ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 *2010. No. 3(99)*
ELEKTRONIKA IR ELEKTROTECHNIKA

AUTOMATION, ROBOTICS
T 125
AUTOMATIZAVIMAS, ROBOTECHNIKA

Hardware Accelerated FPGA Implementation of Lithuanian Isolated Word Recognition System

G. Tamulevičius
Recognition Processes Department, Institute of Mathematics and Informatics,
Akademijos str. 4, LT-08663 Vilnius, Lithuania, phone: +370 5 2660380, e-mail: g.tamulevicius@mch.mii.lt
V. Arminas, E. Ivanovas, D. Navakauskas
Department of Electronic Systems, Vilnius Gediminas Technical University,
Naugarduko str. 41-422, LT-03227 Vilnius, Lithuania, phone: +370 5 2744756, e-mail: dalius.navakauskas@el.vgtu.lt

Introduction

Speech is the most natural way of human communication. At least half of a century researchers and engineers are trying to automate speech recognition and synthesis. Yet there is no automatic speech recognition system equivalent to human recognition with its precision, robustness to noise and speaker, and versatility. Despite of difficulties of implementation it is agreed that speech is the future interface between computer and the human.

Although software implementations of speech analysis and synthesis systems are capable of real time operating, growing number of mobile devices implies the need for hardware implementation of speech processing systems. One technique of the potential speech recognition hardware implementations is Field Programmable Gate Array (FPGA). It is an array of semiconductor logical and memory elements interconnected by reprogrammable logical links. FPGA enables faster, less time consuming hardware implementation and finds a lot of applications in various fields: speech and signal processing [1–3], image processing [4], optimization [5] and networking [6].

FPGA implementations of speech recognizers are already reported in the literature. Implementations differ in various implementation levels: employed recognition approach, used analysis technique or vocabulary size. Hidden Markov Models [7, 8], Gaussian Mixture models [9, 10], Neural Network models [11], and Dynamic Time Warping techniques are used for speech recognition systems. Vocabulary size in reported implementations varies from ten words [11] to 20 000 words [10]. Some of recognizers are implemented on board software processor [12], some are partially built-in hardware.

Our earlier investigation revealed deficiencies of the softcore based recognition system. In the initial implementation of Lithuanian isolated word recognition system the recognition of one word took about 95 seconds. The main reason of prolonged speech analysis was dozens of successive calculations. Most of them were repeated multiple times, thus were inefficient especially considering limited software processor clock rate (100 MHz, in our case). These facts urge replacing mentioned software functions with hardware components and organize parallel analysis in order to speed-up recognition process.

This paper presents hardware accelerated FPGA implementation of Lithuanian isolated word recognition system. First we introduce to the used isolated word recognition technique, in short commenting on made solutions. Afterwards we present hardware implementation issues: chosen recognition steps for hardware implementation, used components and employed integration schemes. Then, we report on comparative results of implemented system versions, outlining recognition rates and efficiency of operation. Finally, we summarize achieved results and draw conclusions.

Isolated Word Recognition Technique

Pattern comparison technique called Dynamic Time Warping was used for isolated word recognition. The idea of pattern comparison is to extract particular features from unknown speech utterance and to compare with reference features (stored in a dictionary) in order to get most similar reference features. The label of predefined most similar reference features will correspond to the word that is recognized.

Generalized structure of Lithuanian isolated word recognition system [13] is given in Fig. 1. Let us shortly discuss main recognition steps.

1. *Start.* The recognition process is started by pressing particular button on the FPGA board. Then dictionary is loaded into DDRAM memory.

2. *Speech*. Separately recorded words are used for recognition. They are loaded into DDRAM via CF memory card in PCM file format (16 bit mono signal quality with 11.025 kHz sampling frequency).

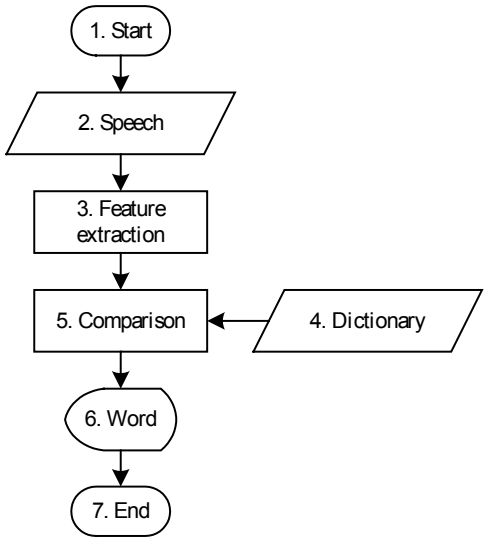

Fig. 1. Generalized structure of the recognition system

3. *Feature extraction*. Speech signal is framed firstly. Frames of 256 samples length (23.2 ms) with the frame shift of 128 samples (11.6 ms) are extracted.

