International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 06 | June -2017 www.irjet.net p-ISSN: 2395-0072

FPGA BASED RECONFIGURABLE COMPUTATION UNIT FOR DSP APPLICATIONS

ALIYA SHAHUL[1], Dr. SANTHOSH KUMAR [2]

[1] Mtech Scholar, Department of ECE, GECI, Kerala, India
[2] Assistant Professor, Department of ECE, GECI, Kerala, India

---

Abstract - *This paper presents the architecture of a reconfigurable and high performance computation unit for DSP applications called Reconfigurable Computation Unit. The architecture operates based on fast Carry-Save (CS) arithmetic. A multirate processor generated using this RCU. Thus a mapping methodology also proposed for datapaths of DSP functions. The proposed multirate processor exploits the features of this flexible ALU. RCU enables fast as well as reconfigurable manipulation of DSP function. Several efficient DSP algorithms can be mapped using this RCU architecture. Experimental results have been shown that, by maintaining same power consumption, the system based on RCU is reconfigurable and faster than system based on other techniques.*

Key Words: **Carry-Save (CS) arithmetic, Multirate processor, Reconfigurable Computation Unit(RCU), Flexible Arithmetic and Logic Unit (ALU), Fractional rate conversion**

1. INTRODUCTION

DSP has a wide range of applications including digital image processing, audio signal processing, video compression even in radars, sonar etc. Most of the high end application domains are based on DSP. So the efficient implementation of computationally intensive DSP functions is very important. DSP functions can be implemented on DSPs, ASICs, FPGAs etc.

DSPs show optimization in the use of transistor and clock cycles for an operation, at the expense of flexibility. Field programmable gate arrays (FPGAs) are easy to code and are flexible but limited optimization of clock. ASICs are is custom-designed for a particular application This optimizes the number of transistors and clock cycles, at the expense of development time , cost and flexibility.

Digital Signal Processors (DSP) is a system that takes real-world signals, digitized it and mathematically manipulated. So basically, a DSP is designed for performing these mathematical functions like add, subtract, multiply and divide very quickly. At the same time in this fast moving world, everyone wish to have systems that are flexible /reconfigurable. So, faster as well as flexible/reconfigurable systems gaining importance.

The main constraint of existing DSP systems is its inflexibility. The main focus of my work is to implement reconfigurable DSP functions. Field Programmable Gate Arrays (FPGAs) are becoming widely popular because of its reconfigurable property. So objective is to implement a flexible DSP system. Here, presenting a flexible ALU for DSP application. The flexible ALU is called as Reconfigurable Computation Unit (RCU).It exploits the advantage of CS arithmetic and enables the execution of large number of templates found in DSP kernels

2. RECONFIGIRABLE COMPUTATION UNIT

As its name indicates, Reconfigurable Computation Unit (RCU) is a flexible ALU for DSP applications. The structure of this reconfigurable ALU is as shown in the Fig -1.The RCU operates on 16-bit operands. For almost all modern DSP applications such a bit-length is sufficient. At the same time the architectural concept can be straightforwardly used for any bit-lengths. The RCU process the data in carry save format (CS-format) .So that the data can be directly reuse in the intermediate results.

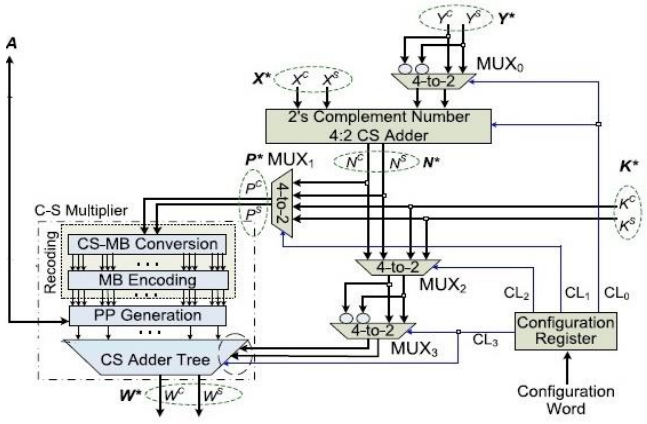

Fig -1: Structure of RCU

RCU enables high-performance and flexible operation based on a library of operation templates. It can be configured to a number of operations. RCU enables intra template operation chaining by fusing the additions performed before/after the multiplication and performs any partial operation template .

