iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 Issue: VI Month of publication: June 2015

DOI:

IIR filter design using CSA for DSP applications

Sagara.K.S[1], Ravi L.S[2]
[1]PG Student, Dept. of ECE, RIT, Hassan, [2]Assistant Professor Dept of ECE, RIT, Hassan

Abstract- In this paper, a design methodology to implement low-power and high-speed 2nd order recursive digital Infinite Impulse Response filter has been proposed. Existing IIR digital filters have a large number of constant multiplications. The proposed method replaces constant multiplications by using addition/subtraction and shift operations. The proposed new adder cell is used as the Carry-Save Adder to implement addition/subtraction operations in the design of recursive section IIR filter to reduce the delay. Furthermore, high level algorithms designed for the optimization of the number of CSA blocks are used to reduce the complexity of the IIR filter. The microwind tool is used to generate the schematic of the proposed CSA based shift-adds architecture design and it will be analyzed by using Micro wind CAD tool to synthesize low-complexity and high speed IIR filters. The proposed design reduces power, propagation delay, area and increase throughput when compared with the existing technology.

Keywords- IIR, CSA, DSH, Xilinx, FPGA

I. INTRODUCTION

Digital filtering is one of the most widely used operations in digital signal and image processing applications. The digital filtering operation is to transform the input signal or image in such a way to enhance or suppress certain features .IIR filters compute their outputs recursively, it need the immediate past output for computing the current one. Thus IIR filter are more difficult to pipeline than FIR digital filters. On the other hand, the IIR digital filters have the advantages of high selectivity and requiring fewer coefficients than the FIR digital filters with similar performance. Consequently. Realization of IIR filters with good overall performance became a challenging aspect. Systolic arrays are architectures which respond to the requirements of VLSI design by their simplicity, modularity and nearest neighbor connectivity. These characteristics give the systolic arrays a leading edge over other VLSI architecture. Several array designs to compute the IIR digital filtering operation have been proposed. The absence of delay elements inside the feedback loop has limited the possibility of pipelining and consequently the throughput rate of the existing fully bit parallel systolic IIR digital filter architectures. One way to achieve pipelining of IIR filters is to insert delay elements in the feedback loop. This can be achieved by using the scattered look ahead technique or the clustered look ahead technique. The proposed system uses CSA technique along with the delay elements to implement the pipelined IIR digital filter. design methodology

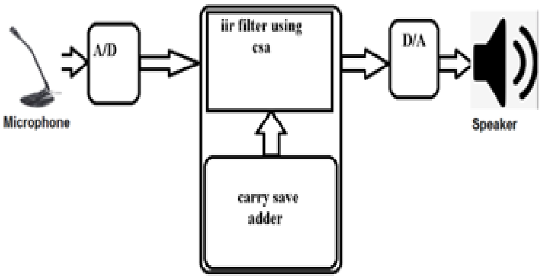


Fig 1: Block diagram of IIR filter using CSA

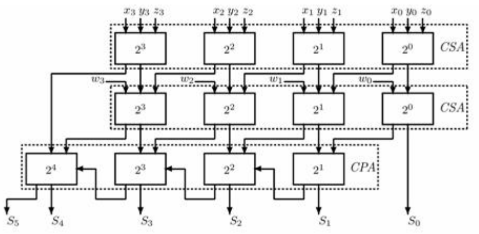


Fig 2: Carry save adder block diagram and schematic

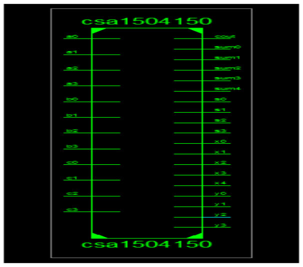


Fig 3: Carry save adder block diagram and schematic

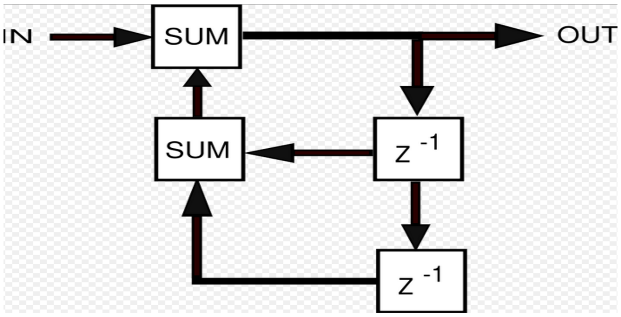


Fig 4: IIR Filter block diagram

The Figure 3 shows the typical block diagram of an IIR filter looks like the following. The z^{1} block is a unit delay. The coefficients and number of feedback/feedforward paths are implementation-dependent and figure 2 shows A carry save adder (CSA) tree consists of CSA operators and one adder at the root of the tree. The CSA operators are used to transform an arbitrary number of operands in the addition process to produce two adding operands, after which the adder at the root of the CSA tree computes the final sum. The proposed design uses this technique to implement the adder/subtraction section along with the delay elements in the pipelining recursive section of the digital IIR filter.

