

Design of Hybrid CPU-FPGA Processor For High Performance Applications .

V.JeanShilpa

Assistant Professor

Department of ECE

B.S.Abdur Crescent Institute of
science & technology,
Vandalur, Chennai,INDIA

DrP.K.Jawahar

Professor & Head

Department of EIE

B.S.Abdur Crescent Institute of science
& technology,
Vandalur, Chennai,INDIA

Karthik S

Assistant Professor

Department of ECE

SRM University,
Vadapalani,
Chennai,INDIA

Abstract—Multicore processors are heart of every computing system, because the computing ability is not restricted to one area. Especially heterogeneous multicore designs serve the purpose to integrated processors with different specification. The designs that run on these processors demand regular up gradation, bit expandable, integrated high speed buses, memory expansion etc. To solve this major issue this multicore processor design on FPGA tries to integrate an FPGA with a dual cortex A9 processor to integrate and run parallel to need the technological barriers. To investigate the performance on the developed processor, two applications were developed in matlab and Simulink ie, the least mean square adaptive filter and hybrid window based FIR filter. The applications are profiled in vivado. The power consumed by porting the application is 1.542W for dynamic and 0.158W for static conditions.

Keywords—FPGA, Adaptive least mean square filter, hybrid window based FIR filter.

I. INTRODUCTION

Multicore design have become flexible with the advent of FPGA boards that embed an ARM cortex dual A9 processor. These boards are termed as embedded boards with hardcore and softcoreprocessors[1]. The need for multicore processor boards have emerged in market because application in current trends demand high computing power which a single processor is unable to processor. Previously applications either demanded high speed or less area or low power. Current applications demand all the three parameters to satisfy. Hence the need for processors which solve these issues are interfaced together to

meet the demands and produce designs for high performance [2]. There is a search for platform which will solve issues related to current updation of the applications, meet the size of the chip.

FPGA are emerging as a platform processor provides a right platform for a portable, reconfigurable, low power and low cost device designs. Most of the applications demanding high performance are getting shifted towards heterogenous multicore designs to grab maximum advantage of different cores to get the over all performance [3].

Application in signal filtering demand platform that are flexible. Signal filtering applications need platform that are adaptable or reconfigurable based on the intensity of noise signals. This papers aims at designing two filters designed in matlab known as adaptive least mean square algorithm and hybrid window based FIR filter. The adaptive least mean square algorithm are developed because these filters aim at filtering only noise signals of constant frequency. If the frequency of the noise signal changes only adaptive filters can reconfigure to meet the demands and produce a smooth noise free signal. This process becomes feasible by smartly changing the coefficients values in the filtering process[4].

II. Design of data intensive applications on multicore processors.

Signal are always corrupted by randomly changing noise signal. Fixed co-efficients filters are static filters which do not change with changing random signals. Hence, filter structures are designed to be adaptive. Least mean square adaptive filter being the most efficient one that adapts filter coefficients

relative to the strength of the noise signal. The most commonly and efficiently used filter being the LMS filter that adapts the coefficients of the filter relative to the strength of the noise signal. Adaptable co-efficient are based on the adaptive signal to noise ratio of the noise corrupted signal. Adaptive filters have already been designed and implemented on various platforms. But in this paper the important and new method identified are sharing of task between two processors. One being the hardcore dual cortex A9 ARM processor and secondly is the softcore FPGA gate array processor. The process followed to share the tasks between both the processors is, the design intended for signal processing is first designed in matlab or Simulink. If the design involves standard signal processing blocks, it can be designed in Simulink. Were as if the design is Custom made by the designer, it can be designed in matlab. When the design process is complete, the design is Pre-tested to estimate the most time consuming block in G-profiler of matlab and post-testing is carried out in Gprofiler of Xilinx vivado tool.

