A Novel Architecture for Different DSP Applications Using Field Programmable Gate Array

Arindam Biswas, Sankalpa Kumar, Tamal Dutta' Swati Bhaumik and Rounak Das

Department of Electrical and Electronics Engineering, Neotia Institute of Technology, Management and Science, Jhinga, West Bengal, India

ABSTRACT

This paper presents a reconfigurable processor for different digital signal processing applications. The performance of the proposed architecture has been evaluated by taking different dsp applications like Low pass filter, high pass filter, finite impulse response (FIR) filter and FFT module. We designed the architecture of the processor and realizing the architecture using adder, multiplier, delay unit and validate it in the FPGA, which show that the hardware scheme is feasible for practical application. The experimental results clearly reveal the novelty of the architecture for dsp applications. This paper investigates the potential use of FPGAs for implementing efficient "Reconfigurable Processor" for different dsp applications. The proposed processor is based on parallel re-configurable which is implemented on FPGA. FPGAs have become an important component for implementing these functions with respect to cost, performance and flexibility. The general purpose SPARTAN 3AN FPGA kit has been employed for developing reconfigurable processor, with all the coding done using the hardware description language VERILOG.

KEYWORDS

Field programmable gate array (FPGA), reconfigurable architecture, digital signal processor (DSP), application specific integrated circuit (ASIC), basic building block (BBU), functional unit (FU), system on chip (SOC), configurable logic block (CLB), look up

1. INTRODUCTION

An FPGA architecture makes an application of programmable logic easier and more convenient. FPGA architecture has a dramatic effect on the quality of the final device's speed performance, area efficiency, and power consumption. [2].Modern day's digital signal processing (DSP) and its applications have become more complex owing to the fact that more and more numbers of datas or signals or user inputs are needed to be screened, processed; also simultaneously needed to be fetched as output. The need of a modern day separate processor is imperative as it's more efficient from regulatory perspective.DSP functions [3],[4] are computationally intensive and exhibit spatial [5], [6] parallelism, temporal [7] parallelism or both. Although Even though higher performance achievement, relatively lower cost and low power dissipation are the major advantages of ASICs, high degree of inflexibility restricts their usage for rapidly changed scenario in the current high end applications as mentioned above. Convenient and well versed mapping is one of the major aspects behind huge popularity of FPGA against ASIC. On the first section of this project, the adopted DSP theorems have been explained rigorously. On the next section, applications of those DSP theorems and their reconfigurable algorithms are duly mapped using map theory (union) and final element has been produced. Section- II of the paper describes different DSP functions (like FIR, FFT, LPF, HPF.) in the proposed architecture and their implementation. Section-III of the paper describes the detailed representation of "Reconfigurable

DOI: 10.5121/ijme.2016.2202

Architecture". Section-III of the paper describes FPGA implementation of the proposed architecture. Section-V analyzes the performance with various simulations, implementation and comparison results and Section- VI concludes the paper.

2. DIFFERENT DSP FUNCTIONS AND PROPOSED IMPLEMENTATION

2.1. Fast Fourier Transform:

A fast Fourier transform is used to calculate the discrete Fourier transform (DFT) of a sequence, suitable for real time applications. Fourier analysis as is known, converts the input signal from discrete time domain to a representation in the frequency domain. Fast Fourier Transform is basically an algorithm for reducing the complexity of calculation in Discrete Fourier Transform (DFT). Although there have been many different FFT algorithms as per different decimation tactics; a simple complex-number computational algorithm has been described here, in adequate of their general properties. The basic building block of FFT are multiplier, adder, subtractor etc.