After framing word endpoints are detected using energy threshold approach. Frame energy is calculated by:

$$E = \frac{1}{N}\sum_{i=0}^{N-1} x^2(i), \qquad (1)$$

here N – frame length; $x(i)$ – signal sample.

The extracted signal frames are windowed using Hanning window function.

Cepstral feature vector is extracted for each frame. 12th order cepstral analysis is applied for signal. Cepstral coefficients are calculated by:

$$c = \frac{1}{N}\left|FFT(\lg\left|FFT(x)\right|)\right|, \qquad (2)$$

here c – cepstral vector; x – signal frame vector; FFT – operator of fast Fourier transform.

4. *Dictionary*. Dictionary or reference feature sequences are designed outside the FPGA board. Dictionary is generated using dedicated PC software and saved in the form of binary file. This file is loaded to FPGA memory using CF memory card.

5. *Comparison*. The result of feature extraction step is sequence of cepstral feature vectors. The sequence is classified using Dynamic Time Warping approach based on Euclidean squared distance. Overall comparison requires following distances to calculate

$$D \approx \frac{1}{3}\sum_{i=1}^{M} T \cdot R_i, \qquad (3)$$

here D – number of distances between two vectors to calculate; T – number of frames of test utterance; R_i – number of frames of the i-th test utterance.

6. *Word.* The label of the assigned class as the recognized word is outputted to the LCD display located in FPGA board.

Hardware accelerated FPGA implementation

Xilinx ML402 board with Virtex-XC4VSX35 was used for implementation [14]. The development board contains 100 MHz clock rate MicroBlaze software (softcore) processor, 64 MB of DDR SDRAM, 1 MB of SRAM, 16 × 2 character LCD display, RS232 serial port, USB interface, Compact Flash connector, and JTAG configuration port.

Software implementation. The first implementation of speech recognizer was software based. Recognition system was implemented on MicroBlaze softcore processor. Experimental testing revealed low recognition process efficiency due to rational calculations used in the algorithm. An extended Floating Point Unit (FPU) was included in this implementation to increase the efficiency. The duration of the recognition process was decreased by 23 times and it took on average about 95 s to recognize the word. Further enhancements were selected in the form of integration of hardware components.

Following time sharing results of experimental testing of softcore based speech recognizer were obtained:

- Feature extraction step took about 38 % of overall recognition time. The reason was time-consuming calculation of FFT.
- Comparison process took approximately 61 % of overall time, because of repetition of calculations – comparison of utterances takes hundreds of distance calculation operations. It is a bottleneck of the system considering the vocabulary size of one hundred and more words.

Remark should be made on amount of analyzed data. The speech signal is redundant signal and use of high quality recordings can give unreasonable amount of data to analyze. Besides, large data amount sharpens the problem of time-consuming and repeating calculations.

Considering these observations following implementation points were chosen for optimization:

- *Time-consuming operations.* The whole feature extraction step was proposed for hardware implementation. This should give a remarkable acceleration of process.
- *Repeating monotonic calculation operations.* These include local distance and frame energy calculation. Calculation of local distance was selected as the most times repeated for hardware implementation.
- *Quality of recordings.* Defining the lowest quality level sufficient to ensure the initial recognition rate would reduce the data amount to analyze.

Hardware implementations. Softcore based implementation with FPU was selected for the acceleration. All hardware components were integrated into software implementation of speech recognizer. Integration is based on using VHDL wrapper code which implements communication of MicroBlaze processor and intellectual property (IP) core by the specified bus. The idea of VHDL wrapper is to create finite state machine integrating together other required components, allowing to synchronize and to control particular hardware units (see Fig. 2 on the next page).