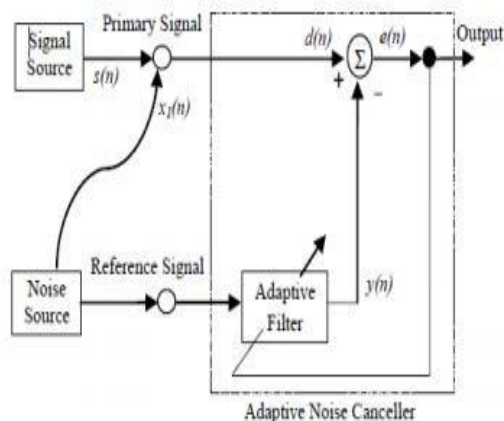


Figure 1. Adaptive Least mean square filter.

Adaptive filters are fed with two inputs, $z(n)$ being the primary input signal, which refers to the corrupted noisy signal, and find out SNR of the signal always the reference signal is taken for comparison. Also the signal's noise signal is considered in finding the error content like the white noise or the gaussian noise. The representation used in Figure 1 are listed, $s(n)$ – source input signal, $z(n)$ -primary signal, $x(n)$ -

noise reference input, $e(n)$ -system output signal, $y(n)$ -output of adaptive filter, $w(n)$ -noise signal, As the figure it self shows this process is cyclic by the feedback loop. The process of noise filtering is carried using continuous feedback system[5].

The least mean square adaptive filter is designed in Simulink with sine wave combined with tonal noise. The audio samples are ported with tonal noise. HDL coder converts this design verilog code. Two Workspace blocks are also present to allow the LMS output and the corrupt audio signals to be output to the MATLAB workspace for audio playback (Figure 2).

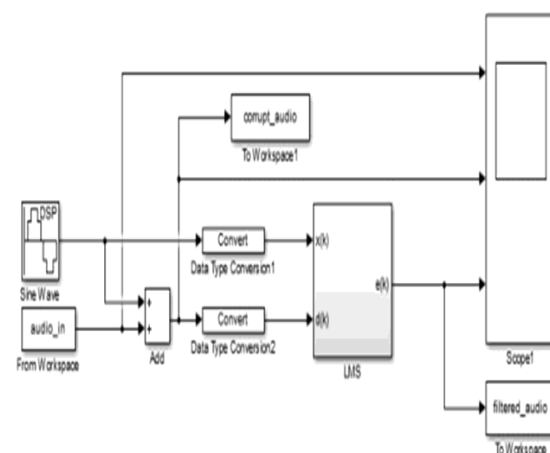


Figure 2. Simulink model of LMS adaptive filter.

Profiling was carried out for the designed Simulink block which describes the following performance functions. Profiling tool present in MATLAB computes execution time for every function present in the C program. In profiling, the number of times each and every function is called is computed. Also the time is computed for the execution of every function. Profiling is a for

mof program analysis that is used to aid the optimization of software; the parameters checked in profiling are calls, time and time/call. The call parameter gives the number of times this function was called or invoked, secondly the time parameter gives the total time spent executing all function invocations as an absolute value and as a percentage of the total simulation time and finally, Time/Call gives the average time required for each invocation of this function, including the time spent in functions invoked by this function. The report generated shows the time/call taken by the LMS function exceeds all other functions. Hence, it is decided to take this block alone to the FPGA part in the programmable

