International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.6, No.5 (2013), pp.333-342
http://dx.doi.org/10.14257/ijsip.2013.6.5.29

Robust Automatic Speech recognition System Implemented in a Hybrid Design DSP-FPGA

Ali Aldahoud*, Hamza Atoui and Mohamed Fezari

**Al-Zaytoonah University of Jordan, Amman, Jordan*
Badji Mokhtar Annaba University
Faculty of Engineering, BP:12, Annaba, ALGERIA
aldahoud@zuj.edu.jo, Hamza.atoui@gmail.com,mohamed.fezari@uwe.ac.uk,

Abstract

The aim of this work is to reduce the burden task on the DSP processor by transferring a parallel computation part on a configurable circuits FPGA, in automatic speech recognition module design, signal pre-processing, feature selection and optimization, models construction and finally classification phase are necessary. LMS filter algorithm that contains more parallelism and more MACs (multiply and Accumulate) operations is implemented on FPGA Virtex 5 by Xilings , MFCCs features extraction and DTW(dynamic time wrapping) method is used as a classifier. Major contribution of this work are hybrid solution DSP and FPGA in real time speech recognition system design, the optimization of number of MAC-core within the FPGA this result is obtained by sharing MAC resources between two operation phases: computation of output filter and updating LMS filter coefficients. The paper also provides a hardware solution of the filter with detailed description of asynchronous interface of FPGA circuit and TMS320C6713-EMIF component. The results of simulation shows an improvement in time computation and by optimizing the implementation on the FPGA a gain in space consumption is obtained.

Keywords*: Configurable computing machines, FPGA-DSP hybrid Design, noise cancellation, LMS Filter Algorithm, speech recognition*

1. Introduction

Configurable computing machines (CCMs) bridge the gap between application specific integrated circuits (ASICs) and general-purpose microprocessors. They retain the flexibility of microprocessors while providing speed and power consumption more comparable to ASICs. CCMs represent a powerful alternative for certain applications, particularly communication systems. Speech processing in human-machine communication needs real-time processing; the utility of using faster and dedicated microprocessors is an obligation. Digital signal processors (DSPs) often lack the necessary speed to implement these algorithms, and additionally have higher power consumption—a significant drawback for embedded applications.

Field programmable gate arrays (FPGAs) are the flexible computational resource in most mainstream CCMs. FPGAs consist of simple computational units called combinational logic blocks (CLBs) linked by a configurable connective mesh. While FPGAs are extremely versatile, they have a significant drawback as a software radio platform: long reconfiguration times. To reduce the hardware needs for a given application—consequently reducing cost and power consumption—it is desirable that platforms support runtime reconfiguration.

ISSN: 2005-4254 IJSIP

The objective of this paper is to present a combination of DSP and FPGA on a design of an embedded system for automatic speech recognition application. LSM filter is implemented in two methods iterative and parallel, the cost of each method is computed using MODELSIM [1].

