

Original Article

Design and Implementation of High Throughput and Efficient FIR Filter Architectures using Unfolding

Braj Kishor¹, Krishna Raj²

^{1,2}Department of Electronics Engineering Department, Harcourt Butler Technical University, Kanpur, Uttar Pradesh, India.

¹Corresponding Author : brajkishoragra@gmail.com

Received: 25 April 2024

Revised: 11 July 2024

Accepted: 31 July 2024

Published: 28 August 2024

Abstract - The unfolding technique can be used to generate word-level parallel processing architectures as it uses fewer clock cycles to compute an output sample. This paper aims to design and implement word-level parallel processing architectures for 2-tap, 4-tap and 11-tap serial low-pass FIR (Finite Impulse Response) filters. To improve the throughput of the proposed architectures, data broadcast types serial FIR filters were used. An unfolding factor with the value of two is taken for designing 2-unfolded architectures for the original serial FIR filter architectures. Proposed 2-unfolded architectures operate at 1200 KHz, and inputs for 2-unfolded architectures are generated by a serial-to-parallel converter circuit at a frequency of 2400KHz. FPGA (Field Programmable Gate Array) system clock is used to generate these frequencies. VHDL (Very High Speed Integrated Circuit HDL) language is used to design 2- unfolded architectures, and Xilinx Vivado 2015.2 tool is used to implement these architectures on the Artix7 Basys3 FPGA board. Designed and implemented 2-unfolded architectures are compared with existing serial and parallel FIR filter architectures.

Keywords - FIR filter, Unfolding, Parallel processing, FPGA, VHDL, Throughput.

1. Introduction

The unfolding technique can be used to modify the original DSP (Digital Signal Processing) program to create a new program that describes more than one iteration of the original DSP program [1]. Unfolding can be used in compiler theory [2]. In the unfolding technique, hardware is replicated to process several inputs and outputs at the same time. High speed and low-power VLSI (Very Large Scale Integration) architectures can be designed by using unfolding techniques [3-5]. To increase the speed of the 3-tap FIR filter, Shilpa Thakral and Divya Goswami et al. [13] have designed a 3-tap FIR filter using the unfolding Technique. An example of the unfolding technique for the original DSP program

$$y(n) = ay(n - 7) + x(n) \text{ for } n = 0 \text{ to } \infty \quad (1)$$

is shown in Figure 1. In this figure, the unfolding factor (J) value is two.

$$y(2k) = ay(2k - 7) + x(2k)$$

$$y(2k + 1) = ay(2k - 6) + x(2k + 1) \quad (2)$$

Equation (2) represents two consecutive iterations of Equation (1). By putting $n=2k$ and $n=2k+1$ in equation (1) 2-unfolded DSP program is created. The unfolding algorithm is as follows: For a given original DSP program, first draw its DFG (Data Flow Graph). Then draw J nodes U_0, U_1, \dots, U_{J-1} for each node U in the constructed DFG. Then, in the last step, draw J edges $U_i \rightarrow V_{(i+w)\%J}$ with $[(i+w)/J]$ delays for $i=0,1,2,\dots,J-1$ for each edge $U \rightarrow V$ with w delay in the constructed DFG. Several researchers have used unfolding techniques to design

high speed and low power VLSI architectures. Manoj Kumar and Karni Ram [6] have used an unfolding algorithm to design 2-parallel and 3-parallel architectures for FIR and IIR (Infinite Impulse Response) filters. Xilinx Vivado 2016.2 tool is used for synthesis and simulation. Manikandan S.K et al. [7] 2013 used the unfolding technique to design high throughput LFSR (Linear Feedback Shift Register) for Bose-Chaudhuri-Hocquenghem (BCH) Encoder. Unfolding factors with values 2 and 3 are taken for constructing 2-parallel and 3-parallel LFSR architectures.

They have used VHDL (VHSIC Hardware Description Language) language to design unfolded LFSR architectures. Xilinx 9.2i tool is used for simulation and implementation. Sangeeta Singh et al. [8] presented the implementation of parallel CRC (Cyclic Redundancy Check) by using the unfolding technique. They have used Verilog HDL language to design parallel CRC architecture. The cadence tool is used for synthesis, and the Xilinx ISE tool is used for simulation. The rest of this paper is structured as follows. In this paper, high speed and efficient parallel FIR filter architectures are designed by using the unfolding technique and compared with existing serial and parallel FIR filter architectures for area (in terms of LUTs) and speed. The organization of this paper is as follows. The proposed unfolded FIR filter structures are discussed in Section 2. Experimental results and synthesis results comparison among FIR filters are discussed in Section 3. Section 4 deals with the conclusion of the paper.



