International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 19 (2018) pp. 14473-14478

Realization of FIR Filter Architecture in Transpose Form Type-II Configuration for Reconfigurable Applications

A. Madan Mohan Reddy[1], D. Lalitha Kumari[2]

[1]PG scholar, [2]Assistant Professor,
[1, 2] Department of Electronics and Communication Engineering, JNTU, Anantapur, Andhra Pradesh, India.

Abstract

The objective of this design to implement Finite impulse response filter architecture in a transpose-form with type-II configuration. This FIR filter is a pipelined architecture and helps a couple of regular multiplications that save computational effort. Transpose-form of filters doesn't directly support block processing technique, no longer like direct-form configuration. Depending on the comprehensive computation evaluation of transpose-form FIR filter, to obtained flow-graph of FIR filter with customized architecture. In implemented FIR filter architecture we utilize 16x8 array multipliers and 24-bit ripple carry adders. The architecture for a transpose type-II form FIR filter involves 11% (63.71 MHz) higher sampling frequency, 2 to 3% less ADP (6195 sq.µm.ns) and less 2%(1.16 n.J) EPS compared to existing direct-form FIR filter for medium and large length filters. In further increases sampling frequency in the implementing architecture, we utilize 24-bit carry look ahead adders for increasing the 15% (92.64 M.Hz) sampling frequency than the implemented FIR filter architecture with ripple carry adders. These FIR filter architectures are designed with the utilization of 90-nm technology library, and it has been implemented utilizing the EDA tool cadence.

Keywords: Block processing, reconfigurable applications, Fine impulse response, type-II configuration, Area-delay Product (ADP), Energy per sample (EPS).

INTRODUCTION

Reconfigurable hardwares have many benefits noted by using David Anderson & ErhanOzalevli over fixed or dedicated hardwares. A dedicated hardware assets high overall performance with low electricity consumption, nevertheless. They lack in lower back of the reconfigurable hardware assets as flexibility and reusability. Many reconfigurable component's systems can be acquired to operate the mandatory signal handling. Each has its very own strengths and weaknesses with regards to performance, programmability and power consumption. The designer want to make trade-offs in their design areas when figuring out the first-rate digital hardware remedy for a signal-processing application. Reconfigurable hardware system includes DSP, ASIC, and FPGA. Selection of a reconfigurable program varies from utility to utility based totally upon the factors just like speed, efficiency, programmability and flexibility in the design.

Block-processing is customarily an approach that is utilized to accumulate high-throughput hardware constructions. It not genuinely just provides throughput-scalable diagram but supplementally enhances the area-delay efficiency. The derivative of the block-based FIR filter structure is normally simple in the direct form configuration, while the configuration of the transposed structure does not directly enhance block processing. However, we consider the computational benefits of these multiple constant multiplications. These transpose types of constructions are inherently pipelined and avail to provide a higher operating frequency to encourage a higher sampling rate.

LITERATURE SURVEY

A reconfigurable digital FIR filter presented via J Park and W Jeony [2] is predicated on a computation sharing multiplier (CSHM) technique. The CSHM approach concretely targets for reduction in the redundant computation by the computation re-use in vector products that can be efficiently utilized for low power applications. As a result of utilization of computation sharing multiplier, a rate of this operation is significantly decremented. Further advancement in the sophistication, a way of implementing low complexity FIR filters, which makes utilization of binary common sub-expression elimination (BCSE) methods were once developed via Smitha & Vinod [3]. Within this function, the sophistication of FIR filters has been dominated by way of the number of adders (sub-tractors) to execute the coefficient multipliers and has been consummated through utilizing BCSE system predicted on canonic signed digit (CSD) representation, which extensively reduces the number of adders from coefficient multipliers. The rate of this operation is an adscititiously limited via way of programmable shifters and CSD conversion.

Pramod Kumar Meher and Chandrasekaran [4] developed the diagram optimization of a single - and - two-dimensional absolutely pipelined computing preparations for its area-delay & power-efficient execution of FIR filter via the systolic decomposition of both DA set up inner-product computation. The systolic decomposition is found to provide a flexible resoluteness of the tackle span of the search tables for DA-predicated computation to cull congruous place & time tradeoff. It's located that the utilization of more diminutive address spans for DA-predicated calculating units, it's probably to decrement the recollection size, on the other hand on the flip side it provides to the magnification of adder simplicity and the latency.