A. *Carry save adder*

The carry-save adder reduces the addition of 3 numbers to the addition of 2 numbers. The propagation delay is 3 gates regardless of the number of bits. The carry-save unit consists of *n* full adders, each of which computes a single sum and carries bit based solely on the corresponding bits of the three input numbers. The entire sum can then be computed by shifting the carry sequence left by one place and appending a 0 to the front (most significant bit) of the partial sum sequence and adding this sequence with RCA produces the resulting *n* + 1-bit value. This process can be continued indefinitely, adding an input for each stage of full adders, without any intermediate carry propagation. These stages can be arranged in a binary tree structure, with cumulative delay logarithmic in the number of inputs to be added, and invariant of the number of bits per input.

B. *IIR Filter*

The general infinite impulse response (IIR) .The filters will now include both feedback and feed forward terms. The system function will be a rational function where in general both the zeros and the poles.

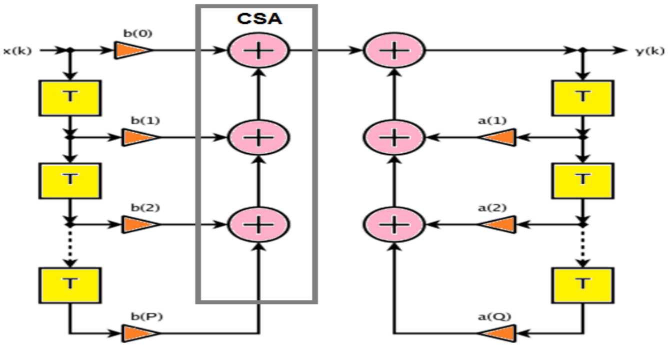


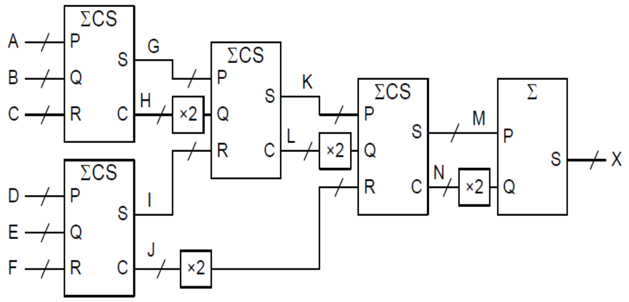
Figure:5 Direct form-I(chebyshev filter-I)

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

A conventional second-order section (SOS) of a digital IIR filter is shown in Fig 4. It contains coefficients c1 and c2, two multipliers, one adder and two unit-delay elements. To maintain stability with the use of fixed precision arithmetic, the filter employs a quantizer (Q). The quantizer is also known as limiter which is used to determine if there is an overflow (or underflow) of vectors and truncate the vectors to a desired length.

C. Carry Save Example

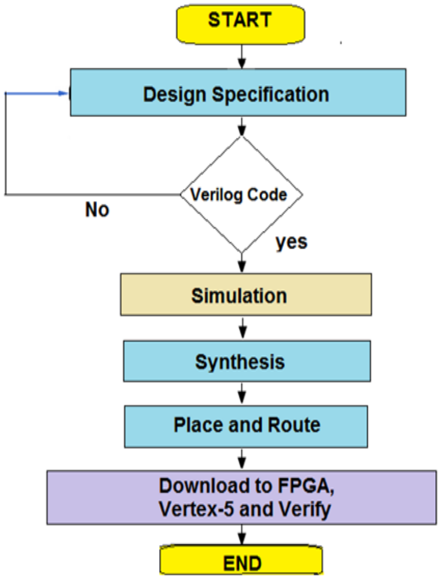
Example1: 13+10+5+11+12+1 = 52



```
 A:    1101        G:  _0010
 B:    1010       2H:  1101_
 C:    0101        I:  _0110          M:    01000
 G:      10        K:  11110         2N:   1011__
 H:   1101_        L: _0010_          X:  0110100

 D:    1011           K:  11110
 E:    1100          2L:  0010_
 F:    0001          2J:  1001_
 I:    0110           M:  01000
 J:   1001_           N: 1011__
                        110100 ⇐ 52
```

II. DESIGN FLOW



6:FPGA Design flow diagram.

