International Journal of Electrical and Electronic Engineering & Telecommunications

ISSN 2319 – 2518 www.ijeetc.com
Vol. 3, No. 2, April 2014

Research Paper

COMPARATIVE ANALYSIS OF DIFFERENT AREA EFFICIENT FIR FILTER STRUCTURES FOR SYMMETRIC CONVOLUTIONS

H V Kumaraswamy[1]*, AmruthKaranth P[1], Samarth Athreyas[1] and Akash Bharadwaj B R[1]

*Corresponding Author: H V Kumaraswamy, ✉ kumaraswamyhv@rvce.edu.in

A comparative analysis based on speed vs area is done on parallel, serial, and cascade serial FIR filter architectures and a DTMF test bench is also generated to test the architectures for various input frequencies. The VHDL code generated from MATLAB HDL code generator is synthesized in XILINX ISE 14.2 and the various parameters which determine the speed and area efficiency of various architectures is tabulated and it is found out that for high speed operations Parallel architecture is the best and for applications involving lot of computations where lot of speed is required serial architecture is well suited.

Keywords: Parallel, Serial, Cascade serial, LUT

INTRODUCTION

In signal processing, a Finite Impulse Response (FIR) filter is a filter, whose impulse response (or response to any finite length input) is of finite duration, because it settles to zero in finite time. This is in contrast to Infinite Impulse Response (IIR) filters, which may have internal feedback and may continue to respond indefinitely (usually decaying). The impulse responseof an Nth-order discrete-time FIR filter (i.e., with a Kronecker delta impulse input) lasts for $N + 1$ samples, and then settles to zero. FIR filters can be discrete-time or continuous-time, and digital or analog. The output y of a linear time invariant system is determined by convolving its input signal x with its impulse response b.

For a discrete-time FIR filter, the output is a weighted sum of the current and a finite number of previous values of the input. The operation is described by the following equation, which defines the output sequence $y[n]$ in terms of its input sequence $x[n]$:

$$y[n] = b_0 x[n] + b_1 x[n-1] + \ldots + b_N x[n-N]$$

$$= \sum_{i=0}^{N} b_i x[n-i]$$

[1] Department of Telecommunication Engineering, R.V. College of Engineering, Bangalore, India.

The main disadvantage of FIR filters is that considerably more computation power in a general purpose processor is required compared to an IIR filter with similar sharpness or selectivity, especially when low frequency (relative to the sample rate) cut-offs are needed. However many digital signal processors provide specialized hardware features to make FIR filters approximately as efficient as IIR for many applications.

SPECIFICATIONS AND ARCHITECTURES

In this chapter the specifications for various architectures, the design of the architectures their diagrams, advantages and disadvantages are mentioned. Initially, the design of the architectures along with their circuit diagrams is explained. Then, the advantages and disadvantages of various architectures are mentioned.

Specification

Fs = 48e3;

Fpass = 9.6e3;

Fstop = 12e3;

Apass = 1;

Astop = 90;

The above is the basic FIR filter specification for various architecture and we have chosen various parameters:

- Sampling frequency is 48000 Hz
- Pass band frequency is 9600 Hz
- Stop band frequency is 12000 Hz
- Pass band attenuation is 1 dB and
- Stop band attenuation is 90 dBs'

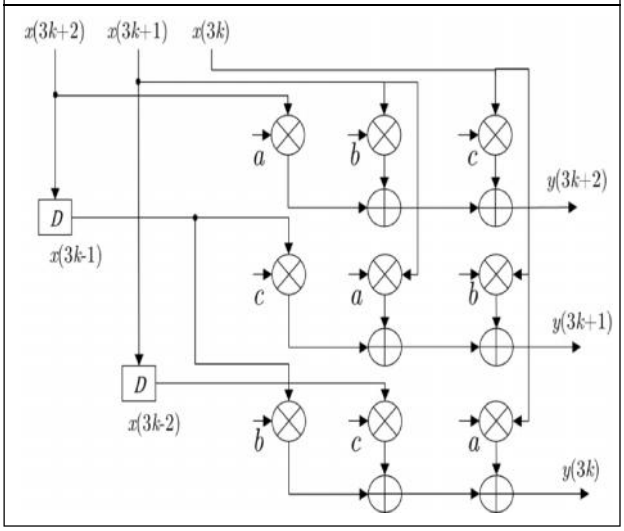
Design of Architectures with Circuit Diagrams

In this section various design of the architectures are mentioned. Architectures like fully parallel, fully serial, cascade are explained.