Kuan-Hung Chen & Tzi-Dar Chiueh [5] have introduced the apperception of low-power digit-predicated reconfigurable FIR filter with a fine granularity. It offers a flexible yet compact and low-power remedy to FIR filters with a wide assortment of precision and tap length. This architecture withal provides scalability, modularity, and cascadability, thus making it conducive to VLSI implementation. However, this architecture achieves low power with the reduced intricacy of filter structure being incremented because of reconfiguration. To be able to enhance the decremented intricacy, two pristinely incipient reconfigurable BCSE architectures for FIR filters with low involution have grown by Mahesh & Vinod [6] and Asgar Abbaszadeh & Khosrov Sadeghipour [7]. Here filter coefficients are partitioned to smaller sub coefficients according to binary expression technique.

Pramod Kumar Meher [8] introduced an incipient strategy for appearance Lookup Table is an area-efficient alternate to a DA-predicated layout of FIR filter with precisely the equivalent throughput of execution. An operand and internal product decompositions, the layout involved traditional LUT-multiplier-predicated and DA-predicated structures such as FIR filter of equal throughput respectively, in which The LUT-multiplier-predicated design necessitated diminished quantity of input at the Price of marginally more preponderant Adder-widths compared to precedent layouts. It is revealed that the suggested LUT multiplier-predicated design injunctively authorized proximately less Area of 15 percent compared with the DA-predicated layout for the same throughput and lower latency of execution.

IMPLEMENTING ARCHITECTURE

In the Implementing, architecture overcomes the drawback of the direct-form Fir filter. We utilize a transpose from configuration Fir filter and pipelining technique. The advantage of pipelining is that it increments the throughput of the FIR filter. The proposed method divided into two sections.

1. Mathematical & Block formulation for transpose form type-II configuration FIR filter.
2. Design an architecture for transpose form type-II FIR filter for reconfigurable applications.

1. Mathematical & Block Formulation of the Transpose Form type-II configuration FIR Filter

The every clock cycle FIR filter receives L inputs and processes to produce L outputs. The FIR filter output A_k is

$$A_k = B_k \cdot C \quad \ldots. (1)$$

Where C as

$$C = [c(0), c(1), \ldots\ldots\ldots\ldots, c(N-1)]^T.$$

The input B_k is defined as

$$B_k = [b_k^0\ b_k^1 \ldots\ldots\ b_k^4 \ldots\ldots b_k^{N-1}] \quad \ldots.. (2)$$

Where b_k^i is the (i+1)th column of B_k are defined as

$$b_k^i = [b(kL-i)\ b(kL-i-1)\ \ldots\ldots b(kL-i-L+1)]^T \quad (3)$$

Substituting (2) in (1)

$$a_k = \sum_{i=0}^{N-1} b_k^i \cdot c(i) \quad \ldots\ldots\ldots (4)$$

Consider N is a composite number so we decomposed into N=ML, then i=l+mL, for $0 \le l \le L-1$, and $0 \le m \le M-1$ in (3), we have

$$b_k^{l+mL} = b_{k-m}^l \quad \ldots\ldots\ldots\ldots (5)$$

Substituting (5) in (2) we have

$$b_k = [\, b_k^0\ b_k^1 \ldots b_k^{L-1}\ b_{k-1}^0\ b_{k-1}^1 \ldots b_{k-1}^{L-1}\ \ldots\ldots b_{k-M+1}^0\ b_{k-M+1}^1 \ldots b_{k-M+1}^{L-1}] \quad \ldots (6)$$

Substituting (6) in (1), we have

$$a_k = \sum_{l=0}^{L-1} \sum_{m=0}^{M-1} b_{k-m}^l \ .\ c(l+mL) \quad \ldots\ldots (7)$$