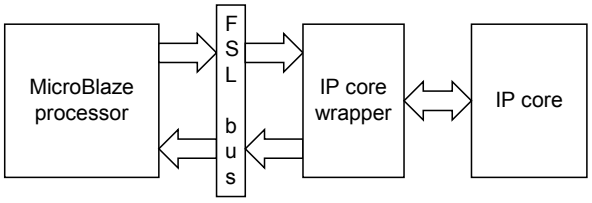

Fig. 2. Integration of hardware blocks

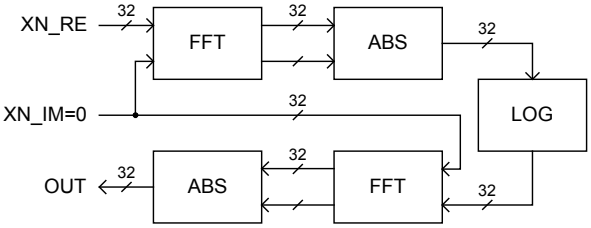

Fig. 3. Hardware implementation of feature extraction

Lithuanian isolated word recognizer implementation was enhanced in two stages.

Firstly, the feature extraction procedure outlined in equation (2) was implemented in hardware level. FFT IP core with burst input/output based on Radix-2 algorithm was selected for this task. Modulus and logarithm modules were used additionally. Fig. 3 shows the hardware implementation of the feature extraction (here buses of control signal are not shown).

The ABS module is implemented using multiplication, sum and square root units according to modulus calculating expression.

FFT IP core operates on 32-bit length values. They are inputted through XN_RE and XN_IM inputs. Only the real input was used in our case. Signal frame data are loaded into FFT core consequently, i. e., sample by sample through the buffer memory. The output is performed in the same manner from the OUT. Calculated feature vector is stored in DDRAM memory. New data is loaded when the unload operation is finished.

The finite state machine was created for FFT IP core (for diagram see Fig. 4). Following states were used in it: Initialization, Read_Inputs, End_Of_Read, Calculate_FFT, Write_To_Multipliers, Write_To_Adder, Write_Outputs, Write_To_Sqrt, End_Of_Write, Reset, Reset_State, Idle. These states are used to ensure continuous calculation process synchronized with data input and output.

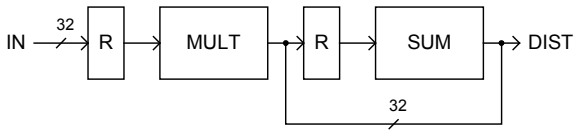

Fig. 5. Local distance calculation scheme

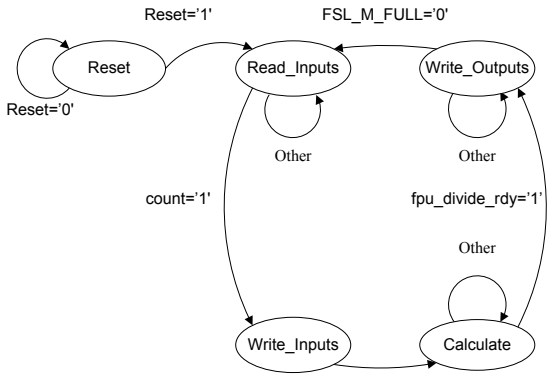

Fig. 6. Finite state machine for arithmetical units

Secondly, acceleration stage was implemented by using hardware accelerators for multiple times executed operations. Calculation of distance between vectors in comparison step was selected for the acceleration. According to equation (3) number of distances to calculate is proportional to dictionary size (number of reference utterances). The number of local distances can reach a few hundreds of thousands. It is a huge calculation load, thus distance calculation was selected for hardware implementation. One multiplication, subtraction and one adding unit were used for calculation of local distance between two feature vectors (see scheme in Fig. 5).

A difference of coordinates of compared vectors is inputted into multiplier (MULT) input IN using the register (R). Multiplier is used to get squared difference which is inputted to accumulator (SUM). The DIST gives the calculated distance between vectors.

A few hardware arithmetical units were used in different recognition steps additionally. These include one multiplier in feature extraction (used for multiplication by window function) and division unit in comparison step.

Fig. 6 gives the scheme of finite state machine created for arithmetical units (note, that the same scheme is used for all units).