2. Related Works

The application to be implemented on the DSP-FPGA module the recognition of a set of isolated words within a noisy environment some related works are cited in [2] J. Manikandan and B. Venkataramani presented the possibility to implement a real-time ASR system using modified one against all SVM classifier and showed just the result based on different features selection. In [3] L. Bouafif *et al.*, discussed the implementation of digital speech processing algorithm on real time in a ADSP-21065L of Analog Device, they used as features Energy and zero crossing and pitch, results of simulation are presented in a table. A Fpga-based adaptive noise cancelling system is published by F. Wolfgang and M. Jorn in [4], they proposed a technique using two microphones in order to eliminate noise by adaptive filters, the tests were done over specific and periodic sounds like fan sounds, motor sounds and cars motor sounds, these sounds have time dependent statistical properties. In 2009 C. G Saracin *et al.*, published in UPB journal [5] how the LMS algorithm can be used for echo cancelation, the coefficients of the adaptive filter were updated in correlation with the transmitted and received data, the input of the simulator is a voice with echo and the output converges to a signal without echo. In [6] Amrita and R. Mehra from NITT India presented the implementation of an efficient noise canceller for digital receivers on an embedded design of type Microblase microcontroller, the proposed method has been simulated using Simulink and system Generator Blocs, synthesized with Xilinx Synthesis tool (XST) and implemented on Virtex4 based xc4vsx35-10ff668 and Spartan 3E based xc3s500e-4fg320 FPGA devices, a comparative speed study is done. In [7] V. Rodellar *et al.*, exposed in PL-2008 Conference a paper on implementation of adaptive noise canceller for robust speech enhancement interfaces using FPGA as a configurable computing device. J. Blomer and D. Rolkosky from university of Minnesota, demonstrated the application of adaptive filters implemented on the Spartan 3E FPGA architecture for active cancellation of audio frequency noise using a cancellation speaker and two microphone measurements: reference and error, two normalized least mean square (NLMS) filters are used : one for prediction and cancellation of the noise source and one used in a novel manner to compensate for a feedback channel from the cancellation speaker to the reference microphone, improvement on the work is done by reduction of power of input signal using the optimal Burg predictor [8].

3. General Presentation of Digital Filters

An adaptive filter is a filter that self-adjusts its transfer function according to an optimization algorithm driven by an error signal. Because of the complexity of the optimization algorithms, most adaptive filters are digital filters. By way of contrast, a non-adaptive filter has a static transfer function. Adaptive filters are required for some applications because some parameters of the desired processing operation (for instance, the locations of reflective surfaces in a reverberant space) are not known in advance. The adaptive filter uses feedback in the form of an error signal to refine its transfer function to match the changing parameters.

Generally speaking, the adaptive process involves the use of a cost function, which is a criterion for optimum performance of the filter, to feed an algorithm, which determines how to modify filter transfer function to minimize the cost on the next iteration.

For an adaptive algorithm that modifies the values of the coefficients of the filters, there are three factors that measure the efficiency of the algorithm: the complexity of calculus measurements and the amount of calculus executed at each step. The speed of adjustment allowing a fast converge of adaptive filter to the Weiner solution (FIR Filters), and the estimation error obtained from the difference between the present Weiner solution and solution given by the adaptive algorithm.

3.1. Adaptive Filter Applications

The main applications of these filters are adaptive cancellation of noise or echo.

In the configuration, the input signal x(n) and a noise source $N_1(n)$ are compared with a desired signal d(n), which consists of a signal s(n) distorted by another noise $N_0(n)$. The coefficients of the adaptive filter are adjusted to reduce the error e(n) to the optimal value which is zero.

Both noise signals for the configuration must not be related with the s(n) signal. Furthermore, the two noise signals should be related one to each other, which means they can come from the same source. The error signal will converge to zero as shown in Figure 1.

As the power of digital signal processors has increased, adaptive filters have become much more common and are now routinely used in devices such as mobile phones and other communication devices, camcorders and digital cameras, and medical monitoring equipment.

3.2. Least Mean Squares (LMS)

LMS algorithms are a class of adaptive filter used to mimic a desired filter by finding the filter coefficients that relate to producing the least mean squares of the error signal (difference between the desired and the actual signal). It is a stochastic gradient descent method in that the filter is only adapted based on the error at the current time. It was invented in 1960 by Stanford University professor Bernard Widrow and his first Ph.D. student, Ted Hoff. LMS algorithms are most used ones because it is simple to implement and it is stable, however it is weak in convergence and requires two inputs: a reference noise which is related to noise that contaminates the input signal and an error signal already calculated

