Design of Systolic FIR Filter Using VHDL Language

S.Kalpana$^{1}$, P.Samundiswary$^{2}$

$^{1}$M.tech Student, Department of Electronics Engineering, Pondicherry University
$^{2}$Assistant Professor, Department of Electronics Engineering, Pondicherry University
Paducherry, India

Abstract—Low power consumption and smaller area are the most important criteria in VLSI design. This paper presents an efficient design of FIR filter using systolic structure with the consideration of adders and multipliers as processing elements. In this paper, 4, 8,16,32,64 tap Systolic Band Pass FIR Filter with ultra wide band frequency (3.1GHz to 10.6GHz) is designed and simulated using Xilinx tool Integrated Software Environment (ISE)-13.1

Keywords—FIR filter, Systolic filter.

I. INTRODUCTION

Digital Signal Processing (DSP) is widely used in real time applications such as video, image processing and wireless communication. The Finite Impulse Response (FIR) digital filters are widely used in digital signal processing applications due to their stability and linear phase properties [1, 2]. The most important operations in real-time DSP applications are multiplication and addition. The execution speed of multiplication and addition determines the overall performance of digital system and arithmetic functions.

Further multipliers and adders are the key components of high performance system such as FIR filter, microprocessor, digital signal processor [3, 4] etc. A system performance can be determined by the performance of adder and multiplier because they are the processing elements in the system. Furthermore, it is generally most area consuming. Hence optimizing the speed and area of the adder and multiplier is the major issue. As a result, a whole spectrum of adder and multipliers with different area-speed constraints has been designed with parallel operation. The rest of the paper is described as below: Section 1 deals with the introduction to the basic concepts of FIR filter. Section 2 deals with the fundamentals of Systolic FIR filter and its features. In Section 3, the proposed systolic FIR filter structure is discussed. Section 4 deals with the analysis of simulation results of systolic based FIR filter and conclusion is drawn in section 5.

II. SYSTOLIC FIR FILTER

Systolic arrays represent an important architectural paradigm in VLSI signal processing implementations due to the fact that it can be used to efficiently exploit the inherent parallelism embedded in DSP algorithms by pipelining and parallel processing. However the derivation of new efficient systolic algorithms is permanently engaged in order to exploit the inherent parallelism efficiently embedded in such algorithms [5,6]. The way of data moving plays a significant role in the determination of the efficiency of a systolic algorithm and its implementation. This is one of the importance features played by cyclic convolution in digital signal processing. Cyclic convolution provides high computing speed, low computational complexity and I/O cost. Moreover, it can be efficiently implemented through systolic arrays.

A. Systolic Array Architecture

A systolic array is composed of matrix-like rows of Data Processing Units (DPUs) called cells. DPUs are similar to Central Processing Units (CPUs), except for the usual lack of a program counter, since operation is transport-triggered, means by the arrival of a data object. Each cell shares the information with its neighbours immediately after processing. The systolic array is often rectangular where data flows across the array between neighbour DPUs, often with different data flowing in different directions. Figure 1 illustrates the architecture of systolic array [7, 8].
The data streams entering and leaving the ports of the array are generated by Auto-Sequencing Memory (ASM) units. Each ASM includes a data counter. In embedded systems a data stream may also be input from and/or output to an external source. An example of a systolic algorithm might be designed for matrix multiplication. One matrix is fed in a row at a time from the top of the array and is passed down the array; the other matrix is fed in a column at a time from the left hand side of the array and passes from left to right. Dummy values are then passed in until each processor has seen the one whole row and one whole column. At this point the result of the multiplication is stored in the array and can now be output a row or a column at a time, flowing down or across the array.

B. Features of Systolic Array

The features of systolic array are discussed below.

Synchrony:

A systolic array is controlled and synchronized by a global clock with fixed length of clock cycles. Data are rhythmically computed (timed by a clock) and passed through the systolic array network. The clock signal serves two purposes: as a sequence reference and also as a time reference.

- Modularity and regularity:

Modular processing units connected with homogeneous interconnections and the computing network can be extended indefinitely.

- Spatial locality and temporal locality:

The array manifests a locally-communicative interconnection structure, i.e., spatial locality. Each cell or processing elements only communicates with its immediate neighbouring cells. There is at least one unit time delay allotted so that signal transactions from one cell to the next can be completed, (i.e) temporal locality.

- Pipeline ability:

The array exhibits a linear rate pipeline ability to speed up processing rate, i.e., it should achieve an \(O(N)\) speedup, in terms of processing rate, where \(N\) is the number of Processing Elements. Here the efficiency of the array is measured by the following:

\[
speed\ up\ factor = \frac{T_s}{T_p}
\]

Where \(T_s\) is the processing time in a single processor, and \(T_p\) is the processing time in the array processor. The major factors favouring systolic arrays for special purpose processing architectures are simple and regular design, concurrency, communication and balancing computation.