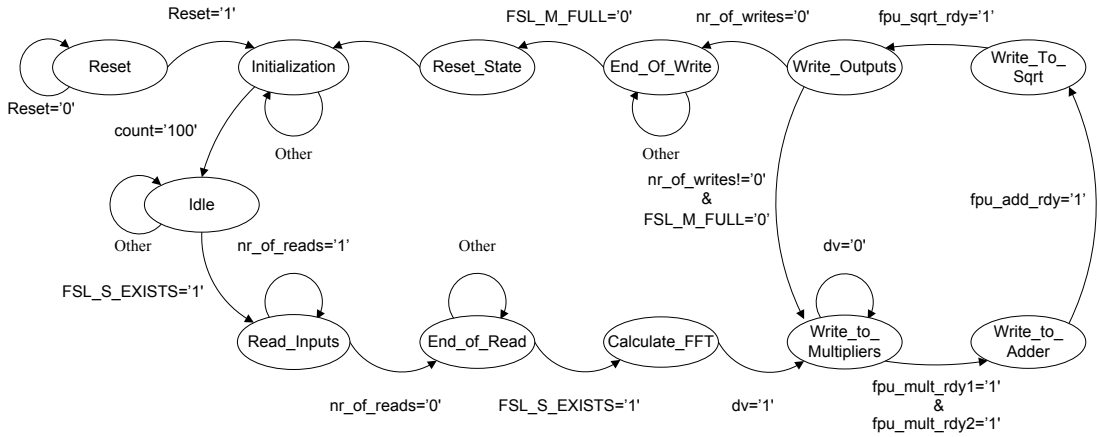

Fig. 4. Finite state machine for FFT IP core

Table 1. FPGA resources used for different implementations

FPGA resource	FPGA implementations					
	Alpha		Beta		Gamma	
	units	%	units	%	units	%
Slice registers	6 272	20	11 776	38	15 090	49
Slices	7 047	45	10 187	66	12 087	78
Inputs/outputs	178	39	178	39	178	39
4-input LUTs	7 382	24	11 912	38	14 074	45
FIFO16/RAMB16	37	19	46	23	46	23
DSP48 blocks	7	3	35	18	47	24

Used FPGA resources for three (cf. following section) implemented recognition systems are shown in Table 1.

Other possibility to speed up the recognition process is to decrease the amount of analyzed data. 11.025 kHz sampled records are typically used. If the sampling frequency will be reduced, e. g., to 6 kHz, then almost twice smaller amount of audio data will be needed to process. The potential threat here is a possible reduction of recognition rate. In order to determine influence of sampling frequency on recognition rate, experiments with different sampling frequency were performed.

Results of Experimentation

The implemented Lithuanian isolated word recognition system was tested in order to evaluate the recognition rate and the efficiency of different implementations.

Two utterances databases were used for testing. The first one consisted of 100 words (various nouns, pronouns, adjectives) pronounced 10 times by one male speaker. The database was intended for speaker dependent isolated word recognition. The speaker independent isolated word recognition was tested using second database. It contained 10 words (numbers from 0 to 9) pronounced 10 times by 10 speakers (5 females and 5 males). There were two versions of each database prepared: with sampling frequency of 11.025 kHz and 6 kHz.

All the experiments were performed using three FPGA implementation versions:

- *Alpha implementation* – based on MicroBlaze softcore processor with extended FPU module.
- *Beta implementation* – based on MicroBlaze softcore processor with extended FPU module and integrated FFT hardware module.
- *Gamma implementation* – based on MicroBlaze softcore processor with extended FPU module, integrated FFT hardware module and integrated arithmetical hardware units.

Experimental investigation of the recognition rates for different recognizer implementations. Experiments were performed both for speaker dependent (SD) and speaker independent (SI) recognition scenarios. The system was trained using first 6 sessions of records (600 words) and tested with the remaining 4 sessions (400 words). The experiments were performed using 11.025 kHz sampled records. Results are given in Fig. 7.

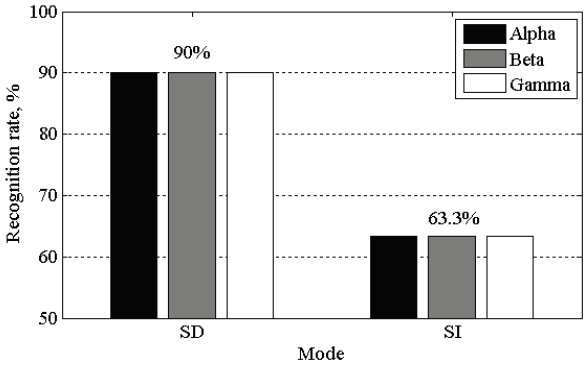

Fig. 7. Recognition rates for various implementations

The recognition rates for different recognizer implementations were identical. This shows identical performance of board software and hardware facilities.