The input B_k of (7) into M small matrices s_k^l and the coefficient vector C is also decomposed into small weight vectors

$$h_m = \{h(mL), h(mL+1), \ldots. h(mL+L-1)\}.$$

Interestingly, s_k^m is symmetric and satisfy the following identity

$$s_k^m = s_{k-m}^0. \quad \ldots\ldots\ldots\ldots (8)$$

Computation (8) expressed in the matrix-vector product using s_{k-m}^0 and h_m as

$$a_k = \sum_{m=0}^{M-1} r_k^m \quad \ldots\ldots (9)$$

$$r_k^m = s_{k-m}^0 . h_m$$

The computation of (9) may be expressed in recurrence form

$$A(Z) = s^0(z)[z^{-1}(\ldots(z^{-1}(z^{-1}h_{M-1} + h_{M-2}) + \quad h_{M-3}) + \cdots) + h_1) + h_0] \quad \ldots (10)$$

Where $S^0(Z)$ and A (Z) is z-domain representation of $S^0(k)$ and A(k).

2. Design an architecture for transpose form FIR filter for reconfigurable applications.

The implementing architecture for a block FIR filter is based on the recurrence relation [10] shown in Fig 1 for block size L=4 and N=16.

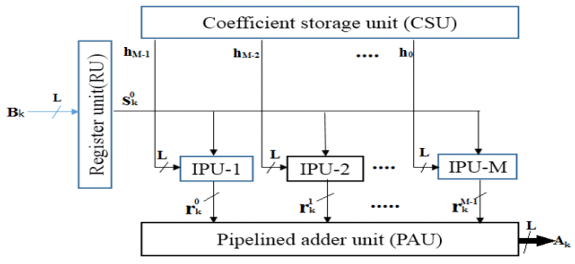

Figure 1. The implementing architecture for transpose-form block FIR filter with L=4.

2.1. Coefficient storage Unit (CSU)

The CSU stores all filter's coefficients to be utilized for the reprogrammable filter purposes that can be employed via utilizing N ROM LUTs, such that any certain channel filter coefficients are received in one clock cycle where N is a length of the filter. The inner structure of CSU show in below Fig 2.

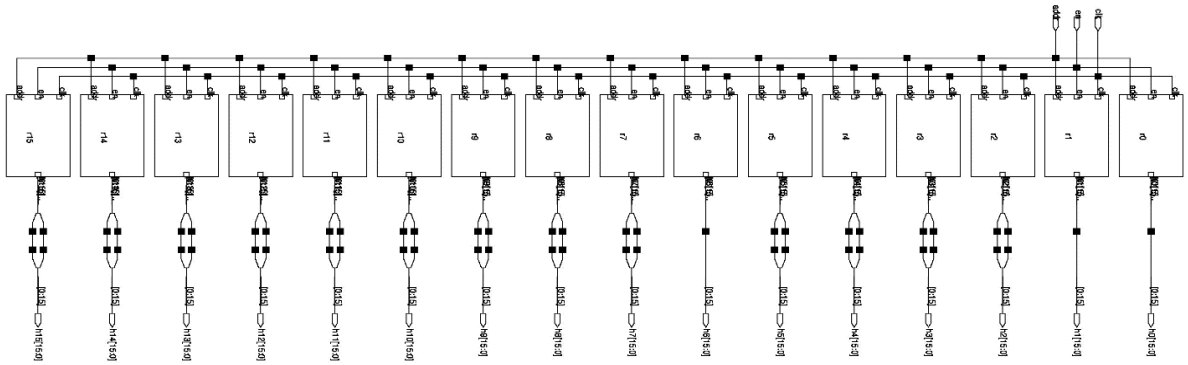

Figure 2: Structure of CSU.

2.2. Register Unit (RU)

The register unit (RU) takes the input samples from A_k during a k_{th} cycle and produces the output samples of L lines of s_k^0 in parallel. The RU unit consists of four registers each register as 8-bits wide that means each register eight flip-flops. The construction of RU unit shown in below Fig 3.