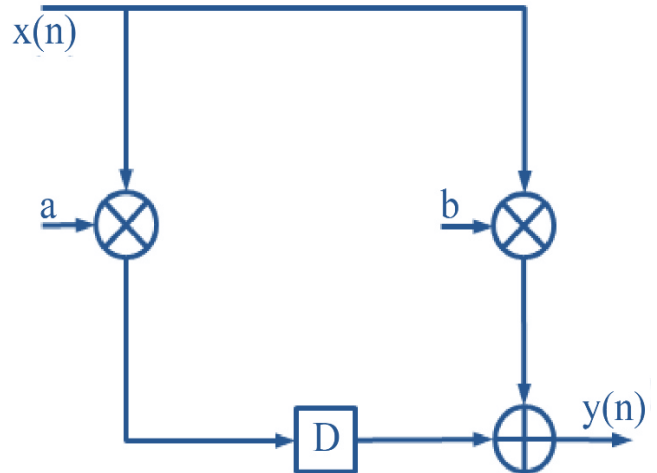
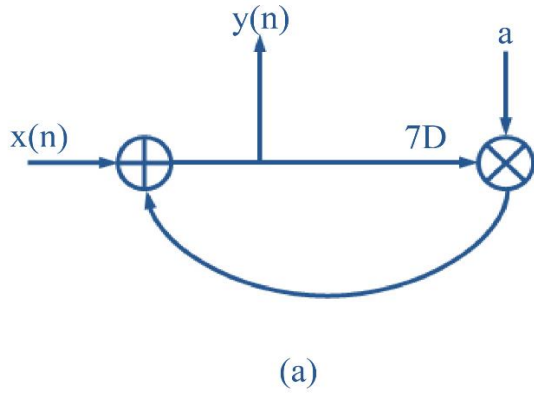


Fig. 2-tap FIR filter architecture

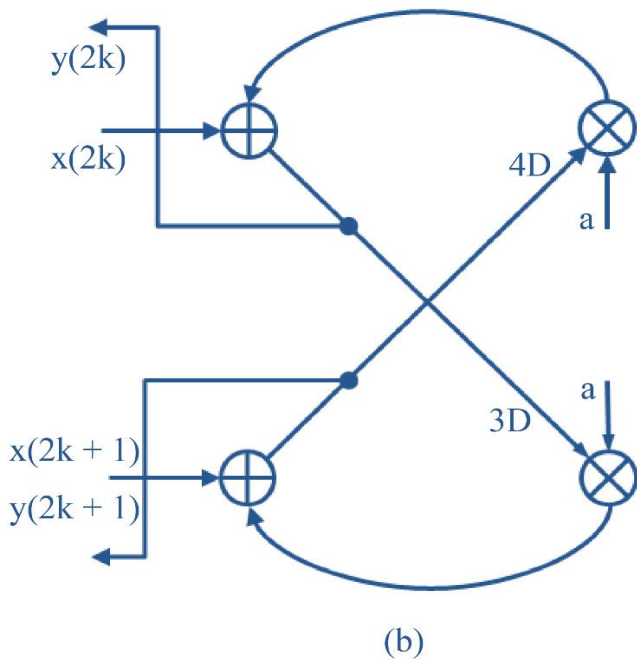


Fig. 1 (a) The original DSP program, (b) The 2-Unfolded DSP program

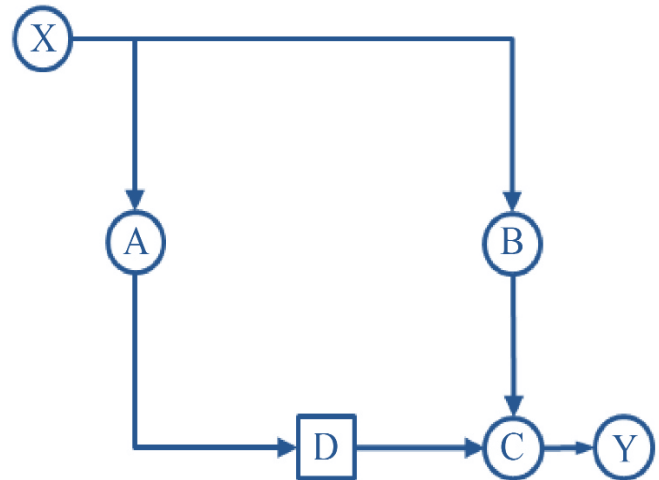


Fig. 3 The DFG for the 2-tap FIR filter architecture

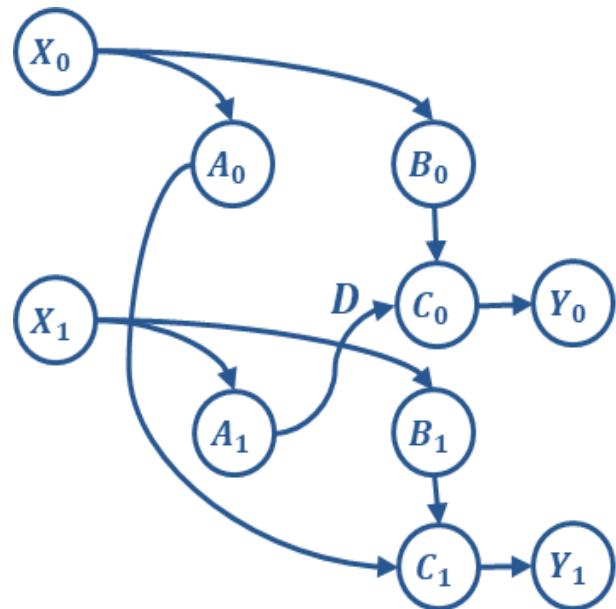


Fig. 4 The unfolded DFG diagram for the tap value 2

2. Details of Proposed Unfolded FIR Filter Architectures

Data broadcast types FIR filter architectures are considered for designing high throughput FIR filter architectures. To generate world level parallel processing architecture by using the unfolding technique, 2-tap, 4-tap, and 11-tap FIR filters of broadcast type structures have been proposed in this paper. The 2-tap broadcast type word serial FIR filter architecture is shown in Figure 2.