Function List					
Name	Time	Calls	Time/call	Self time	Location (must use MATLAB Web)
simulate(lms)	311.811607150	102.0%	1	311.81160715000000	0.00000000 lms
simulationPhase	310.807091950	100.0%	1	310.80709195000000	20.86573290 lms
lms_Outputs_Major	263.87581650	90.8%	441001	0.00064257450000	126.22320270 40.1% lms
lms/LMS (AtomicSubsystem_Outputs_Major)	41.30708940	13.3%	441001	0.000046481101	16.47570510 8.5% lms/LMS
lms/Scope1 (Scope_Outputs_Major)	16.36570510	5.3%	441001	0.000037178351	16.36570510 5.3% lms/Scope1
lms/Data Type Conversion2 (DataTypeConversion_Outputs_Major)	15.86950160	5.1%	441001	0.0000350454703	15.86950160 5.1% lms/Data Type Conversion2
lms/To Workspace (ToWorkspace_Outputs_Major)	15.61570010	5.0%	441001	0.00003543667050	15.61570010 5.0% lms/To Workspace
lms/LMS/LMS Filter (S-Function_Outputs_Major)	15.36726870	4.9%	441001	0.00003491440000	15.36726870 4.9% lms/LMS/LMS Filter
lms/Data Type Conversion1 (DataTypeConversion_Outputs_Major)	14.32086180	4.6%	441001	0.00003247380361	14.32086180 4.6% lms/Data Type Conversion1
lms/To Workspace1 (ToWorkspace_Outputs_Major)	14.24269130	4.6%	441001	0.00003226873240	14.24269130 4.6% lms/To Workspace1
lms/Add (Sum_Outputs_Major)	14.21109110	4.6%	441001	0.0000322686364	14.21109110 4.6% lms/Add
lms/From Workspace (FromWorkspace_Outputs_Major)	13.27588510	4.3%	441001	0.00003012032057	13.27588510 4.3% lms/From Workspace
lms/Sign Wave (S-Function_Outputs_Major)	12.32407900	4.0%	441001	0.00002794964501	12.32407900 4.0% lms/Sign Wave
lms/Update	8.62644440	2.3%	441001	0.000019370816751	8.62644440 2.3% lms
compilationPhase	0.54000000	0.2%	1	0.54000000000000	0.54000000 0.2% lms
terminationPhase	0.00000000	0.0%	1	0.00000000000000	0.00000000 0.0% lms
lms/Terminate	0.04000000	0.0%	1	0.04000000000000	0.01000010 0.0% lms
lms/Scope1 (Scope_Terminate)	0.03120000	0.0%	1	0.03120000000000	0.03120000 0.0% lms/Scope1
initializationPhase	0.01500010	0.0%	1	0.01500010000000	0.01500010 0.0% lms
lms/To Workspace1 (ToWorkspace_Terminate)	0.00000000	0.0%	1	0.00000000000000	0.00000000 0.0% lms/To Workspace1
lms/To Workspace (ToWorkspace_Terminate)	0.00000000	0.0%	1	0.00000000000000	0.00000000 0.0% lms/To Workspace
lms/LMS (AtomicSubsystem_Terminate)	0.00000000	0.0%	1	0.00000000000000	0.00000000 0.0% lms/LMS

Figure 3: Profiling report of Simulink LMS model.

III. Design of hybrid window based FIR filter.

Hybrid window based FIR filter designs an adaptive window function generating co-efficients for FIR filter that are adaptive with the input high frequency noise signals to suppress accordingly. The adaptable parameters in this CAT window takes the feedback of the current input signals frequency and amplitude there by adapts its parameters to generate new co-efficients to suppress the high frequency noise signal. The combined advantages of traditional Hamming and Hann window are employed to give a reduced main lobe width, where maximum energy is concentrated and to achieve faster side lobe roll off [6]. To design the hybrid window is to take a product function of both the window functions and maintain the α value flexible fig 4. For signals deservng maximum energy the desirable value of α is 5 to 6. The device used in the experiment is the

logic portion of FPGA. and. The entire design setup is a developed flexible heterogeneous multicore processor for CAT window. The clock, reset inputs are synchronous to the blocks.