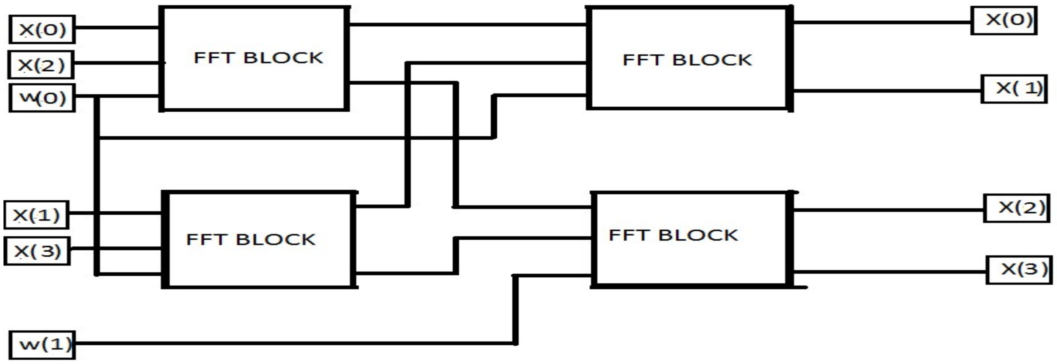

Fig. 1 Schematic diagram of a 4 point FFT

2.2. Finite Impulse Response (FIR)

A finite impulse response (FIR) filter is a filter whose impulse response is of finite duration, because it settles to zero in finite time. For a causal discrete-time FIR filter of order N, each value of the output sequence is a weighted sum of the most recent input values. The figure2 shows 4bits FIR implementation using look up tables (LUTs).The basic building blocks of FIR are adders, multipliers etc .

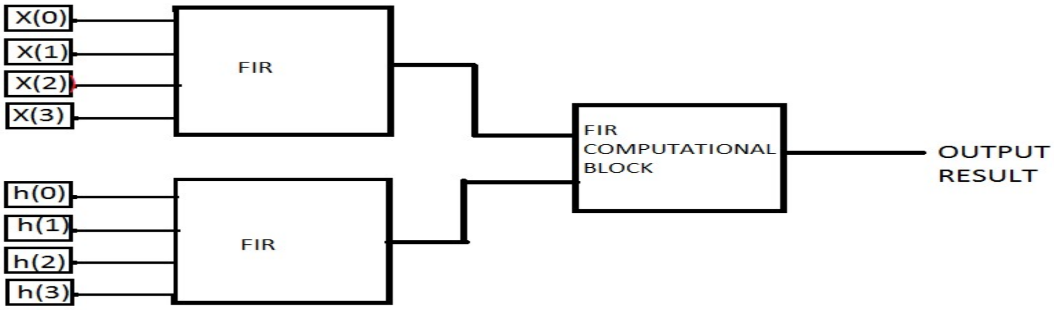

Fig.2: Circuit diagram of FIR

2.3 Filters

Signal in use are generally contains harmonics, and other types of unwanted components. Before employing it to a system we need to eliminate these components for proper processing operation. These noise signals are eliminated using filter circuits. Depending on the zone of operation filters can be of many types like high pass, low pass, band pass, band reject etc

(A)Low Pass Filter*:*

It works on the principle of frequency range variation. Basically if the input signal contains some components having higher frequency than the cutoff frequency then this filter eliminates those components.

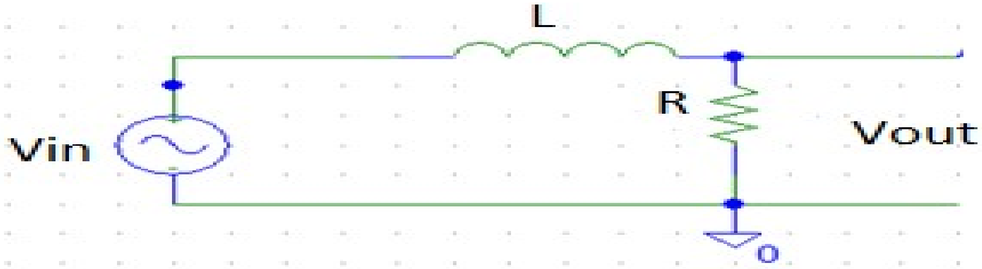

Fig.3 inductive lows pass filter circuit

Here fig 3 is the low pass filter using capacitor, fig (ii) is the low pass filter using inductor. Both of these circuits can be used for implementing low pass filter using direct active elements, but in case of integrated circuit inductor can`t be fabricated therefore in later discussion capacitor circuit is only considered.