The DFG of the original 2-tap FIR filter architecture is shown in Figure 3. To unfold Figure 3 by unfolding factor two, the ten nodes X_i , A_i , B_i , C_i , and Y_i for $i=0,1$ are first drawn. Then second step of the unfolding algorithm is applied to construct edges. The 2-unfolded DFG for the 2-tap FIR filter architecture is shown in Figure 4.

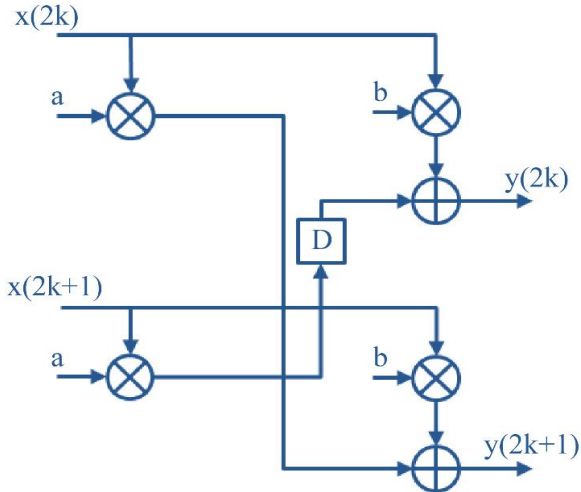


Fig. 5 Parallel FIR filter architecture for 2-tap FIR filter

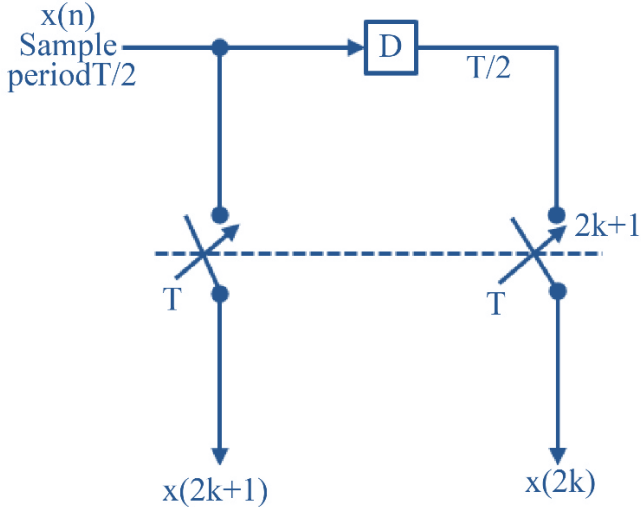


Fig. 6 Serial-to-parallel converter circuit

2-Parallel FIR filter architecture is constructed by applying the unfolding technique in the original word serial 2-tap FIR filter architecture. Figure 5 represents the 2-parallel FIR filter architecture for Figure 2. For generating two input samples for a 2-parallel FIR filter architecture a serial-to-parallel converter circuit is used. In Figure 5, the 2-parallel FIR filter architecture processes two input samples per clock cycle to generate two output samples. The sampling frequency (or throughput) for Figure 2 is given by Equation (3)

$$f_{sample} \leq \frac{1}{T_M + T_A} \quad (3)$$

Figure 7 represents a 4-tap broadcast-type word serial diagram. The DFG of the original 4-tap FIR filter architecture is shown in Figure 8. To unfold Figure 8 by an unfolding factor two, the eighteen nodes X_i , A_i , B_i , C_i , D_i , E_i , F_i , G_i , and Y_i for $i=0,1$ are first drawn. Then second step of the unfolding algorithm is applied to construct edges. The 2-unfolded DFG for the 4-tap FIR filter architecture is shown in Figure 9.

