**Contemporary Engineering Sciences, Vol. 7, 2014, no. 8, 357 - 362**
**HIKARI Ltd, www.m-hikari.com**
**http://dx.doi.org/10.12988/ces.2014.4214**

# Loop Unrolling for Second Order Recursive Digital Filter to Achieve High throughput

**R. Aiswariya and R. Parameshwaran**

SASTRA University, India



**Abstract**

With the quick progression in technology, there is a ton of potential and interest for high speed signal processing particularly to accomplish high throughput required in real-time critical applications. There are various mixture of strategies exist for making strides in the throughput of the Digital system [1]. Parallel processing is one of the strategies which can be given to digital systems for achieving higher throughput simultaneously in the same hardware unit. Main target of this project is to verify parallel processing in second order IIR filter. Three consecutive processing loops are unfolded in the time domain and parallel hardware is implemented [3]. The design can be verified by using Verilog Hardware description Language and the functionality of the filter is verified by the simulations obtained. The corresponding impulse response from the MATLAB verifies the implementation. The critical path is obtained from the timing report of synthesis. The area and power in each case is computed.

**Keywords:** IIR Filter, Throughput, Unfolding, Parallel Processing, CriticalPath

## 1. Introduction

The parallel processing is a transformation technique where multiple instructions are overlapped in execution. But, in case of recursive filters like IIR,

normal replication of hardware cannot be used as the filter response of one unit depends on the response output of other units [4]. Therefore, some special techniques need to be developed. In this project, the units like Single Input Parallel Output, Multiple Input Multiple Output, and Parallel Input Single Output will be discussed, which are used for implementing parallel processing in IIR filter [3].

## 2. Single Input Parallel Output (SIPO)

Parallel processing systems are innately multiple input and multiple output systems because the data are processed parallel in them. Hence, if parallel processing technique is applied to single input single output systems, they should be given to multiple input systems using single input parallel output unit. The data samples are taken at a fast clock which is nearly L times the clock on which the parallel processing system operates. The fast clock shifts the data at a fast rate[5]. In this Figure, the unit takes data at a sampling period of T/4 and shifts the data into the shift register.

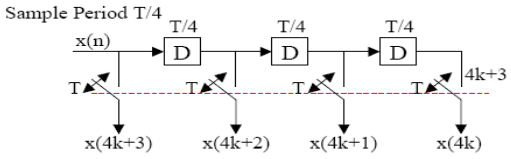



**Figure 1. Serial Input Parallel Output**

## 3. Parallel Input Single Output (PISO)

In order to convert a multiple output system to single output system, Parallel Input Single Output is used as shown in the Figure.2. Here, the data are taken in parallel and outputs as single bit data.

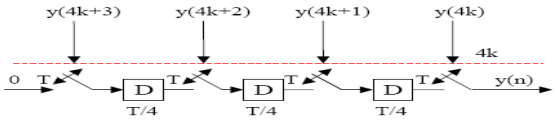



**Figure 2.Parallel Input Serial Output**

## 4. Multiple Input Multiple Output (MIMO)

In a single clock period, multiple samples of data are given as input to the MIMO system and multiple outputs are taken out. Here parallel hardware is used to process the multiple samples. An example of this system is given in the Figure.3 Here, at the k clock cycle, inputs x (3k), x (3k+1) and x (3k+2) are processed and 3 samples y (3k), y (3k+1), y (3k+2) are generated at the output.

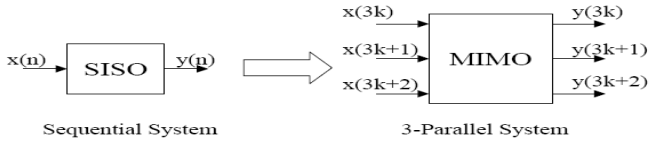
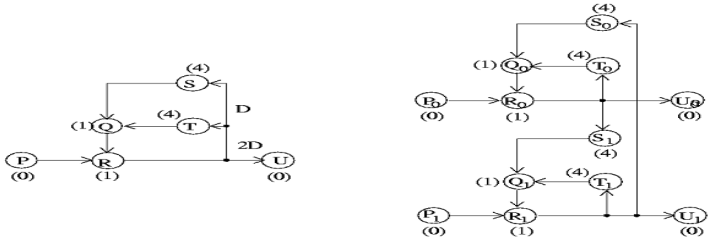