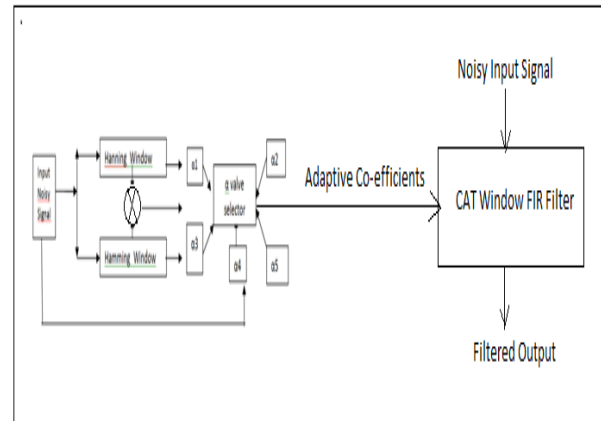


Fig4 : Multicore Architecture of FIR filter with CAT algorithm

ZC702 Zynq ,all purpose platform. This board is a solution for connecting hard processor core with a dedicated silicon area mounted on the board along with the softcore programmable logic core on FPGA . This board is equipped with ARM cortex dual A9 processor forming the hard processor and the programmable logic soft core processor. The hard core processor and the softcore processor communicate through the AXI bus. The purpose of this setup is that, the hardcore processor is stringent in adapting to new changes in the application, nor it allows adding a new processor core to FPGA for providing more processing capabilities. The device is controlled by the FPGA clock running at 100Mhz.

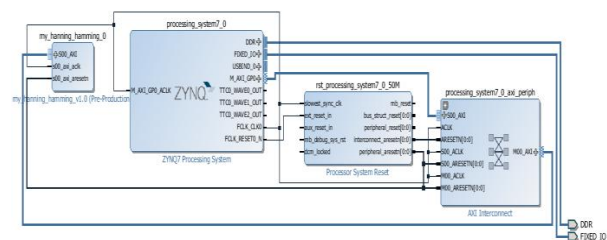


Fig5: Block diagram representation of Hybrid window FIR filter.

Table 1 and fig 25 clearly indicates that the hardcore utilizes more percentage of the hardware resources like LUT's, slice registers and BIO(Bounded input output) pads.

Resources	Core1(ARM Processor)	Core2(FPGA)
-----------	----------------------	-------------

LUT	452	56
Slice Registers	607	137
Bounded IO pads	96	34
Buffers	1	0
DSP Blocks	1	0

Table1 : Core wise hardware utilization area report

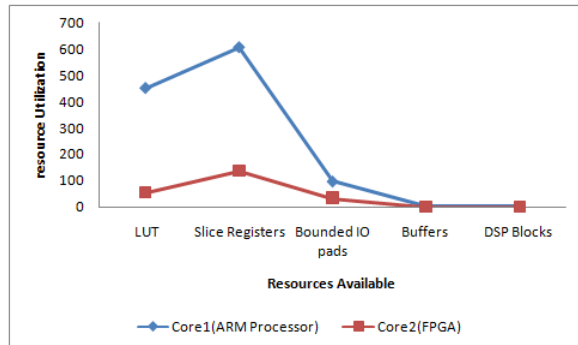


Fig18: Comparative graph of core wise hardware utilization area report.

IV. Conclusion

An efficient method for mapping the most data intensive application on FPGA and the hard core processor embeds the application that does not change with technology. Since FPGA are expandable they aid to expand designs of larger capacity.

References

1. Mario Barbareschi, Antonino Mazzeo, Antonino espoli, 'Malicious traffic analysis on mobile devices: a hardware solution', International Journal of Big Data Intelligence, Vol. 2, No.2 pp. 117 – 126, (2015).
2. A. Majumdar, S. Cadambi, and S. T. Chakradhar, 'An energy efficient heterogeneous system for embedded learning and classification', IEEE Embedded Systems Letters, vol. 3, no. 1, pp. 42–45, (2011).
3. Jun Yan, Wei Zhang, 'Design and implementation of hybrid multicore simulators' International Journal of Embedded Systems, Vol.4, No.3/4, pp.270 – 275, (2010).
4. K.Sravan Kumar, Babak Yazdanpanah, P Rajesh Kumar, 'Removal of Noise from Electrocardiogram Using Digital FIR and IIR Filters with Various Methods', IEEE ICCSP conference, (2015).
5. Ferdouse L, Akhter N, Nipa TH, Jaigirdar FT. Simulation and performance analysis of adaptive filtering algorithms in noise cancellation. International Journal of Computer Science. 2011 Jan;8(1):185–92.
6. Levent Aksoy, Cristiano Lazzari, Eduardo Costa, et al, IEEE Design of Digit-Serial FIR Filters: Algorithms, Architectures, and a CAD Tool, IEEE transactions on very large scale integration (vlsi) systems, vol. 21, no. 3, (2013).