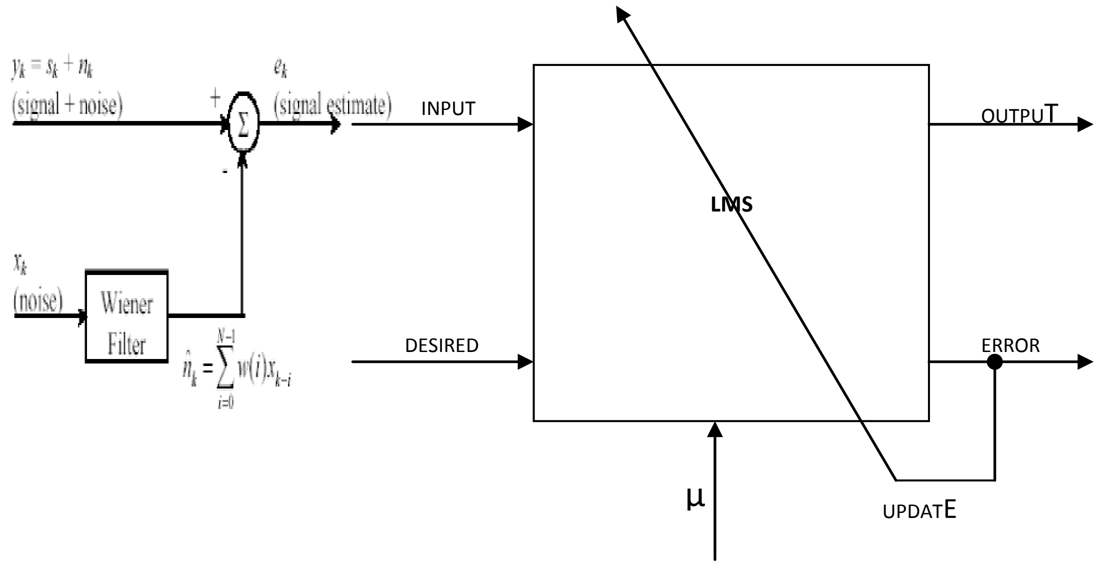

Figure 1. A Block Diagram of Adaptive Cancellation Noise Filter

LMS algorithm implementation

The filter is composed of two main parts: one to compute the filter output and the second is to update the coefficients

The first part is composed of a simple FIR filter with the following equation:

$$y(n)=\sum_{k=0}^{N-1} Coef(k)*x(n-k) \qquad (1)$$
$$N: number\_of\_coefs$$

The second part is the updating of coefficients:

$$\begin{aligned} &Coef(t_{n+1})=Coef(t_n)+\mu*err*x(t_n)\\ &err=output\_filter-desired\_value\\ &\mu: convergence\_factor \end{aligned} \qquad (2)$$

4. FPGA Implementation of LMS Core

There are three approaches to implement a system on FPGA. First approach is to adapt a hierarchical approach for the design of the system. The system can be split into number of sub modules and each module may be implemented either using the IP cores provided by the FPGA vendor or develop the modules on our own adapting the best architectures reported for minimizing the area, power dissipation and maximizing the speed. Pipelining and parallel processing may be used for this purpose. This has the best performance metric at the cost of comparatively larger development and debugging time. The second solution is to develop the system in C- code language and use a C-to-VHDL compiler [14] to implement the system on FPGA, the inconvenient with this approach are the lack to support floating point data types and a standard C-code will generate more syntax errors with these compilers, as each compilers has its own syntax for writing the c-code. Besides, there is no guaranty that the C-compiler will be efficient in mapping the design to FPGA with regard to speed and area. The third approach is to use system-on-programmable chip (SOPC) wherein soft CPU such as MicroBlaze and Cortex-M1 [15-16] are implemented on F¨GA device. It supports programming using high level language like C++ and a complex system can be easily designed and tested, which makes it convenient approach for most of FPGA based implementation, it support floating point calculus. The processor with custom instruction combines the best of both worlds; it gives the flexibility of the soft-core processors and high performance of a full custom design. However, most of soft-core processors are license properties. In our design the first solution is taken as a first solution in order to optimize the computation by pipelining phases and exploring to the maximum the parallel computations.