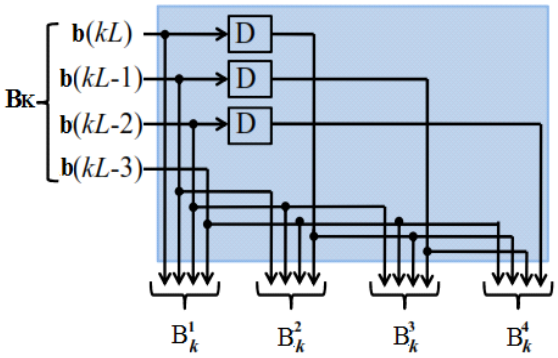

Figure 3. Structure of RU.

2.3. IPU Unit

The inter-product unit (IPU) receives L line of s_k^0 and M short vectors from the CSU unit, each IPU matrix-vector product of s_k^0 with short vector h_m and produces L partial filter output (r_k^m). The construction of the IPU unit shown in below Fig 4. Each IPU unit consists of four IPC units.

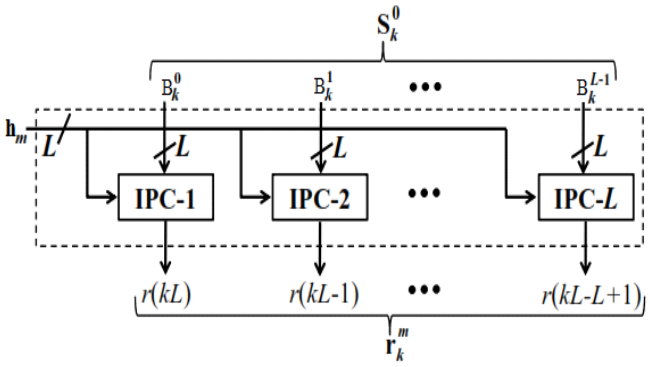

Figure 4. Structure of $(m + 1)_{th}$ IPU.

2.4. IPC Unit

The IPC unit consists of 4 multipliers and 3 adders, here the inputs of multipliers are L lines of s_k^0 from RU unit and are multiples with the short vector of h_m from the CSU unit, and the outputs of the multipliers integrate with adders engenders the output r(KL-1). The structure of an IPC unit shown in below Fig 5, in the IPC we utilize 16X8 array multipliers and the structure shown in Fig 6(a), and 24-bit ripple carry adders and the diagram shown in below Fig 6(b).

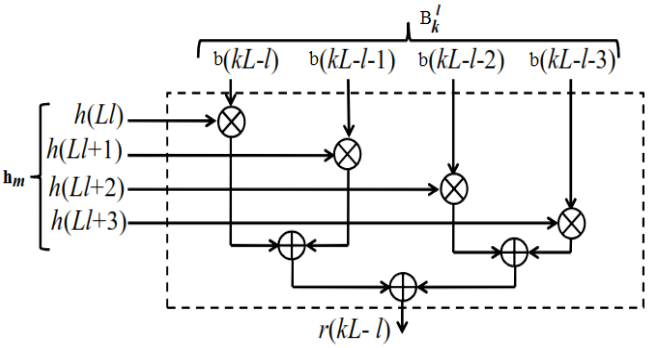

Figure 5: Structure of IPC Unit.