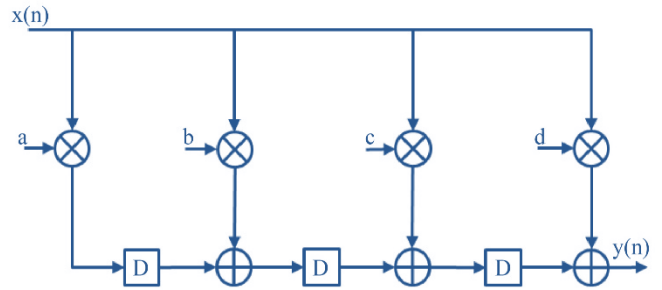


Fig. 7 4-tap FIR filter architecture

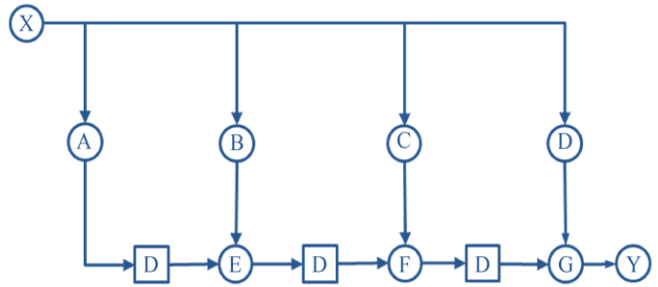


Fig. 8 The DFG for the 4-tap FIR filter architecture

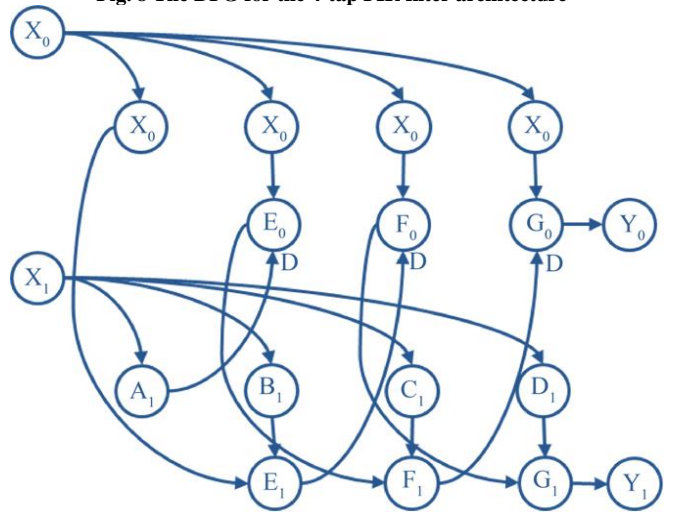


Fig. 9 The 2-unfolded DFG for the 4-tap FIR filter architecture

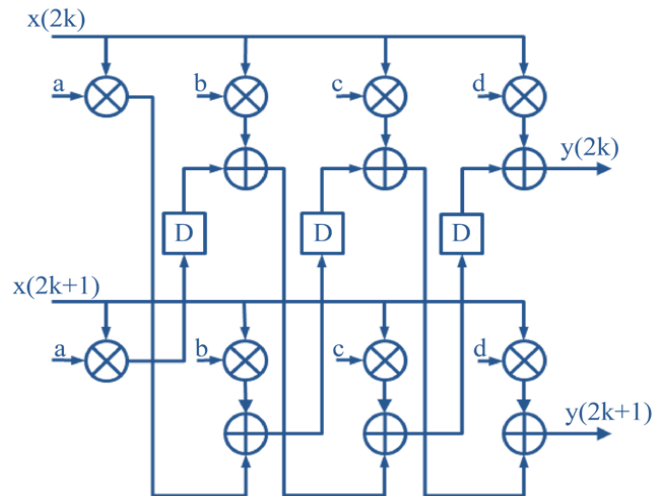


Fig. 10 2-Parallel FIR filter architecture for 4-tap FIR filter

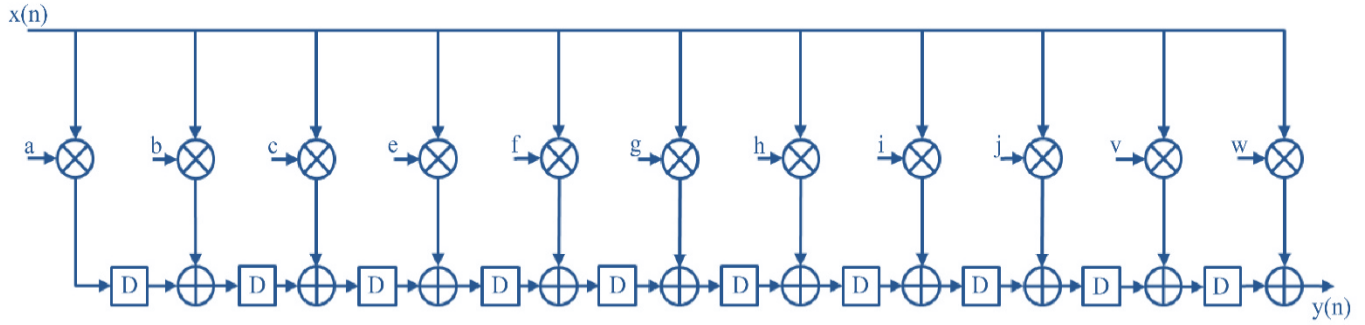


Fig. 11 11-tap FIR filter architecture

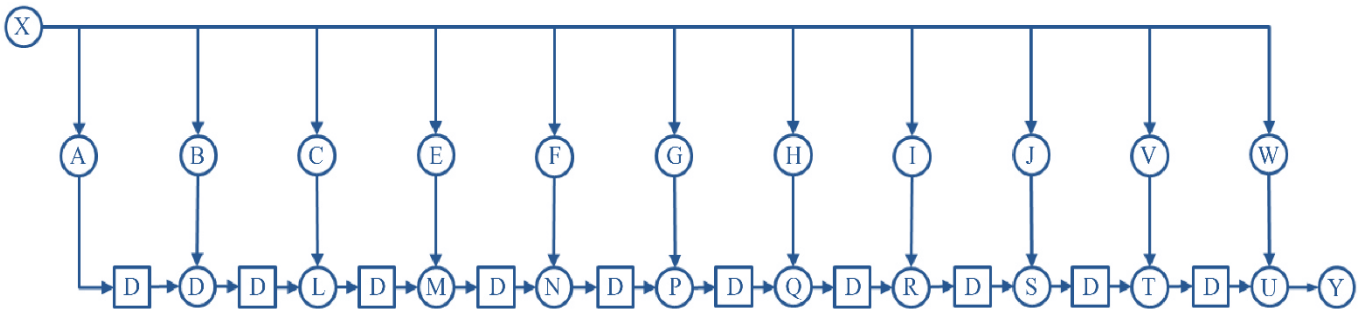


Fig. 12 The DFG for the 11-tap FIR filter architecture

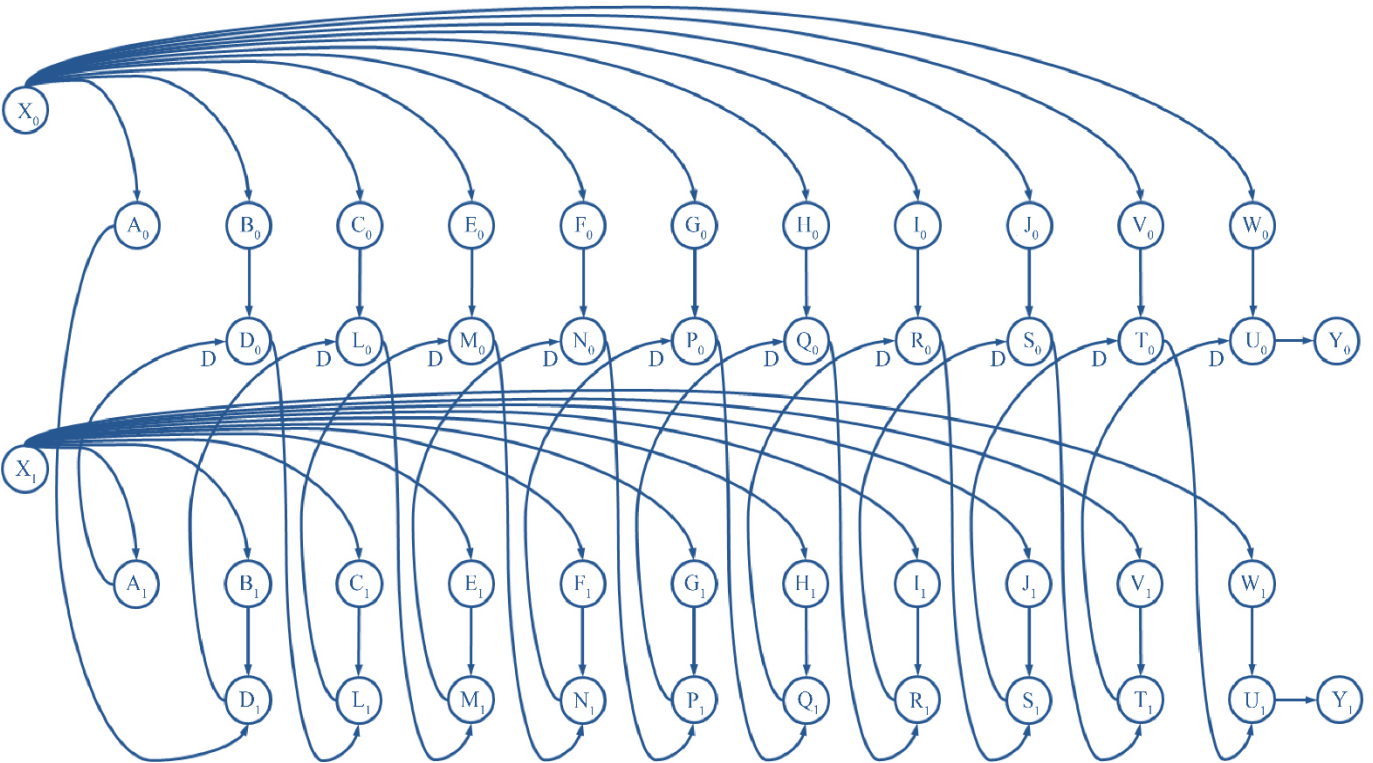


Fig. 13 The 2-unfolded DFG for the 11-tap FIR filter architecture

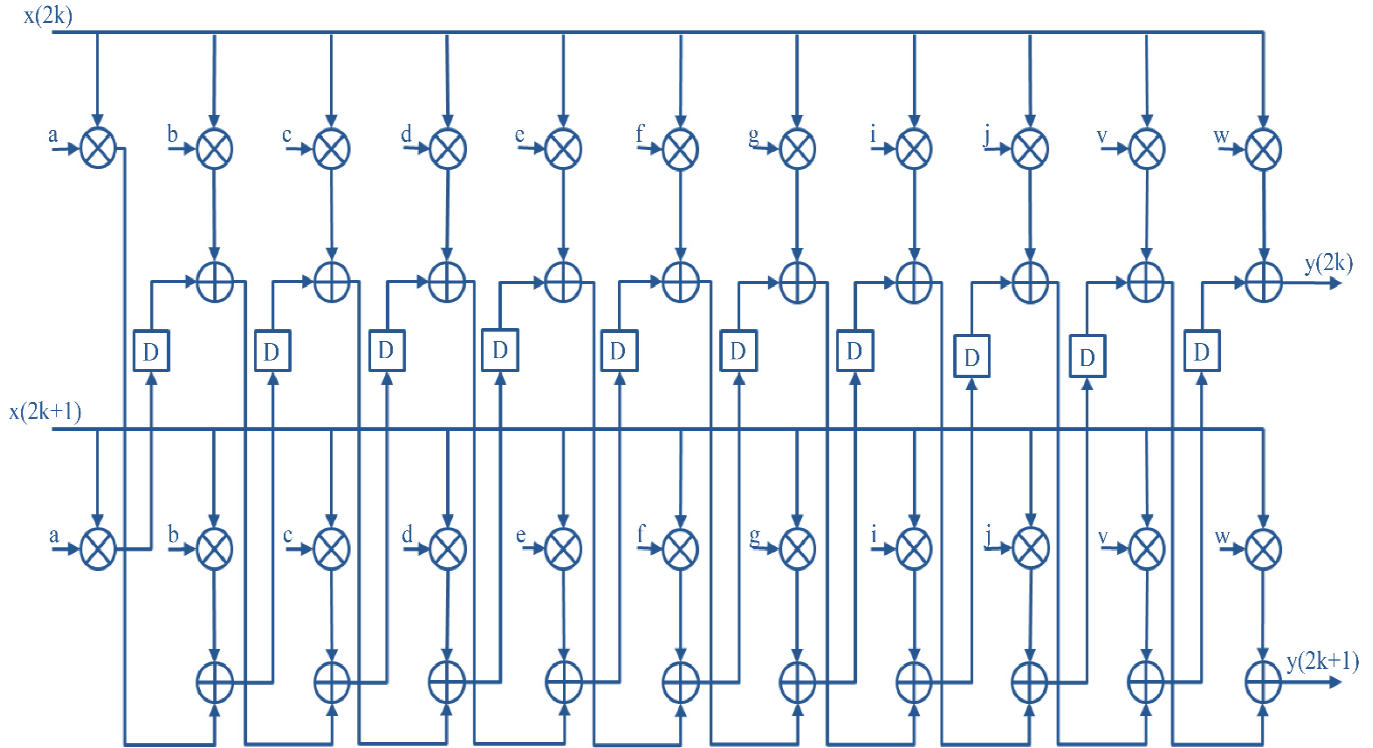


Fig. 14 2-Parallel FIR filter architecture for 11-tap FIR filter

2-Parallel FIR filter architecture is constructed by applying an unfolding technique in the original word serial 4-tap FIR filter architecture. Figure 10 represents the 2-parallel FIR filter architecture for Figure 7. For generating two input samples for a 2-parallel FIR filter architecture a serial-to-parallel converter circuit is used.