Experimental investigation of the efficiency rates for different recognizer implementations. During experiments durations of recognition individual steps were measured. The speech input, endpoint detection, feature extraction and comparison steps were chosen for estimation as they are crucial for overall duration of recognition process. The duration was measured using first database (11.025 kHz sampling frequency) and for all implementations. The results of mean processing duration for single word are given in Table 2.

It can be seen that integration of FFT and hardware arithmetical units helped to speed up the feature extraction and decreased the signal analysis time.

Table 3 presents relative gain achieved in processing speed during individual recognition steps and in overall. A slight difference in duration of speech input (about 50 ms) between various implementations can be noticed. Possibly it can be caused by inconsistent file reading operation.

Table 2. Mean duration of recognition steps for considered implementations

Recognition step	Mean duration, ms		
	Alpha	Beta	Gamma
Speech input	1 224	1 275	1 274
Endpoint detection	235	235	235
Feature extraction	35 919	2 238	1 683
Comparison	57 956	57 952	67 050
Overall	95 334	61 700	70 242

Table 3. Relative gain in processing speed for recognition process

Recognition step	Relative to Alpha gain	
	Beta	Gamma
Speech input	0.96	0.96
Endpoint detection	1.00	1.00
Feature extraction	16.05	21.34
Comparison	1.00	0.86
Overall	1.55	1.36

Results show obvious supremacy of Beta implementation – based on the use of FFT hardware module. It outperforms the Microblaze (Alpha) implementation by 1.55 times and takes 61.7 s to recognize the word. Inclusion of hardware arithmetical units (Gamma implementation) also decreases recognition duration, i. e., by 1.36 times. However, that is lower than expected. It can be seen that Gamma implementation speed up feature extraction step but slackens the comparison process.

The explanation of this could be the different units used in both process steps. The feature extraction includes the multiplying unit and the comparison includes division, summation and multiplying units. Traditionally, division is a time consuming operation in digital signal processing. Direct replacement of the software function with hardware division could slower the process. Thus usage of hardware components should be concerned with organization of calculations in a parallel way.

Experimental investigation of the recognizer using different quality of records. Because of efficiency, implementation with the FFT hardware module (Beta) was selected as the target test system. The database of records with 6 kHz sampling frequency was additionally used for testing purposes. Training and testing conditions were identical to the first type of experiments. The experimental results are given in Fig. 8.

Results show advantage of using 6 kHz sampled records. In comparison with the case of 11.025 kHz sampled records recognition rate was improved by 2.8 % for speaker dependent recognition and by 4.2 % for speaker independent recognition.

Explanation for this fact could be the following. The energy of the human speech signal is concentrated in low frequency region (1–1.5 kHz) and the spectrum could be limited to 2–3 kHz area as all higher frequencies are more likely to be noise components rather than speech. This region could be sampled using 6 kHz frequency according to Nyquist law.

The efficiency of the system using 6 kHz records was tested. Operation durations of separate steps were measured and compared with the case of 11.025 kHz sampled records. Mean duration results for single word are given in Table 4.

Decrease of recording quality speeds up recognition process by 4 times (with no loss of recognition rate) and it takes almost 15.7 s to recognize the word. This is more than 6 times faster in comparison with 95 s required for software based recognizer implementation.

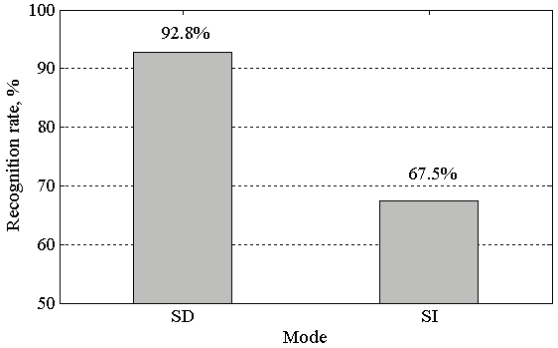

Fig. 8. Recognition rates of 6 kHz sampling frequency records

Table 4. Duration of recognition using different quality records

Recognition steps	Mean duration, ms		Relative gain
	11.025 kHz	6 kHz	
Speech input	1 275	736	1.73
Endpoint detection	235	128	1.84
Feature extraction	2 238	1 366	1.64
Comparison	57 952	13 444	4.31
Overall	61 700	15 674	3.94

Conclusions

1. More than 6 times FPGA implementation based on MicroBlaze softcore processor with extended FPU module of Lithuanian isolated word recognition system was accelerated. That yields final 15.7 s in average duration of recognition performed in Xilinx ML402 board with 100 MHz clock rate processor. Applied techniques that let to acceleration were:

- integration of the FFT IP core – accelerated overall recognition about 1.6 times;
- decrease of speech sampling frequency – additionally accelerated overall recognition about 3.9 times.