We know the reactance of capacitor is $x=\frac{1}{jwc}$ where,

x= reactance w=angular frequency of signal C=capacitance f = frequency of signal

from the equation, impedance increase with the increase of frequency(as we know w ∞f) therefore as the frequency increases circuit blocks the signal, hence the circuit blocks all the noise signals having higher frequency than cut off

.

Cutoff frequency: We know $fc=\frac{1}{2*\pi*R*c}$

Where, f_c =cutoff frequency

R= total resistance of circuit

C= capacitance

Hence by choosing proper value of R and C, any cut off frequency can be applied to the filter.

Calculation for Digital Low pass Filter:

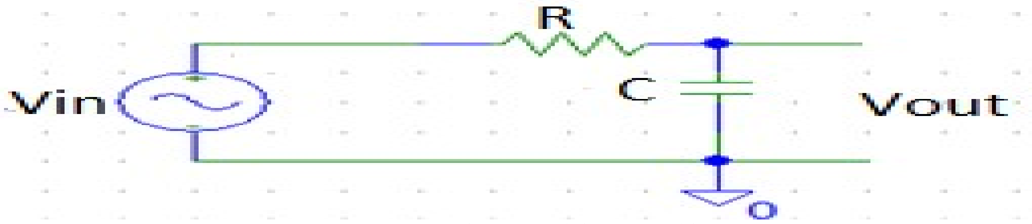

Fig.4: Circuit diagram of low pass filter

In the circuit,

V_{in}= input voltage ; V_{out}=output voltage;

R=Resistance; C=capacitance

Now, Using KVL we have

$V_{in}(s)=I(s)*R+\frac{I(s)}{C*S}$ …………………………………..(i)

$V_{out}(s) =\frac{I(s)}{C*S}$………………….…………………….(ii)

Now dividing equationn.i by equation.ii, we have,

$\frac{Vout}{Vin} = (\frac{1}{C*S}) / (R+\frac{I}{C*S})\ \ \frac{1}{R*C*S+1}$…………………......(iii)

Putting the value of S = (2/T)*($\frac{Z-1}{Z+1}$) from bilinear transformation, We have $H(Z)=\frac{1}{R*C*\frac{2}{T}*\frac{Z-1}{Z+1}+1}$

$$H(Z)=\frac{Z+1}{R*C*\frac{2}{T}*(Z-1)+(Z+1)}$$

$$H(Z)=\frac{Z+1}{\left(Z*R*C*\frac{2}{T}\right)-2*R*\frac{C}{T}+Z+1}$$

$$H(Z)=\frac{Z+1}{Z\left(1+\ 2*R*\frac{C}{T}\right)+\ (1-2RC/T)}$$

$$\frac{Y(Z)}{X(Z)} = (Z+1)/(Zm+n)$$

Z m y(Z) + n y(Z) = Z x(Z) +X(Z)

m y(Z) + nZ^{-1} y(Z) = x(Z) +Z^{-1}x(Z)

m y(n) + n y(n-1)= x(n)+ x(n-1)

$$y(n)=\frac{x(n)+x(n-1)-ny(n-1)}{m}$$

$H(z)=\frac{Y(Z)}{X(Z)}$; y(n)=output ; x(n)= input

X (Z) &Y(Z) are Z transform of x(n)&y(n) respectively.

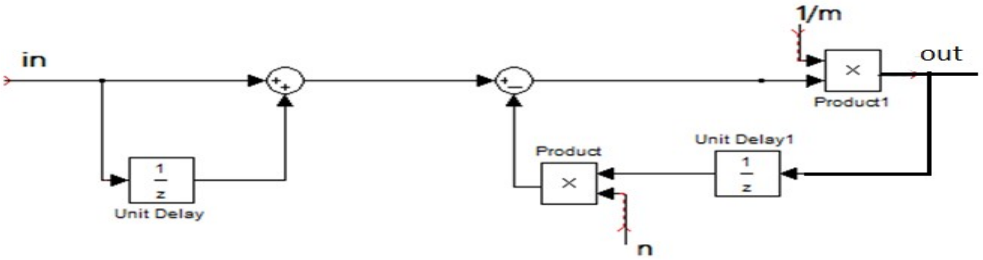

Fig:5 Digital Low pass filter circuit

Here, ‘in’ is input sequence to the system, ‘out’ is output sequence generated by system, ‘m’ and ‘n’ are the terms dependent of circuit resistance and capacitance.