**Figure 3.Multiple Input Multiple Output**

## 5. Unfolding or Loop Unrolling Of IIR Filter

As the IIR filters have feedback path, it cannot be straightly converted to parallel processing systems by hardware replication. The only way to achieve parallel processing is unfolding. Unfolding is a technique in which processing loops of a filter is unrolled in hardware [7]. An example for unfolding is shown in Figure.4




(a) (b)

**Figure 4.Unfolding for a Feedback System**

On the left hand side of the Figure.4, we have single input single output filter. But, in the right hand side, the unfolded system takes in two inputs and processes them together. The first sample is processed in the block marked with subscript '1' to process the second sample. The feedback is unrolled by giving the output of the first sample into the next sample [1].

## 6. Implementation of Parallel Processing

Parallel processing was applied to second order IIR filter with a block size of L=3. Here, a one to three serial input parallel output converter is used at its input stage. The multiple input multiple output system in our implementation takes in 3 inputs simultaneously. The implementation is shown in Figure 5.
The transfer function of the given filter is,

$$H(Z) = \frac{\frac{1}{2} + \frac{1}{4}Z^{-1} + \frac{1}{16}Z^{-2}}{1 - \frac{1}{8}Z^{-1} - \frac{1}{16}Z^{-2}} \quad \rightarrow (1)$$

The single input single output form of the filter is,

$$y(n) = (1/2)*x(n) + (1/4)*x(n-1) + (1(16)*x(n-2) + (1/8)*y(n-1) + (1/16)*y(n-2) \quad \rightarrow (2)$$

After unfolding, the filter can be described as the following equations,

$$y(3k) = (1/2)*x(3k) + (1/4)*x(3k-1) + (1/16)*x(3k-2) + (1/8)*y(3k-1) + (1/16)*\ y(3k-2) \quad \rightarrow (3)$$

$$y(3k+1) = (1/2)*x(3k+1) + (1/4)*x(3k) + (1/16)*x(3k-1) + (1/8)*y(3k) + (1/16)*y(3k-1) \quad \rightarrow (4)$$

$$y(3k+2) = (1/2)*x(3k+2) + (1/4)*x(3k+1) + (1/16)*x(3k) + (1/8)*y(3k+1) + (1/16)*y(3k) \quad \rightarrow (5)$$

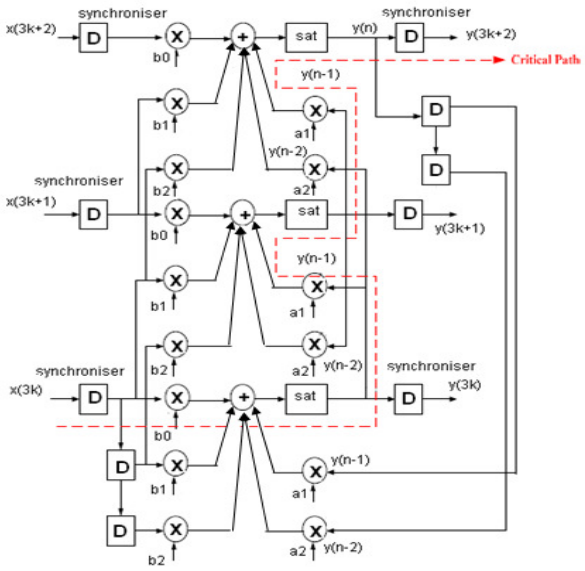



**Figure 6. Parallel Processing (L=3)**

In order to implement these equations, the filter hardware must be replicated for three times and filter loops must be unrolled by three successive operations as seen in the equations above.