4.1. LMS Core Details Design

To optimize LUT consumption, we decided to share the resources in order to use less MAC unites within Virtex 5. The MAC unites are used in the mechanism of computing the output of the LMS filter and the same MAC unites are sued in updating coefficients calculus, Figure 2.a and 2.b shows the two sequential phases that can be done in pipelining mode with the same MAC unites.

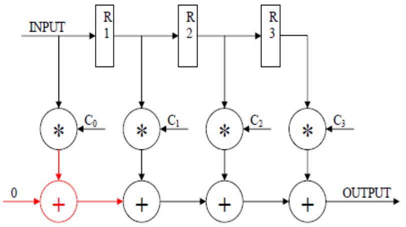

Figure 2.a. Output LMS Filter Computation Phase

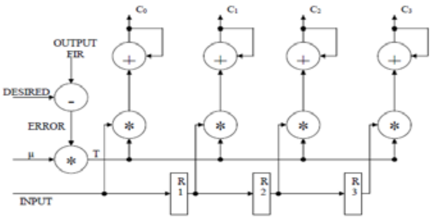

Figure 2.b. Updating Filter Coefficients Phase

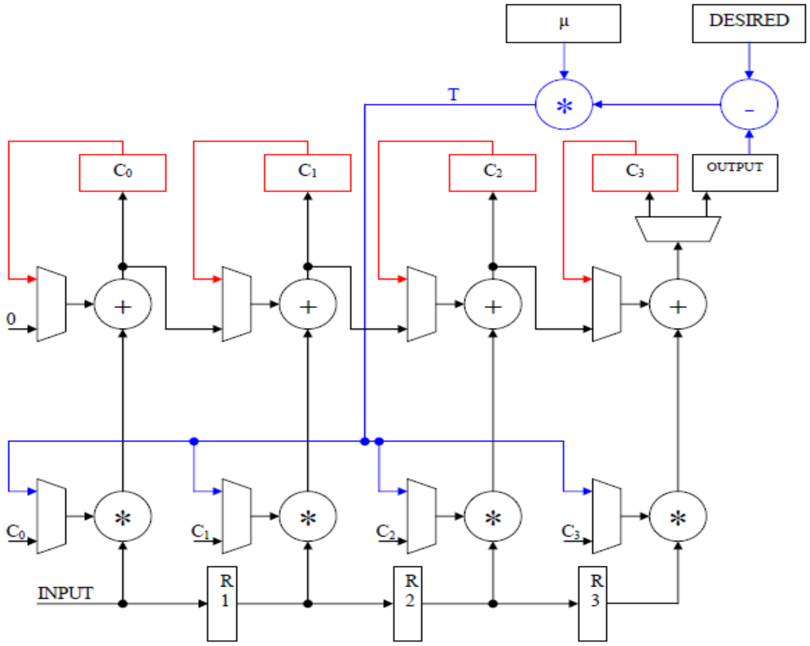

Figure 2.c. LMS by Sharing MAC Resources

4.2. Hardware Description of the LMS Filter

The hardware design is based on LMS-core, delay unit, control and status unit and a set of registers, each element has a role within the design presented in Figure 3.

Input register: is a 32 bits resister coded in IEEE-754 SP format used to load the sample.
MU register: has the same format as Input register used to load error adaptation step.
CSR register: is a 32 bits register used to control the system, in which: the bit B0 (LSB) called Go_Done is used to satr computation by setting it to one, then the system erases this bit to signify the end of computation. The bit B1 called SOFTRST is used to create a soft reset to the system by setting it to one; it will be erased automatically by the system.

Output register a 32 bits register in the same format as the input register, it contains the results of the LMS core.

The control and status unit contains the address decoder unit and the flow state machine; it is related and controlled by some external signals such as: RST, Clock, ADDR, CE, RD and WR.

4.3. The FSM (Finite State Machine) of the Control Path

The finite state machine in the control and status unit is shown in Figure 4 and describes a sequence which is with new input sample, the FSM composed of five states to compute the output of the LMS filter and the soft reset.