2. Additional integration of arithmetic hardware units into accelerated implementation of Lithuanian isolated word recognition system in overall downgraded results. It additionally accelerated feature extraction about 1.3 times, however prolonged comparison process about 1.2 times.

3. Accelerated FPGA implementation of Lithuanian isolated word recognition system slightly improved recognition rates: for speaker dependent recognition – by 2.8 % and for speaker independent recognition – by 4.2 %. Nevertheless, overall results of recognition rates were not the object of this study.

3. Experimental study of three FPGA implementations of Lithuanian isolated word recognition system revealed the fact, that most recognition time is spent in comparison step (e. g., 86 % in the case of 6 kHz records). Thus, comparison process should be further improved. Two possible improvement directions could be named:

- *Optimization of dictionary.* Reduction of the number of dictionary references would speed up the comparison process remarkably. Optimization should ensure non decreasing recognition rate.
- *Acceleration of comparison process itself.* Comparison time could be reduced by parallel organization of calculations and appropriate hardware implementation of comparison process.

Acknowledgements

This work was supported by Lithuanian State Science and Studies Foundation (No. T-09152, contract T-106/09).

References

1. **Yiu K.-F. C., Lu Y., Shi X., Luk W.** FPGA acceleration of a subband beamforming algorithm for speech enhancement

// Proc. of Congress on Image and Signal Processing. – 2008. – No. 5. – P. 742–746.
2. **Puidokas V., Marcinkevičius A.** Research on characteristics of audio DAC sigma-delta modulator on field programmable gate array // Electronics and Electrical Engineering. – Kaunas: Technologija, 2008. – No. 5(85). – P. 97–100.
3. **Amira A., Bouridane A., Milligan P. Roula M.** An FPGA implementation of Walsh–Hadamard transforms for signal processing // Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing Salt Lake City, UT. – 2001. – Vol. 2. – P. 1105–1108.
4. **Hocenski Ž., Aleksi I., Mijakovic R.** Ceramic tiles failure detection based on FPGA image processing // Proc. of IEEE Int. Symp. on Industrial Electronics ISIE 2009, Seolu. – 2009. – P. 2169–2174.
5. **Laptik R., Arminas V., Navakauskas D.** Ant system implementation using MicroBlaze: Some preliminary results on efficiency study // Electronics and Electrical Engineering. – 2009. – No. 6(94). – P. 27–30.
6. **Shinde S. A., Shelake V. G., Kamat R. K.** FPGA based packet splitter implementation using mixed design flow // Electronics and Electrical Engineering. – Kaunas: Technologija, 2008. – No. 8(88). – P. 15–18.
7. **Ke S., Hou Y., Huang Z., Li H.** A HMM speech recognition system based on FPGA // Proc. of Congress on Image and Signal Processing, Snya, China. – 2008. – P. 305–309.
8. **Gonzalez-Concejero C., Rodellar V., Alvarez-Marquina A., Martinez de Icaya E., Gomez-Vilda P.** Designing an independent speaker isolated speech recognition system on an FPGA // Research in Microelectronics and Electronics. – 2006. – P. 81–84.
9. **Lin E. C., Yu K., Rutenbar R. A., Chen T.** A 1000-word vocabulary, speaker-independent, continuous live-mode speech recognizer implemented in a single FPGA // Proc. of the ACM/SIGDA 15th Int. Symp. on Field Programmable Gate Arrays, Monterey, USA. – 2007. – P. 60–68.
10. **Miura K., Noguchi H., Kawaguchi H., Yoshimoto M.** A low memory bandwidth Gaussian Mixture Model (GMM) processor for 20,000-word real-time speech recognition on FPGA system // Proc. of the Int. Conf. on Field-Programmable Technology, Taipei, Taiwan. – 2008. – P. 341–344.
11. **Amudha V., Venkataramani B., Manikandan J.** FPGA implementation of isolated digit recognition system using modified back propagation algorithm // Proc. of the Int. Conf. on Electronic Design, Penang. – 2008. – P. 1–6.
12. **You K., Lim H., Sung W.** Architectural design and implementation of an FPGA softcore based speech recognition system // Proc. of the 6th Int. Workshop on System-On-Chip for Real-Time Applications, Cairo. – 2006. – P. 50–55.
13. **Tamulevičius G.** Pavienių žodžių atpažinimo sistemų kūrimas // Doctoral Dissertation, Vilnius Gediminas Technical University, Vilnius. – 2008. – 130 p.
14. **Xilinx**. MicroBlaze Soft Processor Core. Accessed at: http://www.xilinx.com/tools/feature/csi/microblaze.htm.