In Figure 5, the 2-parallel FIR filter architecture processes two input samples per clock cycle to generate two output samples. The sampling frequency (or throughput) for Figure 7 is given by Equation (4)

$$f_{sample} \leq \frac{1}{T_M + T_A} \tag{4}$$

Figure 11 represents an 11-tap broadcast-type word serial diagram

DFG of the original 11-tap FIR filter architecture is shown in Figure 12.

To unfold Figure 12 by an unfolding factor two, the 46 nodes $X_i, A_i, B_i, C_i, D_i, E_i, F_i, G_i, H_i, I_i, J_i, V_i, W_i, L_i, M_i, N_i, P_i, Q_i, R_i, S_i, T_i, U_i$ and Y_i for $I = 0,1$ are first drawn. Then second step of the unfolding algorithm is applied to construct edges. The 2-unfolded DFG for the 11-tap FIR filter architecture is shown in Figure 13.

2-Parallel FIR filter architecture is constructed by applying an unfolding technique in the original word serial 11-

tap FIR filter architecture. Figure 14 represents the 2-parallel FIR filter architecture for Figure 11. For generating two input samples for a 2-parallel FIR filter architecture, a serial-to-parallel converter circuit is used.

In Figure 11, the 2-parallel FIR filter architecture processes two input samples per clock cycle to generate two output samples. The sampling frequency (or throughput) for Figure 11 is given by Equation (5)

$$f_{sample} \leq \frac{1}{T_M + T_A} \tag{5}$$

All above proposed broadcast type word serial FIR filter architectures have higher throughput compared with non broadcast type 2-tap, 4-tap and 11-tap FIR filter architectures. MATLAB FDA tool [9] is used to calculate filter coefficients for the proposed low pass 2-tap, 4-tap and 11-tap FIR filter architectures. In the above unfolded architectures, D represents delay elements, and for hardware implementation, DFF is used to realize it.

3. Experimental Results and Comparison with Serial and Parallel Filter Architectures

VHDL language is used to design all components (16 bit adder circuits, 8x8 multiplier circuits, 16 bit DFFs) of proposed 2-parallel FIR filter architectures. Xilinx Vivado 2015.2 tool is used for synthesis and simulation for the proposed 2-parallel FIR filter architectures. Calculated 8 bit FIR filter coefficients by using the MATLAB FDA tool are

stored in a constant ROM array. 8-bit signed integer format input samples (x(n)) are stored in a text file, and VHDL reads this text file for doing simulation by using Xilinx Vivado 2015.2 tool.