The states are: Idle: for idle state, OUTP for updating the output register, MCOEF for updating filter coefficients, SHFTD for shifting the input and coefficients and SOFTRAS is the state to restart the internal registers of the filter as shown in Figure 4.

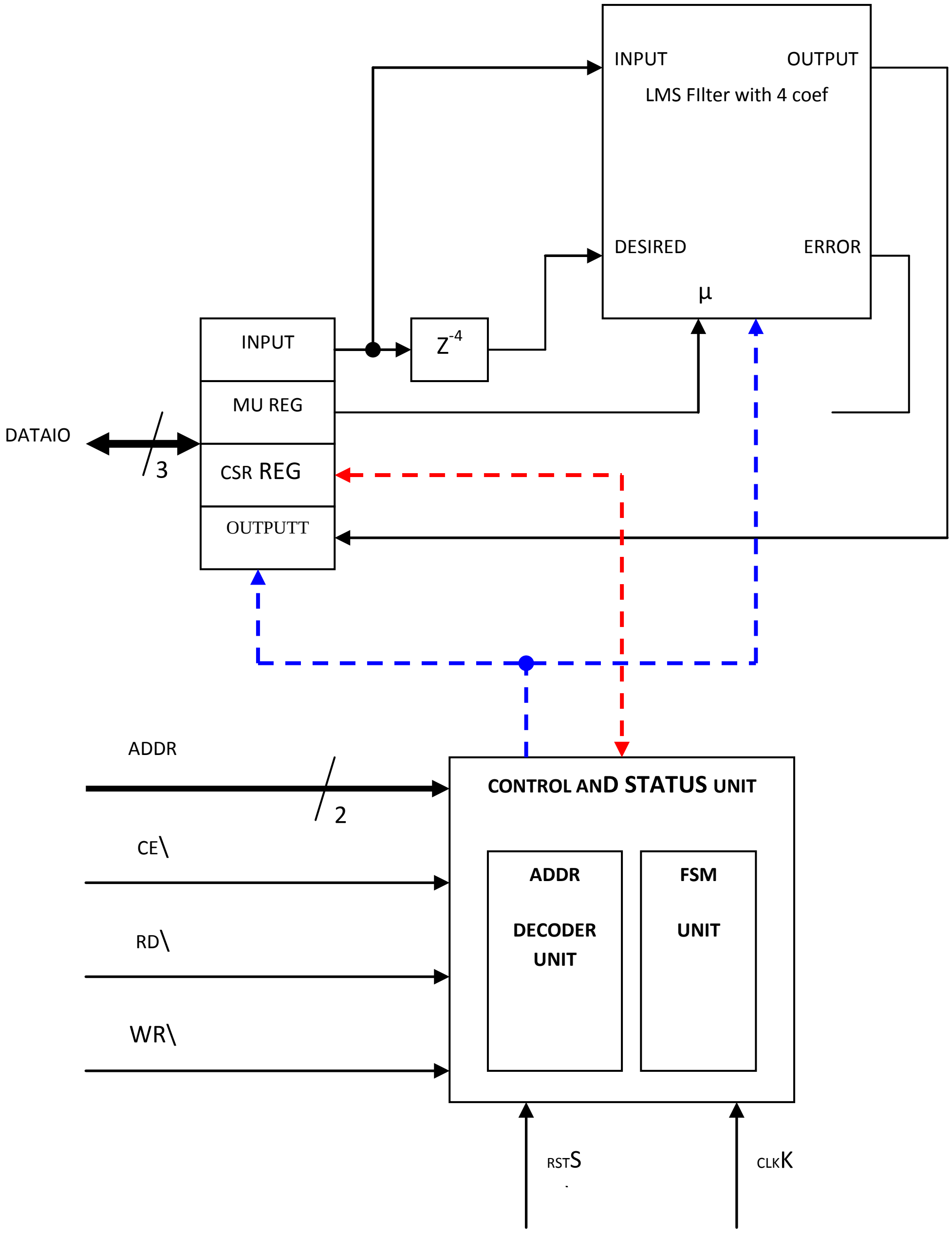

Figure 3. LMS Hardware Units Design

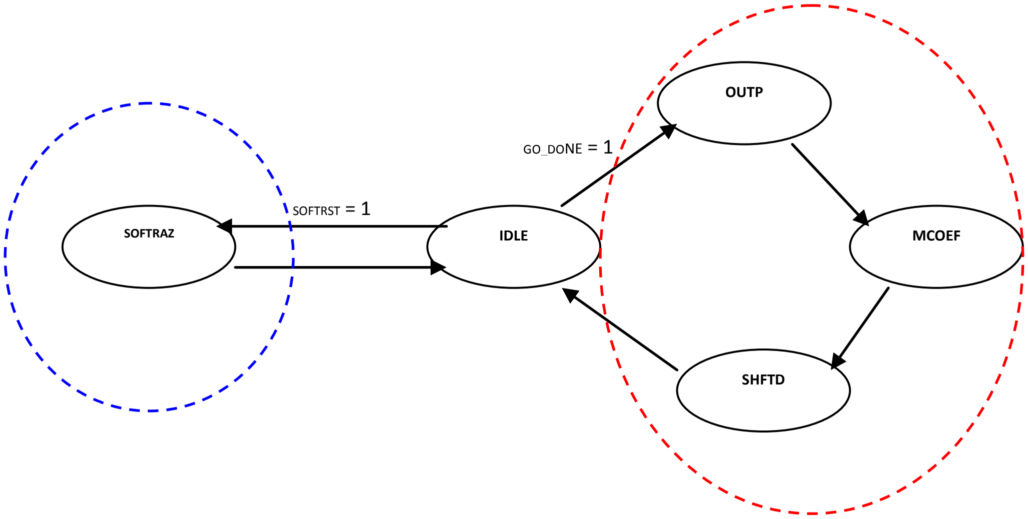

Figure 4. Diagram of the FSM of the LMS Filter

5. Implementation of the Speech Processing as a Case of Application

The speech recognition system designed on a hybrid DSP-FPGA module includes the following steps:

A real time speech acquisition via microphone input connected to the DSP card TMS320C6711, the samples got from the CODEC are transmitted to the FPGA module where the signal is filtered using the LMS algorithm, the filtered samples are retransmitted again to the DSP card via EMIF interface, an automatic segmentation by sliding hamming window of 20 ms and 30% of recovering ratio. Then a computing acoustic and speech parameters such as 12-PLCC (predictive linear spectral coefficients), pitch and two formants are used as features, models were constructed for four Arabic words (Amame, wora, yamine , yassar).four modeles were constructed and values were conserved within the DSP card, DTW was simulated using Matlab, and then implemented using CCS (Code Composer Studio) on the DSP TMS320C6711.

In general, DTW is a method that allows a computer to find an optimal match between two given sequences (*e.g.*, time series) with certain restrictions. The sequences are "warped" non-linearly in the time dimension to determine a measure of their similarity independent of certain non-linear variations in the time dimension. This sequence alignment method is often used in t this example illustrates the implementation of dynamic time warping when the two sequences are strings of discrete symbols. d(x, y) is a distance between symbols, *i.e.*, d(x, y) = | x – y |.

We have implemented the LMS filter core with Virtex 5 Model XC5VLX330T-2FF1738 FPGA [17-18], which supports embedded memory and DSP48Es, the developed chip was tested for 8 SNR ratios, to calculate signal-to-noise ratios accurately for this testing phase, we used artificially mixed signals with known computed white noise added to real time generated words he results are shown in Figure 5. Another test was done on the computation of space and components consumption for the LMS core and presented on Table 1.