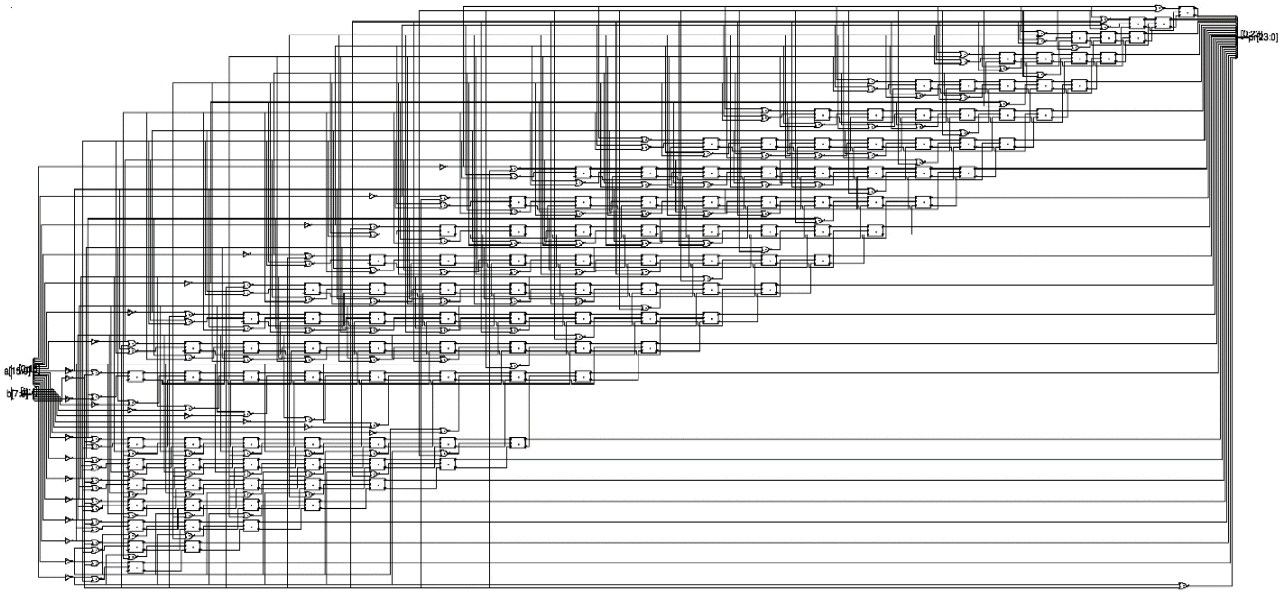

Figure 6(a): 16X8 array multiplier.

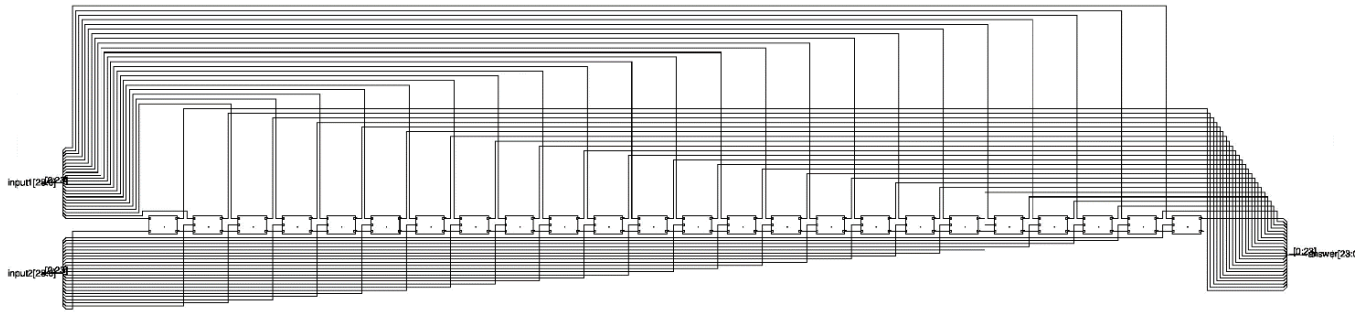

Figure 6(b): 24-bit ripple carry adder**.**

2.5. Pipeline adder unit (PAU)

These partial inner product's outputs r_m are added to the PAU unit shown in below Fig 7 and get output A_k. The PAU unit consists of adders and registers each register as 24-bit wide.

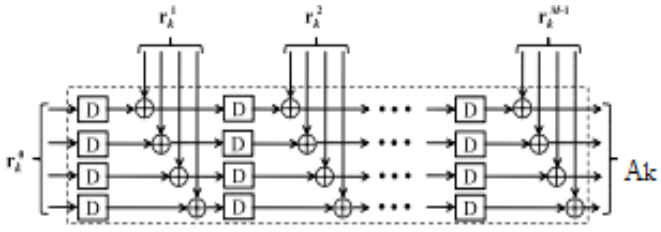

Figure 7. Structure of PAU.

