Analysis of different FIR Filter Design Method in terms of Resource Utilization and Response on Field-Programmable Gate Array

Nilesh B. Bosmiya¹ Prof. R. C. Patel²
¹PG Student ²Professor
¹,²Dept. of Instrumentation & Control, L. D. College of Eng. Gujarat, India

Abstract—In this paper fully parallel FIR filters are designed with different design method on FPGA for resource utilization and response analysis. fully parallel band-pass FIR filters with same specification designed and simulated on ISE. The suggested implementations are synthesized with Xilinx ISE 14.2 version. Results show comparison of three different filter design methods in terms of resource utilization.

I. INTRODUCTION

Digital filters are important part of digital signal processing. Before development of FPGA digital filter were implemented on digital signal processor. Digital signal processors are still widely used but they are not capable for high speed application available in present. After the advancement of microelectronic techniques, communication signal processing has come to third generation and forth generation period, so there is a challenge for adaptive processing techniques that the processing speed needs to be high so FPGA based signal processing techniques is mostly used in latest mobile communication, military communication, consumer electronics and aerospace tracking etc so that it is necessary to find the answer of how to increase operation speed of signal processing algorithms and reduce hardware resources by adopting FPGA to implement every kinds of tasks of digital signal processing. So we look forward for design of digital filter with low area and high speed. Benefits of reducing area:

(a) Less power required
(b) Area benefits for other application on same chip
(c) We can use versions of FPGA which have less capability.

Digital filters are typically used to modify or alter the attributes of a signal in the time or frequency domain. The most common digital filter is the linear time-invariant (LTI) filter. An LTI interacts with its input signal through a process called linear convolution, denoted by \( y = f \ast x \) where \( f \) is the filter's impulse response, \( x \) is the input signal, and \( y \) is the convolved output. The linear convolution process is \([1]\) formally defined by:

\[
y(n) = \sum_{k=-\infty}^{\infty} h(k) x(n-k)
\]

(1)\([1]\)

LTI digital filters are generally classified as finite impulse response (i.e., FIR), or infinite impulse response (i.e., IIR). Calculating the constant coefficients of such a digital filter involves considerable amount of computation and this is generally performed using software tools [1].

With available digital filter design software the production of FIR coefficients is a straightforward process. The Filter Design and Analysis (FDA) tool packaged along with MATLAB is such a tool. The double length floating point notation for filter coefficients, used by the FDA tool poses immense challenges in terms of cost and resources, while implementing on an FPGA [1].

The challenge remains is to map the FIR design into a suitable architecture. To overcome this, the filter coefficients have to be quantized to a fixed point notation. The result of coefficient quantization is that the actual implemented transfer function is different from the ideal transfer function. The simplest and most widely used approach to the problem is to round off the optimal infinite precision coefficients to a b-bit representation [1].

II. PARALLEL AND SERIAL ARCHITECTURES

The basic equation for a single-channel FIR filter is shown in equation [1]

\[
y(n) = \sum_{k=0}^{N-1} h(k) x(n - k)
\]

(2)

The terms in the equation can be described as input samples, output samples, and coefficients. Imagine \( x(n) \) as a continuous stream of input samples and \( y(n) \) as a resulting stream (i.e., a filtered stream) of output samples[1].

The \( n \) and \( k \) in the equation correspond to a particular instant in time, so to compute the output sample \( y(n) \) at time \( n \), a group of input samples at \( N \) different points in time, or \( x(n), x(n-1), x(n-2), \ldots x(n-N+1) \) is required[1].

The group of \( N \) input samples are multiplied by \( N \) coefficients and summed together to form the final result \( y(n)\). Fig. 1 shows the logical structure of an FIR Filter[1].
III. FPGA SIMULATION AND RESULT COMPARISON

An FIR Band Pass filter is designed as per the specifications given in Table 1. With three different design methods which are Equiripple, Least-Squares, Least $P$th-norm.

A special class of FIR filter that is particularly effective in meeting the specifications is called the equiripple FIR filter. An equiripple design protocol minimizes the maximal deviations (ripple error) from the ideal transfer function. The filter designed for the mentioned specifications using equiripple design method is of order 64 [1].

Fig. 3 is for the response of the filter with Equiripple design. Fig. 4 is for the response of the filter with Least-Squares design. Fig. 5 is for the response of the filter with Least $P$th-Norm design.

Fully parallel and filter was designed and its behavioural simulation was done using Xilinx ISE 14.2. Resource utilization for different filter design methods is shown in Table 2.
Fig. 6: Waveform screenshot of software simulation for fully parallel design

<table>
<thead>
<tr>
<th></th>
<th>Equiripple</th>
<th>Least-Squares</th>
<th>Least Pth-Norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantization</td>
<td>Q16.14</td>
<td>Q16.14</td>
<td>Q16.14</td>
</tr>
<tr>
<td>Slices</td>
<td>2651</td>
<td>2797</td>
<td>2798</td>
</tr>
<tr>
<td>Slice Flip Flops</td>
<td>1055</td>
<td>1055</td>
<td>1055</td>
</tr>
<tr>
<td>LUTs</td>
<td>4065</td>
<td>4358</td>
<td>4328</td>
</tr>
<tr>
<td>Max. Freq. (MHz)</td>
<td>7.378MHz</td>
<td>7.373MHz</td>
<td>7.548MHz</td>
</tr>
</tbody>
</table>

Table. 2: Resource utilization for different filter design method

Now the Starting portion of simulation result is shown in Fig. 6.

IV. CONCLUSION

We can clearly see that from table II and response figures equiripple design is superior than other two methods.

ACKNOWLEDGEMENTS

Author thanks Prof. R. C. Patel for his valuable guidance for this paper. Author is also thankful to his staff and colleagues for their co-operation.

REFERENCES