Table 1. Device Utilization Summary (Estimated Values)

Logic Utilization	Used	Available	Utilization
Number of Slice Registers	458	207360	0%
Number of Slice LUTs	17345	207360	8%
Number of fully used Bit Slices	208	17595	1%
Number of bonded IOBs	39	960	4%
Number of BUFG/BUFGCTRLs	2	32	6%
Number of DSP48Es	10	192	5%

5.1. Experimental Results

We have made some test on the DTW algorithm implemented on DSP processor, the conditions of test are: 4 Arabic words with the meaning(forward, backward, left and right), 50 occurrences for each word, and different SNR used SNR = {0, 5, 10, 15, 20, 25, 30, clean } dB. The DSP processor transmit samples to LMS core, once filtered the samples are retransmitted to the DSP to compute and extract Features and then classify the word based on DTW technique.

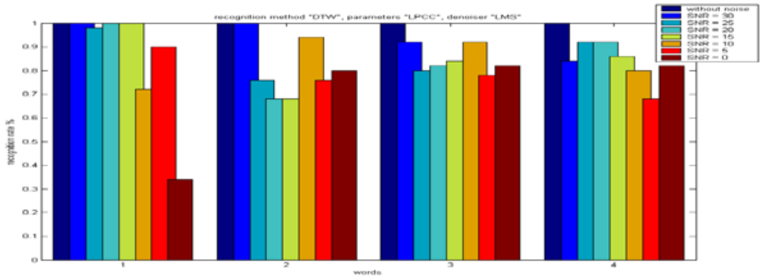

Figure 5. DTW Implemented on DSP Results with Different SNR Ratios

5.3. Comparative Study with other Commercial Speech Recognition Processors

VR-stamp from sensory has been tested in some application, it is not robust in a noisy environment, Sensory's VR Stamp™ simplifies the design of speech recognition products by integrating all key components into a convenient 40-pin DIP footprint module. A low-noise audio channel and standardized footprint allow rapid prototyping, less debugging and shorter time to market. Drawback of this processor is the lack of pre-processing phase for adaptive noise cancellation and response time is considerable (1 to 2 seconds) [19].

The RSC-4128 represents Sensory's next generation speech and analog I/O mixed signal processor. The RSC-4128 is designed to bring high performance speech I/O features to cost sensitive embedded and consumer products. Based on an 8-bit microcontroller, the RSC-4128 integrates speech optimized digital and analog processing blocks into a single chip solution capable of accurate speech recognition; high quality, low data-rate compressed speech. However it has a poor recognition rate in outdoor tests

The HM 2007 has an inbuilt hardwired Artificial Neural Network system. For each time the user says the word, the HM2007 integrates this word into a neural network. Later, in recognition mode, the HM2007 tries to match the spoken word against other words in its neural net. If a match is made, the index of that word in the vocabulary is returned. If no match is found, or if the user spoke too quickly or too slowly, an appropriate error code is returned. This processor is limited in vocabulary words and not robust to noisy environment.

The proposed design, integrates a powerful denoising algorithm, it can be used for large vocabulary and since it is implemented on FPGA it is fasted in response.

6. Conclusion

In this work, we presented the possibility to implement a speech recognition system based on an embedded FPGA-DSP system, the results of integrating LMS core in an FPGA module is very interesting, TMS320C6711 enables to design a system with very high computational power and large memory space with minimal count o components on the circuit board space and then simplifies design phase, these processors are very suitable for speech processing, the designed module can be easily installed to control the navigation of a mobile robot. Based on Table 1 it is obvious that very little unites were used to implement the LMS core, the remaining of these unites can be used later to implement the desired classifier DTW directly on the FPGA module and eliminate completely the DSP processor in order to accelerate more the process of recognition.

Acknowledgments

We are grateful the Dr. Mokhtar Nibouche from Department of Electrical and Computer Engineering, University of the West of England Bristol UK for his support during training days on DSP boards.

References

[1] ModelSim Advanced verification and Debugging, Xilinx tutorial ver 6.0 published seprt, **(2004)**.

[2] J. Manikandan and B. Venkataramani, “Design of a real time automatic speech recognition system using Modified One agains All SMV classifier”, ScienceDiret journal of Microprocessor and Microsystems, www.elsevier.com/locate/micpro, vol. 35, **(2011)**, pp. 568-578.

