos generator



Synchronization diagram

Control signal

Fig. 11. Result of simulation using application based on HPN

Table 1. Summary of implementation TVDFT made on the DRCA and generally available solutions

	DRCA build on UCM		Standard HDL project	
Number of Slices:	34 out of 14336	0%	279 out of 14336	5%
Number of Slice Flip Flops:	33 out of 28672	0%	392 out of 28672	1%
Number of 4 input LUTs:	68 out of 28672	0%	1270 out of 28672	4%
Number of bonded IOBs:	197 out of 720	27%	197 out of 720	27%
Total estimated power consumption	410 mW		622mW	
Min. input arrival time before clock:	2.383 ns		2.383 ns	
Minimum period:	2.690 ns		20,302ns	
Max. Freq.:	372MHz		256MHz	
Max. output required time after clock:	3.847 ns		8.821 ns	
Maximum combinational path delay:	3.847 ns		3.847 ns	

of DRCA appliance the TVDFT algorithm was implemented. Presented in this paper rapid prototyping method is an solution suited to wide range of DSP algorithms.

References

[1] S. Zoran, Prototyping embedded DSP systems - from specification to implementation, In the Proc. EUSIPCO 2004, Vienna 2004, pp. 1625-1632

[2] A. A. Petrovsky, Mietody i mikroprocesnyje sredstwa obratotki sziroko polosnych i bistroprotiekajuszczych procesow w realnow wremieni (in russian) Nauka i Tiechnika, Minsk, 1988

[3] L. Wanhammar , DSP integrated circuits, Academic Press, USA, 1999.

[4] G.Rubin, M. Omieljanowicz, A. Petrovsky, Reconfigurable FPGA- based hardware accelerator for embedded DSP, MIXDES 2007, Ciechocinek, 2007, pp.147-151

[5] G. Rubin, A. Petrovsky, M. Omieljanowicz, Multilevel hardware Petri nets for rapid prototyping design platform. The 12th Intern. Conference Mixed design on integrated circuits and systems, MIXDES 2005, Kraków, Poland, 20-25 June 2005, pp.147-152.

[6] M. Adamski, M. Węgrzyn, Hierarchically Structured Colored Petri Net Specification and Validation of Concurent Controllers, Proc. In 39th Interenational Scientific Colloquium, IWK'94, Ilmenau, Germany, 1994, Band 1, pp. 517-522.

[7] G. Rubin, A. Petrovsky, M. Omieljanowicz, Rapid prototyping of Real time DSP-systems based on accurate simulation computation processes using Petri nets, WSFiZ Press, vol. I, Bialystok, 2005, pp. 409-417.

[8] A. Petrovsky, P. Zubrycki, A. Sawicki, Tonal and noise separation based on a pitch synchronous DFT analyzer as a speech coding method, The proc. of the ECCTD 03, vol.III, Cracow 2003, pp. 169-172.

[9] M. Omielianowicz, P. Zubrycki, A. Petrovsky, G. Rubin FPGA-based algorithms and hardware for generating digital sine wavesd, Mixed design of integrated circuits and systems : MIXDES'2002 : 9th International Conference, Wrocław June 20-22, 2002, pp. 279-284.

SZYBKIE PROTOTYPOWANIE DEDYKOWANYCH SYSTEMÓW W OPARCIU O ARCHITEKTURĘ WSPÓŁDZIELONEJ PAMIĘCI: METODA I PRZYKŁAD

Streszczenie: Celem tego artykułu jest zaprezentowanie metody szybkiego prototypowania redukującego koszty opracowania dedykowanych systemów obliczeniowych. Zaprezentowane rozwiązanie jest nakierowane na projektowanie systemów wbudowanych czasu rzeczywistego. W pierwszej części publikacji opisano zasady wyboru uniwersalnego modułu obliczeniowego i zaprezentowano uzyskaną architekturę przygotowaną do implementacji sprzętowej w układach FPGA. Określono też metodę modelowania za pomocą dedykowanych sieci Petri, nazwanych sprzętowymi sieciami Petri, wykorzystywanych w postaci oprogramowania typu CAD. To oprogramowanie pozwala na szybkie utworzenie modelu bloku obliczeniowego, następnie na przeprowadzenie automatycznej weryfikacji jego poprawności i ostatecznie wygenerowania wektorów sterujących pracą. Wykorzystując opracowane: uniwersalną architekturę modułu obliczeniowego i narzędzie typu CAD stworzono metodę pozwalającą na szybkie uzyskanie sprzętowego prototypu (na bazie podzespołów FPGA) układów obliczeniowych z zakresu cyfrowego przetwarzania sygnałów w czasie rzeczywistym. W drugiej części artykułu przedstawiono przykład wykorzystania zaproponowanej metody do zaprojektowania układu obliczeniowego realizującego przekształcenie TVDFT.

Słowa kluczowe: szybkie prototypowanie, architektura ze współdzieloną pamięcią, FPGA

Artykuł zrealizowano w ramach pracy badawczej S/WI/4/08.