Fully Parallel Architecture

A fully parallel architecture uses a dedicated multiplier and adder for each filter tap; all taps execute in parallel. We can perform the same processing for different signals on the corresponding duplicated function units. Further, due to the features of parallel processing, the parallel DSP design often contains multiple outputs, resulting in higher throughput than not parallel. Three tap fully parallel architecture is as shown in the Figure 1.

Figure 1: Three-Tap Fir Fully Parallel

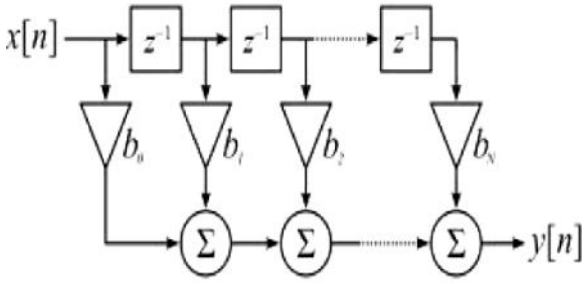


This results in the largest area since it employs a dedicated multiplier and adder for each ûlter tap and all taps execute in parallel. A fully parallel architecture is optimal for speed. However, it requires more multipliers and adders than a serial architecture, and therefore consumes more chip area.

Fully Serial Architecture

In fully serial architecture, instead of having a dedicated multiplier for each tap, the input sample for each tap is selected serially and is multiplied with the corresponding coefficient. For symmetric (and anti-symmetrical) structures the input samples corresponding to each set of symmetric taps are pre-added (for symmetric) or pre-subtracted (for anti-symmetric) before multiplication with the corresponding coefficients. The product is accumulated sequentially using a register and the final result is stored in a register before the next set of input samples arrive. This implementation needs a clock rate that is as many times faster than input sample rate as the number of products to be computed. This results in reducing the required chip area as the implementation involves just one multiplier with a few additional logic elements like multiplexers and registers. A fully serial architecture conserves area by reusing multiplier and adder resources sequentially.

Figure 2: Serial Architecture



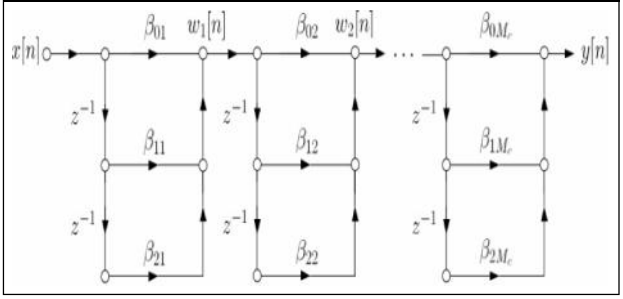
Cascade Architecture

A cascade-serial architecture closely resembles a partly serial architecture. As in a partly serial architecture, the filter taps are grouped into a number of serial partitions that execute in parallel with respect to one another. However, the accumulated output of each partition is cascaded to the accumulator of the previous partition. The output of all partitions is therefore computed at the accumulator of the first partition. This technique is termed accumulator reuse. No final adder is required, which saves area.

The cascade-serial architecture requires an extra cycle of the system clock to complete the final summation to the output. Therefore, the frequency of the system clock must be increased slightly with respect to the clock used in a non-cascade partly serial architecture.

To generate a cascade-serial architecture, you specify a partly serial architecture with accumulator reuse enabled. If you do not specify the serial partitions, the coder automatically selects an optimal partitioning. Cascade architecture is as shown in the Figure 3.

Figure 3: Cascade Architecture



RESULTS AND INFERENCE

The simulation results for four architectures are obtained for 1) Family: ARTIX7, 2) Device: XCTA100T, 3) Package: CSG324 and the delays and the hardware is tabulated. The results for the architectures are compared and speed versus area performance is compared.

Simulation Results

Simulation results for various architectures are obtained and are shown in the below figures.