Received 2009 12 03

G. Tamulevičius, V. Arminas, E. Ivanovas, D. Navakauskas. Hardware Accelerated FPGA Implementation of Lithuanian Isolated Word Recognition System // Electronics and Electrical Engineering. – Kaunas: Technologija, 2010. – No. 3(99). – P. 57–62.

Enhancement of FPGA implementation of Lithuanian isolated word recognition system is presented. Software based recognizer implementation was used as the basis for enhancement. The feature extraction (as the most time required process) and local distance calculation (as the most times performed process) were selected for hardware implementation. Reduction of recording quality of speech was selected as the way to reduce the amount of the data to analyze. Experimental testing shows correctness of made solutions. Integration of Fast Fourier Transform module reduced the recognition time by 1.6 times, and lower quality of records increased the recognition rate by 2.8 % for speaker dependent and by 4.2 % for speaker independent recognition. The overall achieved acceleration is 6 times, average time of recognition of one word is 15.7 s. Ill. 8, bibl. 14. (in English; summaries in English, Russian and Lithuanian).

Г. Тамулявичюс, В. Арминас, Э. Ивановас, Д. Навакаускас. Ускорение распознавания слов литовского языка используя апаратные средства программи́руемой логической интегра́льной схе́мы // Электроника и электротехника. – Каунас: Технология, 2010. – № 3(99). – С. 57–62.

Рассматривается вопрос ускорения распознавания слов литовского языка реализованного в программируемой логической интегральной схеме. В качестве начальной версии для оптимизаций выбрана программная реализация распознователя. Для ускорения процесса распознования этапы анализа речи и расчета векторного расстояния реализованы аппаратными средствами. Для этого использован модуль БПФ, арифметические модули. В целях уменьшения объема анализируемых данных выбрано снижение частоты дискретизаций записи речи. Результаты экспериментов подтвердили правильность принятых решений. Внедрение модуля БПФ ускорила процесс распознавания в 1,6 раза. Сниженное качество ускорила процесс в 4 раза и повысило точность распознавания зависимого от говорящего на 2,8 %, а независимого – на 4,2 %. Общее достигнутое ускорение процесса распознования – 6 раз, достигнутое среднее время распознования слова – 15,7 с. Ил. 8, библ. 14 (на английском языке; рефераты на английском, русском и литовском яз.).

G. Tamulevičius, V. Arminas, E. Ivanovas, D. Navakauskas. Lietuvių kalbos pavienių žodžių atpažinimo sistemos įgyvendinimo LPLM spartinimas aparatinėmis priemonėmis // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2010. – Nr. 3(99). – P. 57–62.

Pristatomas lauku programuojamomis loginėmis matricomis įgyvendintos lietuvių kalbos pavienių žodžių atpažinimo sistemos tobulinimas. Tuo tikslu visiškai programinis atpažintuvo įgyvendinimas papildytas atskirų atpažinimo etapų aparatinėmis priemonėmis. Įgyvendinimui pasirinktas ilgiausiai trunkantis požymių išskyrimo etapas ir daugiausia kartų atliekamas atstumo tarp dviejų vektorių skaičiavimas. Duomenų kiekiui sumažinti pasirinkta prastesnė pavienių žodžių įrašų kokybė. Eksperimentai parodė, kad aparatinio greitosios Furjė transformacijos bloko naudojimas paspartino sistemos darbą 1,6 karto. Prastesnė įrašų kokybė paspartino atpažinimo procesą 4 kartus ir padidino nuo kalbėtojo priklausomo atpažinimo tikslumą 2,8 %, o nepriklausomo nuo kalbėtojo – 4,2 %. Bendras gautas atpažinimo paspartėjimas – 6 kartai, o vidutinė vieno žodžio atpažinimo trukmė – 15,7 s. Il. 8, bibl. 14. (anglų kalba; santraukos anglų, rusų ir lietuvių k.).