RCU comprises of four 4:2 multiplexers MUX1, MUX2, MUX3 and MUX4, carry save adder, configuration register, multiplier and carry save adder tree. The adder is a carry save adder, so that unwanted delays due to carry propagation can also be avoided. The multiplier is a sum to modified booth multiplier[2]

Configuration register is controlled by a 16-bit control word. For each control word the register generates appropriate 4-bit control signal. The control signal properly controls the multiplexers and generates alternative execution paths. The multiplexer MUX0 and the MUX3 determines whether an addition or a subtraction is required.MUX2 and MUX4 do selection. Different functions that generated using RCU is called as templates. Template library is as on table 1.

Table -1: Library of templates

Control Word	CL3 CL2 CL1 CL0	Functions
100	0000	(X*+Y*)A+(X*+Y*)
101	0001	(X*-Y*)A+(X*-Y*)
102	0010	K*A+(X*+Y*)
103	0011	K*A+(X*-Y*)
104	0100	(X*+Y*)A+K*
105	0101	(X*-Y*)A+K*
106	0110	K*A+K*
107	0111	K*A+K*
108	1000	(X*+Y*)A-(X*+Y*)
109	1001	(X*-Y*)A-(X*-Y*)
110	1010	K*A-(X*+Y*)
111	1011	K*A-(X*-Y*)
112	1100	(X*+Y*)A-K*
113	1101	(X*-Y*)A-K*
114	1110	K*A-K*

3. PROPOSED SYSTEM

Here proposing a high performance architectural scheme for a multirate processor based on this RCU. Here input is filtered and then passed through a fractional rate converter. The structure of the system is as shown in the Fig -2.

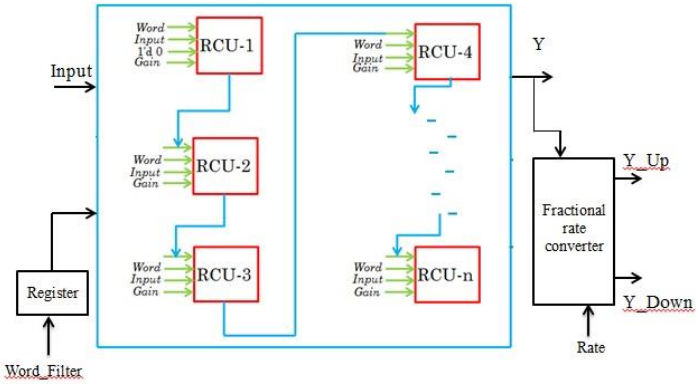

Fig -2: Multi Rate Processor

This system consists of a register, FIR and a fractional rate converter. Input is fed directly to the system. Input can passed through low pass filter, high pass filter, band pass filter and band reject filter depending upon the control word that given to the register. This filtered output undergoes fractional rate conversion at desired rate.

3.1 Register

Register is controlled by the 16-bit control word 'word_filter'. Register stores the gain values for the different filtering action. According to the 'word_filter', required set of gain values is passed to the next section. As FPGAs process only binary data, floating point gain values is digitalized (8-bit integer + 8 bit decimal) and stored in register. The register organization is as shown in the figure 4.2.

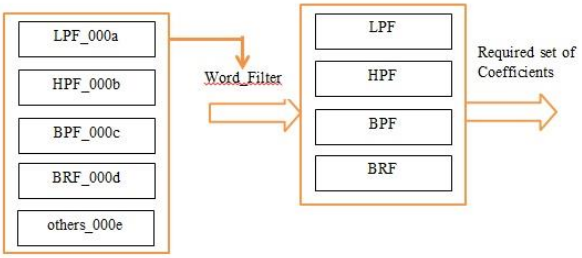

Fig -3: Register Organisation

3.2 FIR

FIR section is designed using a number of RCUs. FIR is generated from the templates library of RCU. Each RCU is controlled by a control word in order to generate required templates. The templates generated by each RCU are properly combined to get an efficient and reconfigurable FIR.As the datapath is also programmed, alterations can be done easily. The order of the system can be changed easily, we can incorporate additional elements etc.