B) Highpass Filter:

In this filter all the signal components having lower frequency than the given cutoff frequency are eliminated. Therefore if any system required to pass only certain amount of frequency then a high pass filter can be applied before the system.

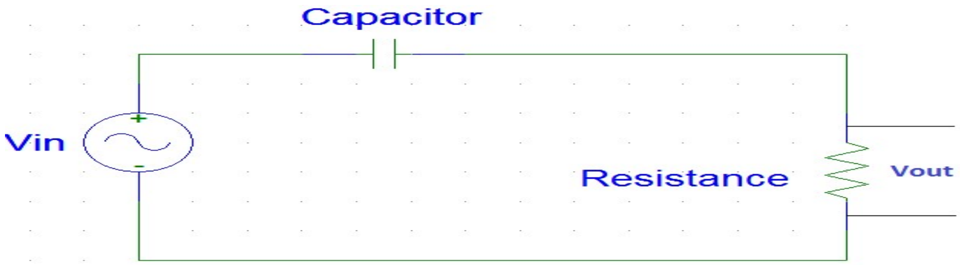

Fig:6 Capacitive high pass filter circuit

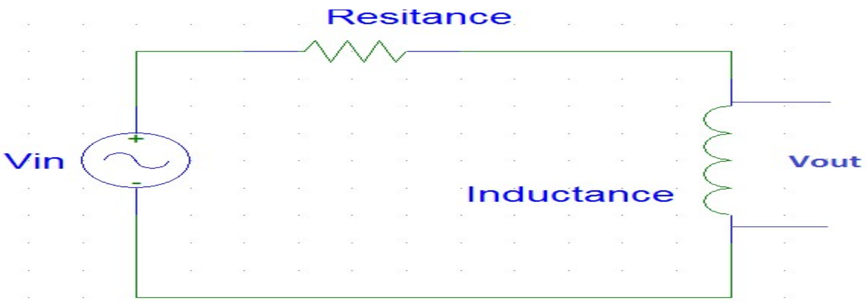

Fig.7 Inductive high pass filter circuit

As the inductor circuit is not implementable in integrated circuit, here capacitive circuit is considered only. We know for a capacitor, reactance decreases with the increase of frequency, as f ∞ w ∞ (1/x); Where x is the reactance, f and w are frequency and angular frequency of the sequence. Now impedance Z.Therefore the output voltage across the resistance is Vout. Now as, with increase of f, x decreases therefore Z also decreases, so Vout increases. Hence for low frequency the circuit will show output tending to zero but for a frequency more than cutoff frequency, system will provide respective output value.

Calculation for Digital High pass Filter:

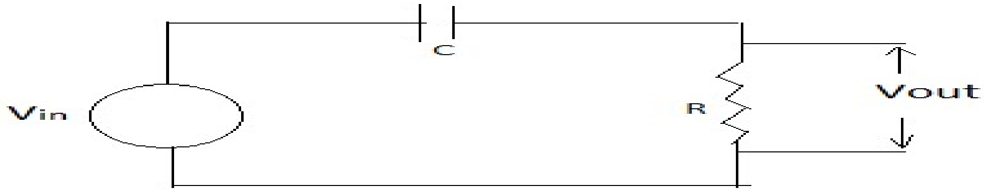

Fig 8.Circuit diagram of high pass filter

In the circuit,

V_{in}= input voltage; V_{out}=output voltage;
R=Resistance;   C=capacitance

Applying K.V.L,
$V_{in}(S) = \frac{1}{cs} * I(s) + R*I(s)$…………..………………(i)
$V_{out}(s) = R*I(s)$ ……………………………………(ii)
Therefore,
$H(s) = \frac{R*I(S)}{I(s)*(\frac{1}{c*s}+R)}$ ……………………................... (iii)