2.6. Carry Look Ahead Adder

In further increasing sampling frequency in implementing architecture we use carry look ahead (CLA) adders, this adder's to help to increases the sample frequency of Fir filter architecture because of this adders very fast compared to ripple carry adders. The basic 4-bit carry look ahead diagram shown in below Fig 8, by using 4-bit CLA we design 24-bit CLA.

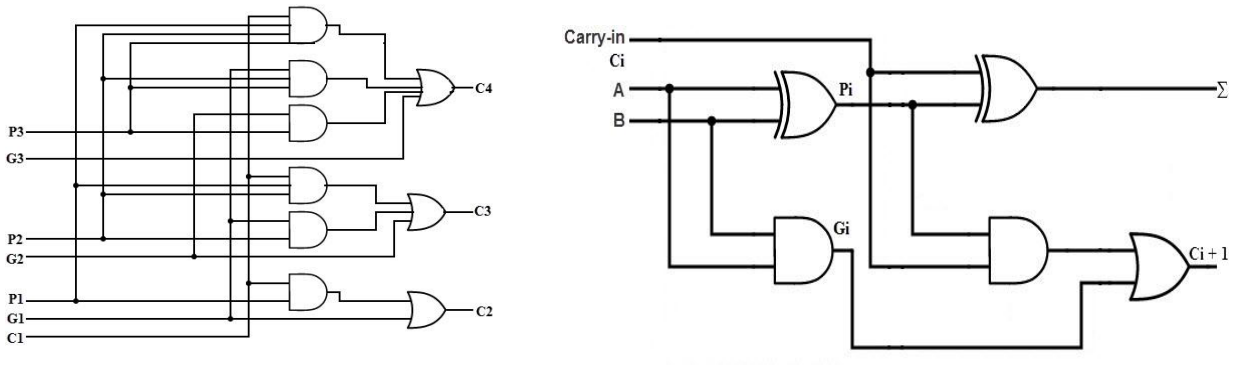

Figure 8: 4-bit Carry Look Ahead adder.

SYNTHESIS & SIMULATION RESULTS

We have coded the architecture in VERILOG for filter length 16, and block size 4 in additionally, we have coded the direct-form FIR architecture extracted from [9] for same filter length and equal block size, same multipliers and adders, We have to consider an input pattern is 8-bit word length, coefficients are 16-bit word length and 24-bit word length for the intermediate and the output signals of each design. All the designs are synthesized by the utilization of Cadences Design Compiler with a 90-nm library. The cadences synthesis results are shown in Table I for comparison. The comparisons of sampling frequency shown in Table II.

Table I: Synthesis Results of Implementing Structure and Existing Structure.

Structure	N (Filter length)	MCP (ns) Or Delay	ADP (Sq.µm.ns)	EPS (n.J)
Direct-form structure of [9] (L = 4)	16	7.5	241901	20.52
Implementing Structure (L =4)	16	6.7	235706	19.36

N: Filter Length MCP: Minimum clock Period

Table II. Comparison of sampling frequency's with various architectures.

Structure	N (Filter Length)	MCP(n.s) Or Delay	Sampling Frequency (M.hz)
Direct-form structure of [9] (L=4)	16	7.5	533.3
Implementing method using ripple carry adders (L=4)	16	6.7	597.01
Implementing method using carry look ahead adders (L=4)	16	5.8	689.65

- ADP= area/sample frequency.
- EPS=power/sample frequency.
- Sample frequency=block-size/MCP.

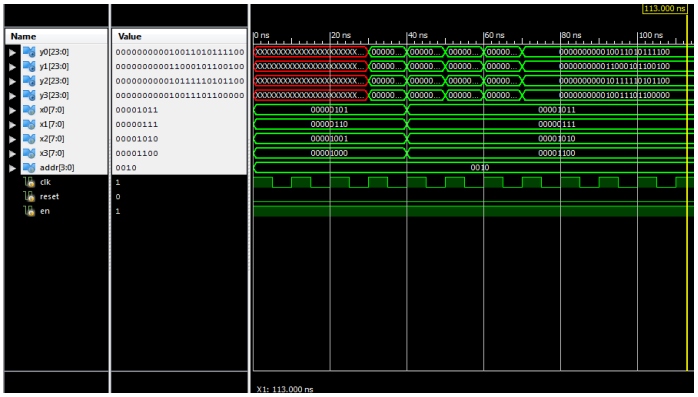

Figure 9: Simulation results for Implementing FIR filter architecture.