A. Three key verification points for FPGA implementation

1) Behavioral simulation

2) Post-place & route static timing analysis

3) Download and verify in circuit

B. Choose the appropriate configuration scheme

C. Create Verilog code

1) Optimized for Xilinx FPGAs and performance

D. Synthesis

1) Synopsys, Mentor, XST

2) Pushbutton flow with scripting capabilities

E. Place & route

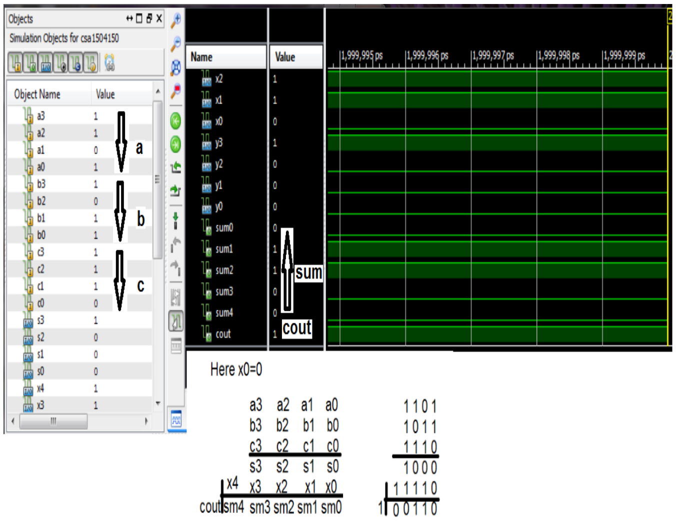
1) Completed by the user

2) Xilinx implementation tools

3) Pushbutton flow, scripting capabilities

F. Download and verify in circuit.

III. SIMULATION RESULTS



Object Name	Value
a3	1
a2	1
a1	0
a0	1
b3	1
b2	0
b1	1
b0	1
c3	1
c2	1
c1	1
c0	0
s3	1
s2	0
s1	0
s0	0
x4	1
x3	1

Name	Value
x2	1
x1	1
x0	0
y3	1
y2	0
y1	0
y0	0
sum0	0
sum1	1
sum2	1
sum3	0
sum4	0
cout	1

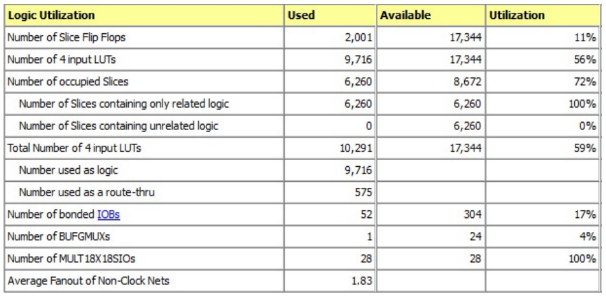
Here x0=0

```
        a3  a2  a1  a0          1101
        b3  b2  b1  b0          1011
        c3  c2  c1  c0          1110
        s3  s2  s1  s0          1000
     x4 x3  x2  x1  x0         11110
cout sm4 sm3 sm2 sm1 sm0      1 00110
```

Fig: 7 Output for carry save adder

IV. SYNTHESIS RESULTS

Table:1 Utilization summary of IIR Filter using CSA.



Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	2,001	17,344	11%
Number of 4 input LUTs	9,716	17,344	56%
Number of occupied Slices	6,260	8,672	72%
Number of Slices containing only related logic	6,260	6,260	100%
Number of Slices containing unrelated logic	0	6,260	0%
Total Number of 4 input LUTs	10,291	17,344	59%
Number used as logic	9,716		
Number used as a route-thru	575		
Number of bonded IOBs	52	304	17%
Number of BUFGMUXs	1	24	4%
Number of MULT18X18SIOs	28	28	100%
Average Fanout of Non-Clock Nets	1.83		