$H(s) = \frac{R*c*s}{1+R*c*s}$

$H(s)= \frac{1}{1+\frac{1}{R*C*S}}$

Putting $S=(2/T)*\frac{Z-1}{Z+1}$,

$H(Z)= \frac{1}{1+\frac{1}{R*C*\left(\frac{2}{T}\right)*\frac{Z-1}{Z+1}}}$

$H(Z)= \frac{Z-1}{P(Z+1)+(Z-1)}$ [let, $P=\frac{T}{2*R*C}$]

$H(Z)= \frac{Z-1}{Z(P+1)+(P-1)}$

$H(Z)= \frac{Z-1}{Za+b}$ [let, a=P+1, b=P-1]

Let, Y=Vout, X=Vin

$(Y(z)/X(z))= \frac{Z-1}{Za|b}$

$(Za+b)Y(z)=(z-1)X(z)$

$ZaY(z)+bY(z)=zX(z)-X(z)$

$ZaY(z) =zX(z)-X(z)- bY(z)$

$aY(z) =X(z)-Z^{-1}X(z)- b Z^{-1}Y(z)$

$Y(n)=\frac{X(n)-X(n-1)-bY(n-1)}{a}$

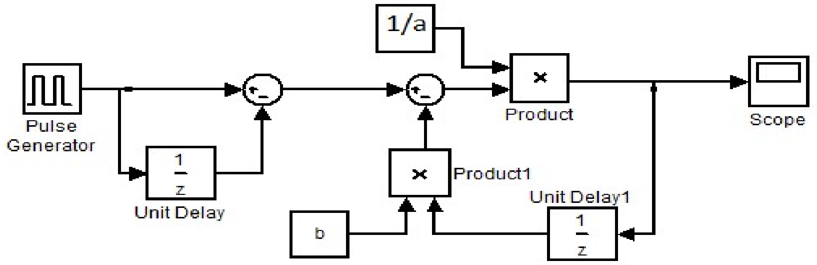

Fig:9 Digital High pass filter circuit

Here pulse generator works as input and scope is taken to display output.

3. Proposed Reconfigurable Architecture:

The proposed reconfigurable architecture is the combination of all the architectures of single element DSP modules. The computing architecture and software model is flexible enough from the perspective of data path and logic pathway. The reconfigurable design adds to the uses of control flow. The proposed architecture has been obtained using union theorem of graph theory, where optimum numbers of electronic components (adders, subtractors, multipliers, delay circuits) and switching devices (multiplexer, demultiplexer) have been accounted for.The architecture is such that different configurations can be processed at a time and outputs can be checked one by one.The Table1 shows the control signals for multiplexers to achieve a particular dsp applications. The Primitives have been chosen in such a fashion that a particular dsp application can be made by connecting these primitives in an appropriate fashion [10]. A number of switches (Multiplexer) are required to maintain the interconnections among the primitives for implementing the applications. The architecture was validated in the FPGA. The FPGA architecture is specified by its hardware description language (HDL), similar to that used for an application-specific integrated circuit (ASIC) [11]. FPGAs of recent make have large resources of logic gates and RAM blocks to implement complex digital computations. As FPGA designs employ very fast I/O devices and bidirectional data buses, it becomes a challenging task to verify correct timing of valid data within the setup and the hold time. Floor planning enables resources allocation within FPGA to meet these time constraints [12–15]. The main advantage of proposed architecture is to perform different control systems applications by generating different control signals of these switches.