Here, it is an FIR using the RCU. Likewise lot of DSP functions like FFT, DCT etc can also be generated out of this RCU. Also here the template library consists of fourteen functions. By making small alterations in the basic structure of RCU, the template number can be increased. Thus more and more efficient and reconfigurable DSP functions can be generated out of this RCU. So the application of this RCU in DSP is very large.

3.3 Fractional Rate Converter

This section consists of an up sampler and a down sampler. The input to this section undergoes both upsampling and down sampling. Sampling rate can be controlled by a 16-bit control word 'Rate'. As per the current structure of the system, rate can only be 2, 4 and 8.This is a reconfigurable system, so by small alteration rate can be expandable.

Up-sampling is the process of increasing the sampling frequency. Usually upsampling is done before digital to analog conversion in order to relax the requirements of the analog low pass antialiasing filter. This technique is used in audio CD. . Opposite to upsampling ,downsampling is the process of decreasing the sampling rate. Downsampling is done to decrease the bit rate when transmitting over a limited bandwidth. Various systems in digital audio signal processing often operate at different sampling rates. The connection of such systems requires a conversion of sampling rate. The system that satisfies this requirement is called as Multi Rate Processor.

4. RESULTS

Simulation results are given below. Coding was done using Verilog HDL (Hardware Description Language). The simulation was done in Xilinx ISE Vivado Design Suite 14.2.

4.1 RCU

Fig 4 shows the simulation result of RCU. It shows the output corresponds to the different control signals.'inp' represents the 16 bit control word. DSPans represents the output.

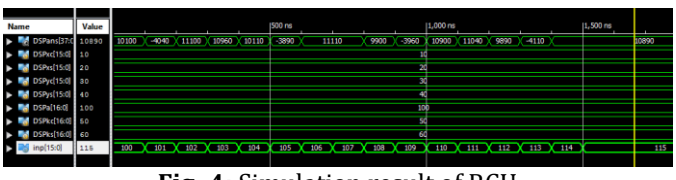

Fig -4: Simulation result of RCU

4.2 Proposed System

Simulation result of multi rate processor is as shown in the Fig -5.Here the signals clk and rst represents clock and reset respectively.'Y' is filtered output,Y_UP is upsampled output and 'Y_DWN'is down sampled output.

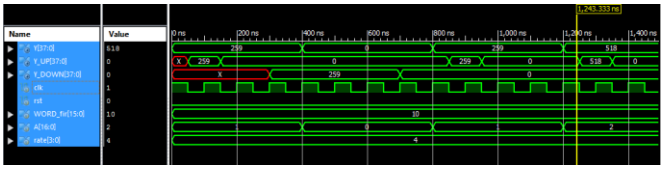

Fig -5: Simulation result of multi rate processor

4.3 Timing Simulation

Timing simulation is done to calculate the path delays as well as to check whether the system meets setup and hold time constraints. Apart from the input and output, several intermediate signals are shown in the waveform which results due to the interconnections made on the FPGA during place and route process of implementation. These additional signals also have contribution to total delay.

The FIR is designed in two ways. One based on RCU. Other one based on Bough Wooley Multiplier (BWM).The timing simulation of FIR using these two different method is shown below. Fig -6 shows timing simulation of RCU based FIR and Fig -7 shows the Timing simulation of BWM-adder based FIR. In figure, the vertical yellow lines indicate the time taken for the output to change after the application of input.

Fig -6:Timing simulation of RCU based FIR

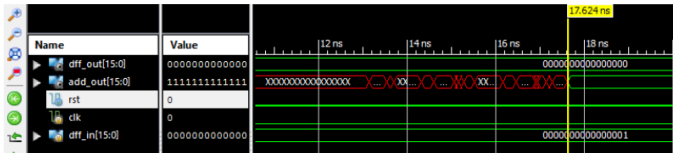

Fig -7: Timing simulation of BWM-adder based FIR

4.4 Performance Analysis

Compared the performance of FIR designed using RCU and an FIR designed using Bough Wooley multiplier (BWM) and adder. Then it is found that the delay of RCU based FIR as 3ns and that of other as 16ns.The power consumption of both systems is found to be same, .321W. Thus proved that RCU based architecture is reconfigurable as well as faster.