- Simple and regular design:

In integrated-circuit technology, the cost of design grows with the complexity of the system. By using a regular and simple design by exploiting the VLSI technology, great savings in design cost can be achieved. Furthermore, simple and regular systems are likely to be modular and therefore can be adjusted to meet various performance goals.

III. PROPOSED SYSTOLIC FIR FILTER

Systolic design architecture represents an efficient hardware implementation for computational intensive DSP applications because of its features like simplicity, regularity and modularity of structure [9, 10]. In addition, they also possess significant potential to yield high-throughput rate by exploiting high-level of concurrency using pipelining or parallel processing or both. To utilize the advantages of systolic processing, several algorithms and architectures have been suggested for systolization of FIR filters. However, the multipliers in these structures require
a large portion of the chip-area and consequently enforce limitation on the maximum possible number of Processing Elements (PE’s) that can be accommodated and the highest order of the filter that can be realized shown in below figure 1.

![Fig2: Systolic FIR Filter structure](image)

The systolic cell shown in Fig.2 consists of a multiplier and an adder which can be realized by various structures. When the scalability is considered, the bit level systolic structure is adopted so that pipeline registers can be inserted among the bit operation units easily, which remarkably improves the system throughput and meets the different speed requirements.

![Fig3: Systolic array at bit level](image)

As an example, the multiplier input is 4-bits width and the adder’s is 8-bits width, the computing process is shown in Fig.3 (b). The hardware structure is shown in Fig. 3 (c), where HA denotes the half-adder and FA means the full-adder, the critical path consist of two half-adders and four full-adders. If the multiplier input is n-bits width and the adder’s is 2n-bits width, the critical path comprises two half-adders and 2n – 2 full-adders.

IV. SIMULATION RESULTS

Very high speed hardware description language (VHDL) has strong abstract description ability to support hardware design, verification, synthesis and testing. VHDL can describe the same logic function in multiple levels, such as it can describe the structure of the circuit composition in the register level and describe the function and performance of the circuit in the behavioural level. VHDL has been used to implement hardware description of FIR filter and Systolic FIR filter. The filter design has been done in Spartan-6 Platform using ISE 13.1 tools all in one design suit from Xilinx.

This work mainly describes the design and simulation of FIR BPF filter using VHDL language and MATLAB 7.4. Then the delay, power and number of flip flops has been determined for 4, 8, 16, 32 and 64 tap Systolic band pass FIR filters using Xilinx 13.1 tool.

![Figure 4: RTL view of systolic band pass FIR filter with order 4](image)

![Figure 5: Output waveform of 4th order systolic band pass FIR filter](image)
Figure 6 & 5 represents the RTL view and the output waveform for 4th order systolic band pass FIR filter. X_in and Y_out is the input and output of the systolic FIR filter as shown in Figure 5.

Figure 6 and 7 represents the RTL view and the output waveform for 8th order systolic band pass FIR filter with X_in inputs and Y_out as outputs. Figure 8 and 9 represents the RTL view and the output waveform for 16th order systolic band pass FIR filter with X_in inputs and Y_out as outputs.
Figure 13: Output waveform of 64th order systolic Band pass FIR filter

Figure 10 and 11 represents RTL view and the output waveform for 32nd order systolic band pass FIR filter with $X_{in}$ inputs and $Y_{out}$ as outputs. Figure 12 and 13 represents RTL view and the output waveform for 64th order systolic band pass FIR filter with $X_{in}$ inputs and $Y_{out}$ as outputs. Power, Delay, and number of Flip Flops noted for simulated systolic FIR band pass filter are listed in the Table-I.

<table>
<thead>
<tr>
<th>No. of taps</th>
<th>Delay(ns)</th>
<th>Power(w)</th>
<th>No. of flip flops</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>12.066</td>
<td>0.014</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>14.428</td>
<td>0.081</td>
<td>36</td>
</tr>
<tr>
<td>16</td>
<td>15.946</td>
<td>0.114</td>
<td>135</td>
</tr>
<tr>
<td>32</td>
<td>18.682</td>
<td>0.227</td>
<td>284</td>
</tr>
<tr>
<td>64</td>
<td>21.754</td>
<td>0.529</td>
<td>567</td>
</tr>
</tbody>
</table>

From the Table I, the systolic FIR filter has better performance than that of an ordinary FIR filter in terms of speed and power.

V. CONCLUSION

This work mainly describes the design and simulation of Systolic FIR BPF filter for various taps using ISE Xilinx tool. Design of filter coefficients for band pass filter has been performed by using Filter Design and Analysis tool in MATLAB 7.6. These coefficients are converted into binary numbers manually and used to design Systolic FIR BPF filter with the help of VHDL program. From the simulation results, it is observed that systolic band pass FIR filter has better performance than ordinary band pass FIR filter in terms of power and delay. The work can be extended by developing an efficient algorithm for higher order Systolic based FIR filter.

REFERENCES