CONTROL BLOCKS	DM1(S_6) (1:4 DE MUX)	DM2(S_7) (1:2 DE MUX)	DM3(S_8) (1:2 DE MUX)	DM4(S_9) (1:2 DE MUX)	M1(S_1)	M2(S_2)	M3(S_3)	M4(S_4)	M5(S_5)
FIR	00	0	-	0	0	0	-	1	1
LPF	01	1	0	1	1	1	0	0	0
HPF	10	-	1	1	-	-	1	0	0
FFT	11	-	-	-	-	-	-	-	-

Table1: Control Signals for Reconfigurable Architecture

Table2 shows that how many fundamental units (adder, multiplier, delay units etc), different switches like multiplexers and demultiplexers will be required to achieve different dsp applications.

Function	Multiplier	Delay	Adder	Subtractor	Multiplexer (2:1)	De-multiplexer	
						(4:1)	*(2:1)*
FIR	4	3	3	-	4	1	*1*
LPF	2	2	1	1	7	1	*1*
HPF	2	2	-	2	4	1	*1*
FFT	-	-	-	-	-	1	-
Combined	4	3	3	2	7	1	*1*

Table2: Basic Building Blocks for DSP Functions:

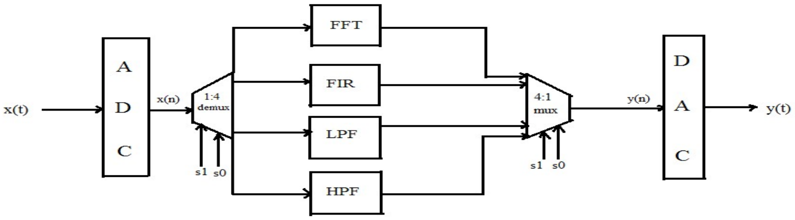

Fig10. Reconfigurable Architecture with BBU

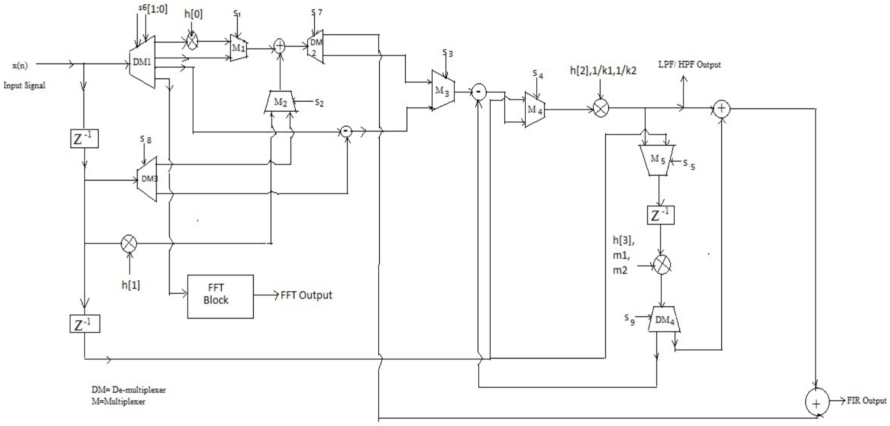

Fig.11. Reconfigurable Architecture using Fus

Function	Multiplier	Registers	Adder	Subtractor	LUTs
FIR	16	-	18	18	356
LPF	2	8	3	3	19
HPF	2	8	4	4	16
FFT	4	-	12	12	70
Reconfigurable Architecture	4	12	7	6	40

Table 3: Synthesis Report

4. FPGA IMPLEMENTATION

This section describes the design, implementation, and verification of the proposed reconfigurable architecture on an FPGA platform.

4.1 Design and Implementation:

In the design stage, the FPGA spartan3AN, with a 50-MHz clock is adopted as the FPGA hardware platform. As the proposed architecture is realized digitally, for that we have realized individual blocks using functional units (Adder, Multiplier, Delay units). It is evident, that a particular dsp application can be achieved by providing appropriate control signal to the control units of the multiplexer. Implementing different dsp applications we are following certain steps. First identify the BBS and then place them in appropriate positions and then replace the BBBs with FUs and connect them through switches. Data routing from output of one FU to inputs of one or more FUs can be achieved by providing appropriate control signals to the switches. Thus by changing control signals of these switches, different dsp applications can be established. As the FPGA kit SPARTAN 3AN contains 8 switches for inputs, 4-bit operation is done. Switches are being used as follows: Input to the FPGA chip for the present applications are clock and reset discrete signals. The main output is another signal. The FPGA device utilization details for the different BBUs are shown in HDL synthesis report.