Table -2: Perfomance analysis

FIR	Delay	Power
RCU	3ns	.321 W
BWM	17ns	.321 W

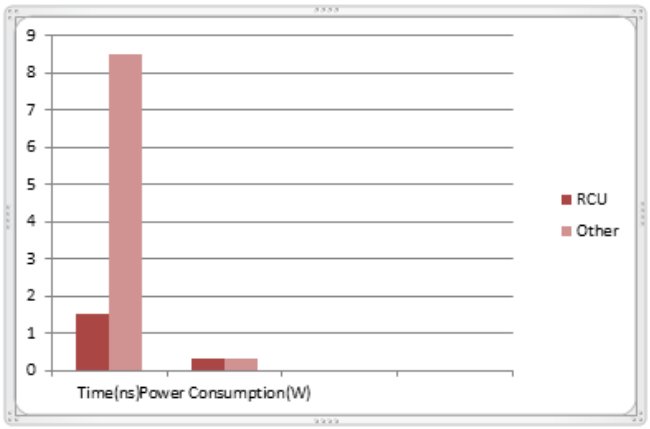

Fig -8: Performance analysis in chart

5. CONCLUSIONS

As most of the high end application domains are DSP based, the efficient implementation of computationally intensive DSP functions is very important. An ALU for DSP functions called reconfigurable computation unit (RCU) has been implemented. The RCU is faster as well as reconfigurable/flexible one. A multirate processor based on this RCU has been developed. By maintaining same power consumption, it is found that the system based on RCU is faster than system based on other techniques. By proper designing lot of faster and reconfigurable DSP functions can be generated using this RCU. The application of this RCU is very large in DSP.

REFERENCES

[1]Kostas Tsoumanis, Sotirios Xydis, Georgios Zervakis, and Kiamal Pekmestzi" Flexible DSP Accelerator Architecture Exploiting Carry-Save Arithmetic" IEEE Trans. on very large scale integration (VLSI) systems, vol. 24, no. 1, January 2016

[2]Kostas Tsoumanis, Student Member, IEEE, Sotiris Xydis, Constantinos Efstathiou, Nikos Moschopoulos, and Kiamal Pekmestzi"An Optimized Modified Booth Recoder for Efficient Design of the Add-Multiply Operator" IEEE transactions on circuits and systems—i: regular papers, vol. 61, no. 4, April 2014

[3]Sotiris Xydis, Isidoros Sideris, George Economakos and Kiamal Pekmestzi" A flexible architecture for dsp applications combining high performance arithmetic with small scale configurability" 16thEuropean Signal Processing Conference,Lausanne, Switzerland, August 25-29, 2008, copyright by EURASIP

[4]P. Stelling, V. Oklobdzija, "Implementing Multiply-Accumulate Operation in Multiplication Time," in proccedings of 13th IEEE Symposium on Computer Arithmetic, 1997, p. 99.

[5]Yuyun Liao, D.B. Roberts, "A High-Performance and Low-Power 32-bit Multiply-Accumulate Unit with Single-Instruction-Multiple-Data (SIMD) Feature," Solid-State Circuits IEEE Journal, vol. 37, no. 7, pp. 926–931, July 2002.

[6]S. Xydis, G. Economakos, and K. Pekmestzi, "Designing coarse-grain reconfigurable architectures by inlining flexibility into custom arithmetic data-paths," Integr., VLSI J., vol. 42, no. 4, pp. 486–503, Sep. 2009.

[7] S. Xydis, G. Economakos, D. Soudris, and K. Pekmestzi, "High performance and area efficient flexible DSP datapath synthesis," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 3, pp. 429–442, Mar. 2011.

[8]Hosangadi, F.Fallah, and R.Kastner, "Optimizing high speed arithmetic circuits using three-term extraction" in Proc. Design, Autom. Test Eur. (DATE), vol. 1. ,pp. 1–6.