Figure 4: Gain Characteristic for Fully Serial Architecture

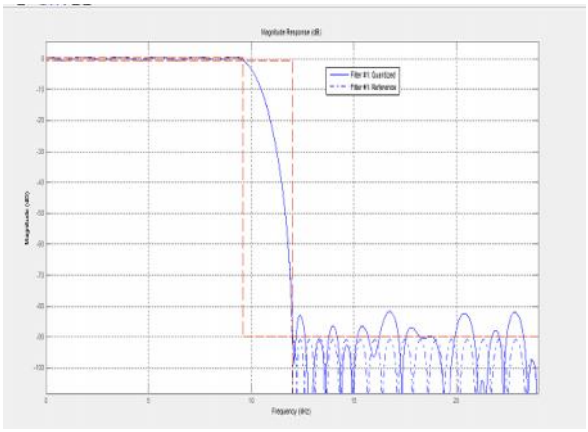


Figure 5: Simulation Results for Fully Serial Architecture

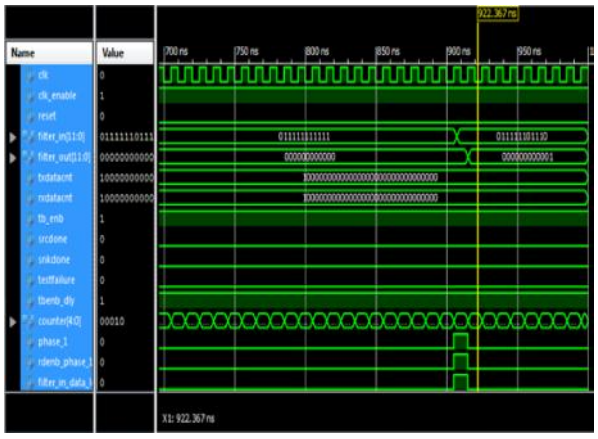


Figure 6: Simulation Results for Fully Parallel Architecture

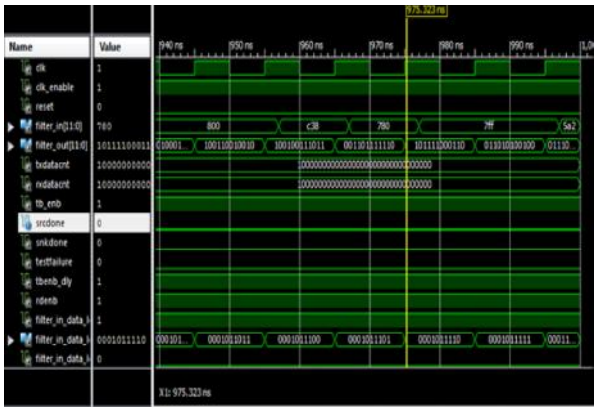
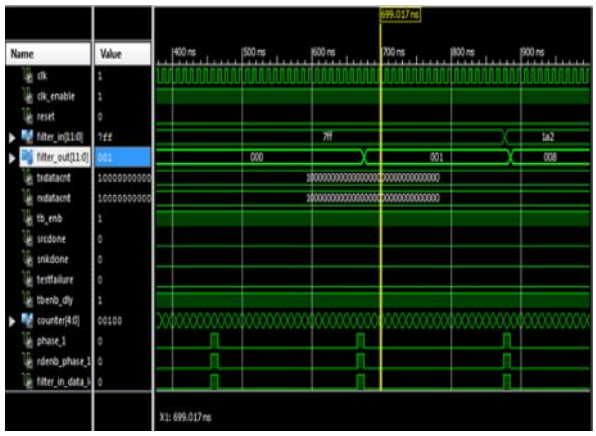


Figure 7: Simulation Results for Cascade Serial



Comparison of Delay Results

Delay results of various architectures are found after simulation and are tabulated in the table 5.1 above. From the table it is inferred that fully parallel is the most efficient architecture in terms of the speed. It has only 0.814 ns of total delay as opposed to a greater delay of fully serial and cascade architectures. That is fully parallel can perform at a faster rate than the other architecture and hence can be called a performance efficient architecture.