[3] L. Bouafif, K. Ouni and N. Ellouze, “A real time implementation of a digital speech processing algorithm under DSP -21065”, Int. Jouranl of Research and Review in Computer Science, vol. 2, no. 6, **(2011)**.

[4] W. Fohl, J. Matthies and B. Schwarz, “A FPGA-Based adaptive Noise Cancelling System”, Proc. of the 12th Conf. on Digital Audio effects (DAFs), Italy, **(2009)** September.

[5] C. G. Saracin, M. Saracin, M. Dascalu and A. M. Lepar, “Echo Cancellation using the LMS Algorithm”, U.P.B. Sci. Bu. Series C., vol. 71, no. 4, **(2009)**.

[6] R. H. Amrita, “Embedded design og an efficient Noise Canceller for Digital Receivers”, IJEST, vol. 3, no. 2, **(2011)** February, pp. 1252-1257.

[7] V. Rodellar, “FPGA implementation of an Adaptive Noise Canceller for robust Speech Enhancement Interfaces”, Programmable Logic, **(2008)**, pp. 13-18.

[8] SudhanshuBaghel and RafiahamedShaik, “FPGA Implementation of Fast Block LMS Adaptive Filter Using Distributed Arithmetic for High Throughput”, IEEE, **(2011)**, pp. 443-447.

[9] A. Elhossini, S. Areibi and R. Dony, “An FPGA Implementation of the LMS Adaptive Filter for Audio Processing”, IEEE International Conference on Reconfigurable Computing and FPGA’s, ReConFig 2006, **(2006)**, pp. 1-8.

[10] T. Lan and J. Zhang, “FPGA Implementation of an Adaptive Noise Canceller”, Information Processing (ISIP), 2008 International Symposiums, **(2008)** May, pp. 553-558.

[11] A. Di Stefano, A. Scaglione and C. Giaconia, “Efficient FPGA Implementation of an Adaptive Noise Canceller”, Computer Architecture for Machine Perception, 2005. CAMP 2005. Proceedings. Seventh International Workshop, **(2005)** July, pp. 87-89.

[12] U. Meyer-Baese, “Digital Signal Processing with Field Programmable Gate Arrays”, Springer, **(2004)**.

[13] W. C. Chew and B. Farhang-Boroujeny, “FPGA implementation of acoustic echo cancelling”, Proceedings of the IEEE Region 10 Conference 1999 (TENCON’99), vol. 1, **(1999)** September 15-17, pp. 263-266.

[14] Handel-C languagereference manual, duoc number: Rm-1003-4.4, **(2007)**.

[15] Nios II processor Refenrence handbook N115V1-7-2, Altera Corporation, **(2008)** May.

[16] MicroBlaze Development Kit Tutorial Xilinx Inc. **(2002)**.

[17] http://www.xilinx.com/support/documentation/virtex-5_board_and_kit_documentation.htm.

[18] http://www.xilinx.com/support/documentation/virtex-5.htm.

[19] M. Fezari, “VR-Stamp with DSP-TMS320C6711 for hand-free voice-driven monitoring robots navigation”, Proceedings IEEEXPLORE of Conference ICITeS’12 at Hammamet Tunisia Marsh, **(2012)**.

Authors

Ali Aldahoud is an IEEE senior of Al-Zaytoonah University of Jordan, Amman-Jordan Dean of the Faculty of Science and IT, has many publications in different journals and participated in many international conferences. aldahoud@zuj.edu.jo

Hamza Atoui is a PhD student at BAdji mokhtar annaba University faculty of engineering dept Elerctronics, has participated in many international conferences, his main interests are: speech processing and DSp/ FPGA algorithms implementation.

Mohamed FEZARI is an Associate professor at Badji Mokhtar Annaba University Faculty of Engineering Dept: Electronics, he has many publications in DSP, he has participated in many International Conferences, his main interest are: speech processing, human machine / robot interaction and WSN