Simulated output waveforms for 2-parallel 11-tap FIR filter, 2-parallel 2-tap FIR filter and 2-parallel 4-tap FIR filter architectures are shown in Figures 15, 16 and 17, respectively. In all these figures, clk represents 100MHz system clock, clk 12Khz represents 1200KHz clock frequency for 2-parallel FIR filter architectures, xin represents 8-bit input sequence and y2k, y2kplus1 represents generated output samples for 2-parallel FIR filter architectures.

Also, VHDL writes generated output samples y2k and y2kplus1 in text files for all proposed 2-parallel FIR filter

architectures. For implementing all the above 2-parallel target FPGA device is 7a35t-cpg236. RTL schematic diagrams for 2-parallel 11-tap FIR filter, 2-parallel 2-tap FIR filter and 2-parallel 4-tap FIR filter architectures are shown in Figures 18, 19 and 20, respectively.

Synthesis results for the implemented 2-parallel 11-tap FIR filter, 2-parallel 2-tap FIR filter and 2-parallel 4-tap FIR filter architectures are shown in Table 1, Table 2 and Table 3, respectively. For evaluating the performance of the implemented FIR filter architectures, speed comparison among FIR filter architectures is shown in Table 4, area (in terms of LUTs) comparison among FIR filter architectures is shown in Table 5 and power consumption (in watt) comparison among FIR filter architectures is shown in Table 6.

Table 1. Synthesis results for the 2- parallel 11-tap FIR filter

Area(in terms of LUT)	Delay(ns)	Total Power Consumption(in watts)
510	6.689	39.983

Table 2. Synthesis results for the 2- parallel 2-tap FIR filter

Area(in terms of LUT)	Delay(ns)	Total Power Consumption(in watts)
105	6.673	29.402

Table 3. Synthesis results for the 2- parallel 4-tap FIR filter

Area(in terms of LUT)	Delay(ns)	Total Power Consumption(in watts)
111	6.735	31.677