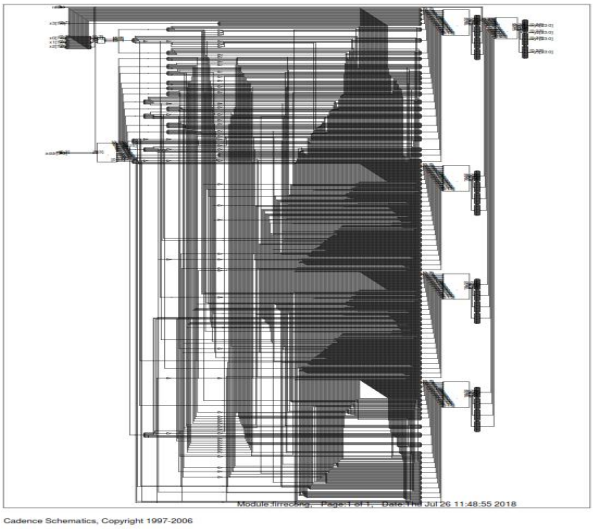

Figure 10. Cadences Synthesis block diagram for implementing architecture.

CONCLUSION

In this paper, we have explored the cognizance of FIR filters in transpose type-II configuration for area-delay efficient cognizance of reconfigurable applications. We have considered Application-specific integrated circuit synthesis results shows

11% (63.71 M.Hz) increases in sampling frequency, 2% to 3% less ADP (6195 sq.µm.ns) & 2% to 3% less EPS (1.16 n.J) to change in implemented structure for L=4 and N=16. In further replace ripple carry adders with carry look ahead adders we increase 15% (92.64 M.Hz) of sampling frequency compared to implementing method in ripple carry adders.

REFERENCES

[1]. B. K. Mohanty and P. K. Meher, "A High-Performance FIR Filter Architecture for Fixed and Reconfigurable Applications," IEEE Trans. very large scale integration (VLSI) systems. vol. 24, no. 2, pp. 444-452, Feb. 2016.

[2]. M. Meimand, Park, Wang, Jeong, Roy, and Choo, "Computation sharing programmable FIR filter for low-power and high-performance applications," IEEE J. Solid-State Circuits, no. 2, vol. 39, Feb. 2004, pp. 348-357.

[3]. P. Vinod and G. Smitha, "A new binary common sub expression elimination method for implementing low complexity FIR filters," July 2007, pp. 2327-2330.

[4]. P .K Meher, Chandrasekaran, and Amira, "FPGA realization of FIR filters by efficient and flexible systemization using distribute arithmetic," IEEE Trans. Signal Process, no. 7, vol. 56, Jul. 2008, pp. 3009–3017.

[5]. Chiueh, and Chen "A low-power digit-based reconfigurable FIR filter," IEEE Trans. Circuits Syst. II, Exp. Briefs, no. 8, vol. 53, , Aug. 2006, pp. 617–621.

[6]. Mahesh and P. Vinod, "New reconfigurable architectures for implementing FIR filters with low complexity," IEEE Trans, no. 2, vol. 29, Feb. 2010, pp. 275–288.

[7]. Sadeghipour.Asgar Abbaszadeh and Khosrov,"A new hardware efficient reconfigurable fir filter architecture suitable for FPGA applications," IEEE 17th International Conference on Digital Signal Processing (DSP), Aug 2011, pp. 1-4.

[8]. P. K Meher, "New approach to look-up-table design and memory based realization of FIR digital filter," IEEE Trans. Circuits Syst. I, Reg. Papers, no. 3, vol. 57 Mar. 2010, pp. 592–603.

[9]. K Mohanty and P. K Meher, "A high-performance energy-efficient architecture for FIR adaptive filter based on new distributed arithmetic formulation of block LMS algorithm," IEEE Trans. Signal Process, no. 4, vol. 61, Feb. 2013, pp. 921–932.