Table 1: Delay Results

	Fully Serial (ns)	Fully Parallel (ns)	Cascade Serial (ns)
Logic Delay	1.524	0.35	1.524
Route Delay	1.422	0.464	1.476
Total Delay	2.946	0.814	3.0

Comparison of Slice Registers and Slice LUT Results

Table 2 shows the number of slice registers, LUT's, multipliers, adders/subtractors and RAM's used by each architecture. Slice Registers and Slice LUT results of various architectures are found after simulation and are tabulated in the table above. From the table it

Table 2: Slice Registers and Slice LUT Results

	Fully Serial	Fully Parallel	Cascade Serial
Number of Slice Registers	764	708	798
Number of Slice LUTs	294	213	385
Multipliers	1	29	2
Adders/Subtractors	4	58	6
RAM	1	0	2

can be inferred that fully serial is the best architecture among the three in terms of the area consumption to design an FIR filter as it uses only 1 multiplier and 4 adders/subtractors which is a very meagre as opposed to other architectures and the number of slice registers and slice LUTs are comparable and hence is not a deciding factor for the area. Multipliers approximately weigh twice more than the number of adders/subtractors. Hence, fully parallel consumes the maximum area among the three architectures.

CONCLUSION

We have studied the realisation of FIR filters using various architectures like fully parallel, fully serial, partly serial and cascade serial architectures. The realisation of these architectures is achieved through MATLAB. The filter characteristics for audio specifications are verified using filter design toolbox. Later, by the function called generate hdl the vhdl code for the architectures are generated and is executed to find out the speed versus area performance of each architecture.

Based on the results obtained from the synthesis report it is evident that in terms of speed the partly serial and in terms of area the fully parallel structure gives the best results.

It can be extrapolated from that the speed versus area performance results can be utilised to choose the best available architecture in areas like video conferencing, high performance multimedia DSP structures to improvise the filter performance and utilisation silicon area and bandwidth.

This project has the potential to be realised on a FPGA in future and later the performance of the various different architectures can be studied. Based on the result obtained the best possible architectures in terms of speed Vs area can be used in various applications like multimedia processing, real time signal analysis and conversion in various military and medical applications.

REFERENCES

1. Behrooz Parhami and Ding-Ming Kwai (2003), “Parallel Architectures and Adaptation Algorithmsfor Programmable FIR Digital FiltersWith Fully Pipelined Data and Control Flows”, *Journal of Information Science and Engineering*, Vol. 19, pp. 59-74, Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106-9560, USA.
2. Chao Cheng and Parhi K K (2004), “Hardware Efficient Fast Parallel FIR Filter Structures Based on Iterated Short Convolution”, *IEEE Transactions on Circuits and Systems—I: Regular Papers*, Vol. 51, No. 8, pp. 1492-1500, ISBN: 1057-7122/04.
3. Florian Achleitner, Philipp Miedl and Thomas Unterluggauer (2011), “Filter Structures for FIR Filters”, January 2, Conference Paper, Telematics Graz,

University of Technology, Graz University of Technology.

4. Raveenakoppula and Masthan S K (2011), "An Efficient Distributed Arithmetic Architecture for High Order Digital Filters", *International Journal of Computer Trends and Technology*, Vol. 2, No. 2, Electronics and Communication Engineering Jayamukhi Institute of Technology and Sciences Narsampet, Andhra Pradesh, India.
5. Sirisha A, Balanagu P and Suresh Babu N (2012), "Design of Fir Filter Using Look Up Table and Memory Based Approach", Department of Electronics and Communication Engineering, Chirala Engineering College, Chirala.
6. Yu-Chi Tsao and Ken Choi (2012a), "Area-Efficient VLSI Implementation for Parallel Linear-Phase FIR Digital Filters of Odd Length Based on Fast FIR Algorithm", *IEEE Transactions on Circuits and Systems—II: Express Briefs*, Vol. 59, No. 6, pp. 371-375, ISBN: 1549-7747.
7. Yu-Chi Tsao and Ken Choi (2012b), "Area-Efficient Parallel FIR Digital Filter Structures for Symmetric Convolutions Based on Fast FIR Algorithm", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 20, No. 2, pp. 366-371, ISBN: 1063-8210.
8. Zhenqi Liu, Fan Ye and Junyan Ren (2012), "Low Cost Parallel Fir Digital Filter Structures Utilizing the Coefficient Symmetry", State Key Laboratory of ASIC and System, Fudan University, Shanghai 200433, China, ISBN : 978-1-4673-2475-5/12.