## 7. Methodology

The design flow starts by taking the coefficients of the IIR. The coefficients and bit width informations are generated by the MATLAB by generating two files define.v and coeff_val.v. These files are included in the main verilog code. The functionality of the filter is verified by comparing the impulse response from MATLAB with the corresponding Verilog output. All verilog simulations were done by using Modelsim. Synthesis reports are taken from synopsis compiler.

## 8. Result Analysis

### 8.1 Simulation Result

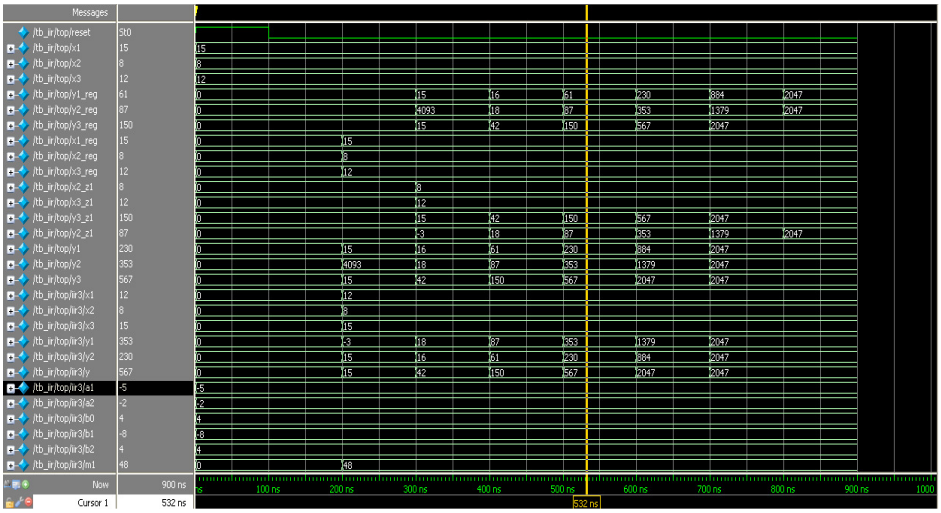



**Figure 7. Simulation result of IIR filter using Parallel Processing**

The x1,x2,x3 are the input variables and y1,y2,y3 are the output variables of the parallel processed IIR. The a1,a2,b0,b1,b2 are the coefficient vectors.

### 8.2. Comparison table of Area, Timing, Power report for SISO and MIMO

| PARAMETERS | SISO | MIMO |
|---|---|---|
| AREA | 2241.931034 $\mu m^2$ | 7568.875528 $\mu m^2$ |
| POWER | 2.9758 mw | 0.92027 mw |
| DATA ARRIVAL TIME | 9.58ns | 0.33ns |

## 9. Conclusion

In this project, a second order IIR filter was verified with parallel processing technique-MIMO. From the above table, the timing improvement is achieved over the normal filter but the area is increased. The frequency response is obtained from MATLAB. Then the filter was designed in verilog using parallel processing with the block size of L=3 and the simulation results were obtained. It can be used in all DSP system and communication system as it produces a maximum throughput compared to other techniques [6].

## References

[1] C. Ifeachor and sW. Jervis, Digital Signal Processing – A Practical Approach, PearsonEducation, 2002.
[2] H. Kaeslin, Digital Integrated Circuit Design: from VLSI Architectures to CMOS Fabrication, Cambridge University Press, 2008.
[3] K. Parhi, VLSI Digital Signal Processing Systems: Design and Implementation, John Wiley& Sons, Inc., 1999
[4] C. H. Papadimitriou and K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Dover Publications, Inc., 1998.
[5]. Naveed A. Sherwani, Algorithms for VLSI Physical Design Automation, Kluwer AcademicPublishers, 1999.
[6] Shaw.A.K. and Imtiaz.M, (1996) 'A general Look-Ahead algorithm for pipelining IIR filters', in Proc. IEEE ISCAS, pp. 237-240.
[7] S. M. Sait and H. Youssef, VLSI Physical Design Automation: Theory and Practice, World Scientific, 1999.

**Received: February 1, 2014**