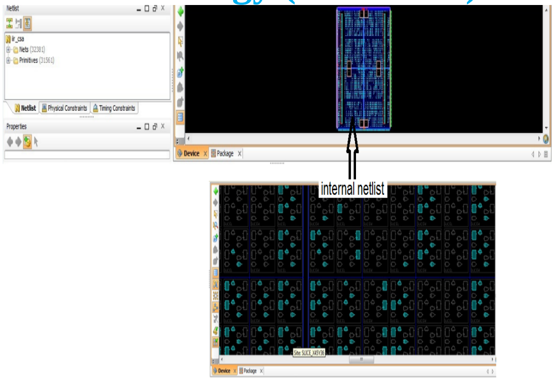


Fig:8 Netlist Design

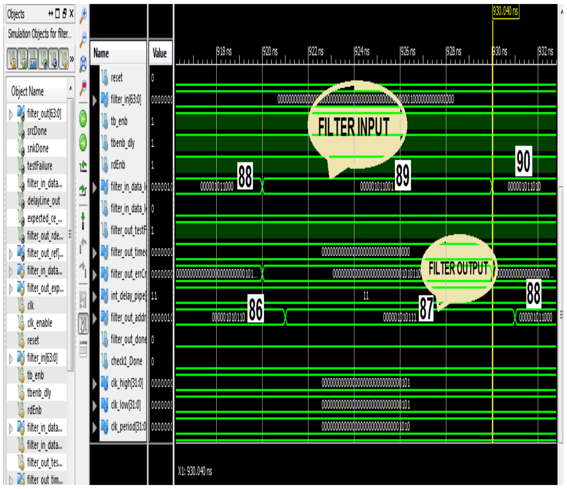


Fig 9: Output for IIR Filter using CSA.**VI.FPGA Implementation results**

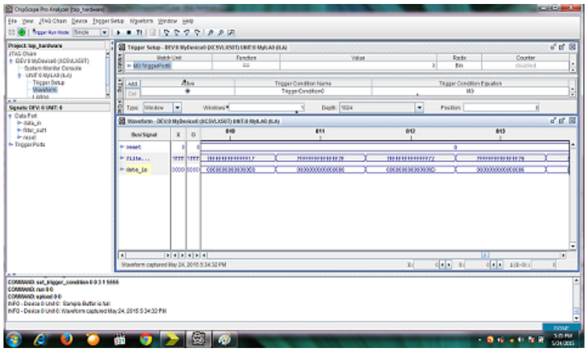

ChipScope Pro Analyzer [top_hardware]
Trigger Setup - DEV:0 MyDevice0 (XC5VLX50T) UNIT:0 MyILA0 (ILA)
Waveform - DEV:0 MyDevice0 (XC5VLX50T) UNIT:0 MyILA0 (ILA)
Project: top_hardware
Signals: DEV:0 UNIT:0
reset
filte...
done_in

Fig 10: Chipscope results

V. CONCLUSION

In this paper, IIR Filter using Carry Save Adder is presented Each design is described, verified and tested for a correct functionality using Verilog and simulation. This Project gives a brief insight into how exactly IIR filter carried out. Relatively short and comprehensive material has been presented on Isim simulator.. Speech signal is converted to digital value implemented in MATLAB. Removes the noise in signals with the help of CSA using IIR filter

REFERENCES

[1] Deepa Yagain, Dr. Vijaya Krishna .A, Akansha Baliga, “Design of High speed adders for Efficient Digital Design Blocks” ISRN Electronics, 2012, to be published.

[2] G. Ramana Murthy, C. Senthilpari, P.Velrajkumar, and Lim Tien Sze, “Monte-Carlo analysis of a new 6-T Full-Adder Cell for Power and Propagation Delay Optimizations in 180nm Process”, in The 2nd Int. Conf. on Engineering and Technology Innovation, Taiwan, 2012, to be published.

[3] Ravinder Kaur, Ashish Raman,” Design and Implementation of High Speed IIR and FIR Filter using Pipelining”, International Journal of Computer Theory and Engineering, vol. 3, no. 2, 2011, pp. 292-295.

[4] K.K. Parhi, D.G. Messerschmitt, “Pipelined interleaving and parallelismin recursive digital filters - Part I: Pipelining using scattered look-aheadand decomposition”, IEEE Trans. Acoust. Speech Signal Process, vol.37, no. 7, 198, pp. 1099-1117.

[5] Z. Jiang, A.N. Wilson, Jr, “Design and implementation of efficientpipelined IIR digital filters”, IEEE Trans. Signal Processing, vol. 43,no.3, 1995, pp. 579-590.

[6] Sweety Kashyap, Mukesh Maheshwari “Implementation of High Performance IIR Filter Using Low Power Multiplier and Adder “Research Scholar, Department of Electronics and Communication Jaipur National University, Jaipur, Rajasthan, India.