5. ANALYSIS

It is evident that the hardware requirement is reduced. This is due to the fact that LUTs used in the FPGAs are totally replaced by the BBS like multiplier, adder etc, in the proposed architecture so, power requirement is decreased, and hardware complexity is reduced, and speed which is closed to ASIC. Figure 3 shows the hardware comparison of

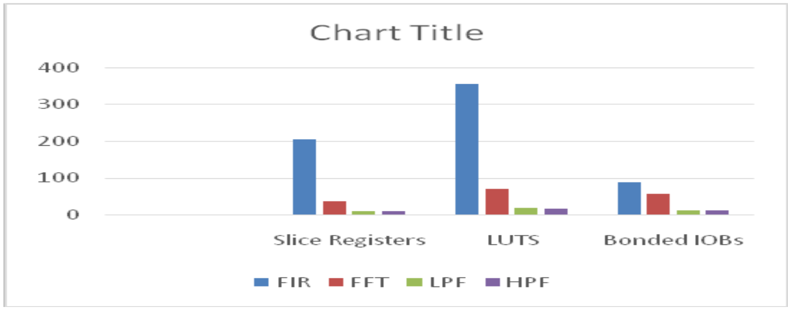

Fig.13.Hardware Comparison of Different DSP Applications

6. CONCLUSIONS

This study proposes the hardware implemented of reconfigurable processor for different dsp applications and validate it in the FPGA which has the merits in term of fast speed, cost, power consumption and flexibility. The experimental results reveal that these FPGAs can offer the cost effective solution because of the availability of basic DSP building blocks like multipliers, adders and MAC units. Hence, the higher performance of the FPGAs compared to conventional DSP processors is achieved. Here a new architecture "Reconfigurable dsp Processor" is introduced which is software controlled hardware implemented using SPARTAN3AN FPGA kit. FPGAs provide an affordable, customised option for testing the performance of new architecture. The prototype has been developed to evaluate the performance of the proposed logic in real-time. However, it is worth investigating the cost complexity analysis of the architecture if the basic building blocks like adders/ subtractors, multipliers can be represented by other number system i.e DBNS, RNS etc. Future work includes a VLSI implementation of the proposed architecture.

REFERENCES

1. Amrita Saha, Amitabha Sinha. Radio Processor – A New Reconfigurable Architecture for Software Defined Radio,
   Proc. ICCSIT, IEEE conf., Singapore, 2008.
2. Amrita Saha, Amitabha Sinha. An FPGA Based Architecture of A Novel Reconfigurable Radio Processor for Software Defined Radio, IEEE *conf.*, April 2008.
3. Amiya Karmakar, Amitabha Sinha. Novel Architecture of a Reconfigurable Radio Processor for Implementing Different Modulation Schemes, IEEE *conf.*, 2011.
4. Chris H Dick. Design and Implementation of High-Performance FPGA Signal Processing, *Datapaths for Software Defined Radios*. Xilinx Inc., www.xilinx.com.
5. www.ti.com
6. www.analog.com
7. Xilinx. *Introduction and overview, vertex-II Pro Platform FPGAs*, March 9th, 2004.
8. Chris H Dick. Design and Implementation of High-Performance FPGA Signal Processing. *Datapaths for Software Defined Radios*. Xilinx Inc., www.xilinx.com.
9. Parthi KK. VLSI Digital Signal Processing Systems. *A Wiley-Inter science Publication*, 1999.
10. *Buyer's guide to DSP Processors*, 2004 Edition. Barkley Design Technology Inc., www.BDTI.com.
11. Joanne De Groat, Gursharam Reehal, Nagarjuna. Synthesizing FPGA Digital Modules for Software Defined Radio, *IEEE conf.* 2008.