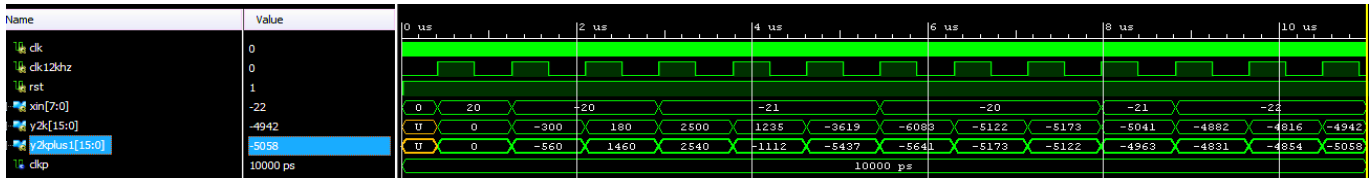


Fig. 15 Simulated output waveform for 2-parallel 11-tap FIR filter

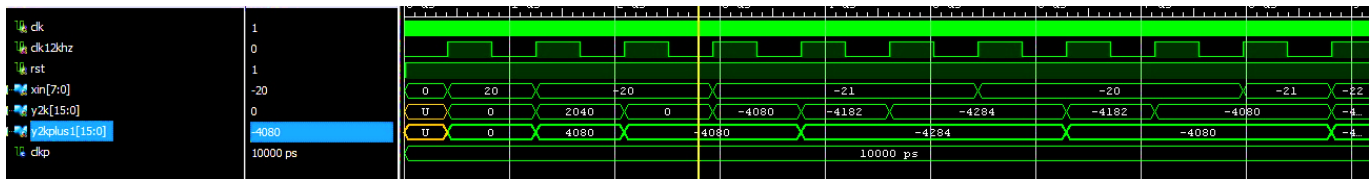


Fig. 16 Simulated output waveform for 2-parallel 2-tap FIR filter



Fig. 17 Simulated output waveform for 2-parallel 4-tap FIR filter

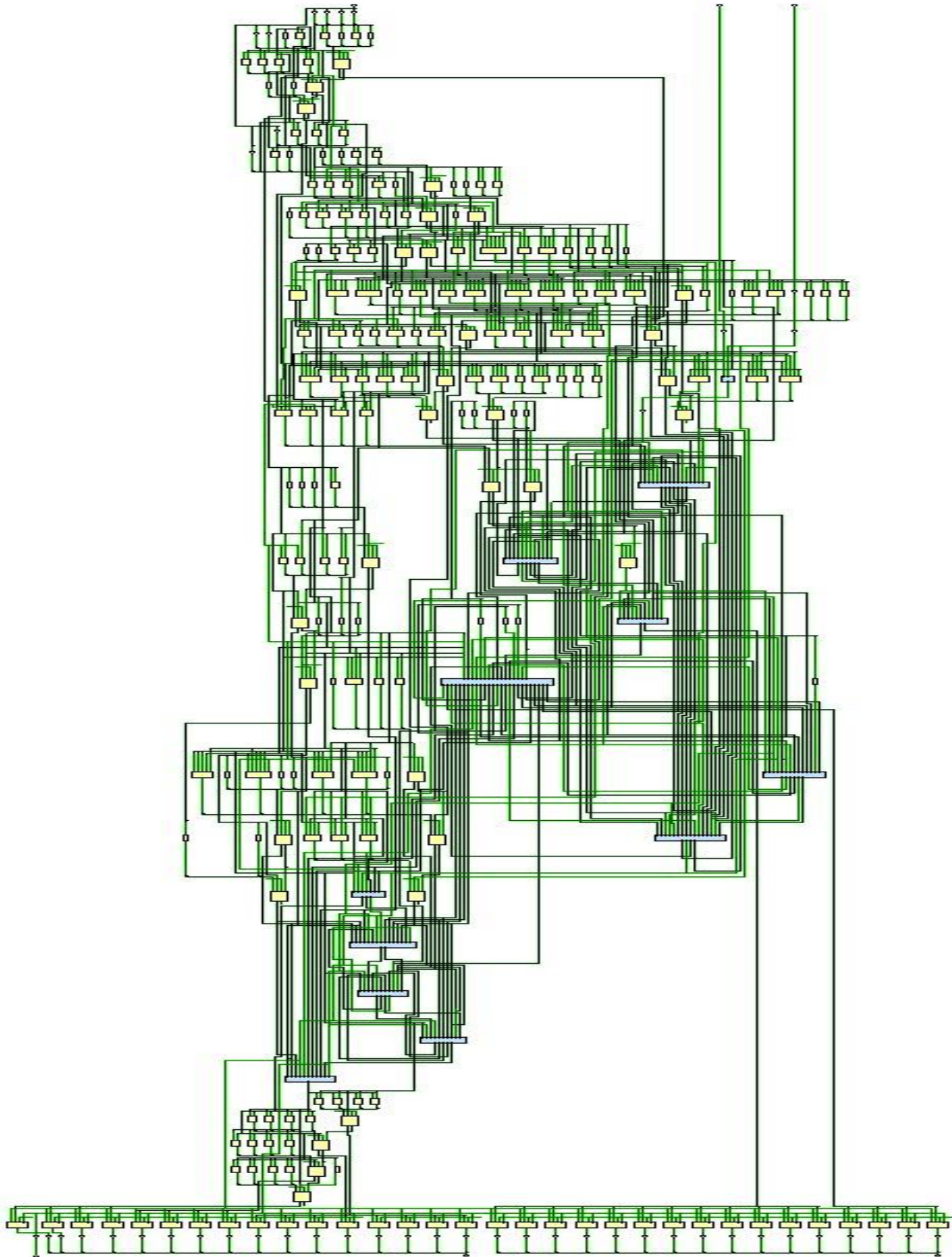


Fig. 18 RTL schematic diagram for 2-parallel 11-tap FIR filter

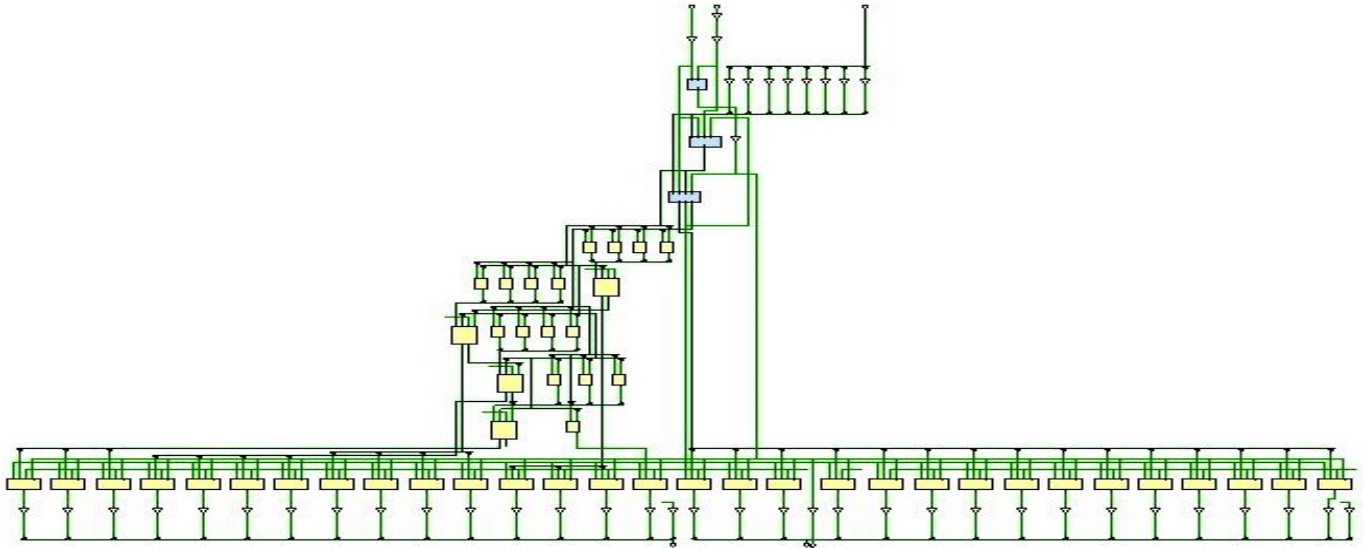


Fig. 19 RTL schematic diagram for 2-parallel 2-tap FIR filter

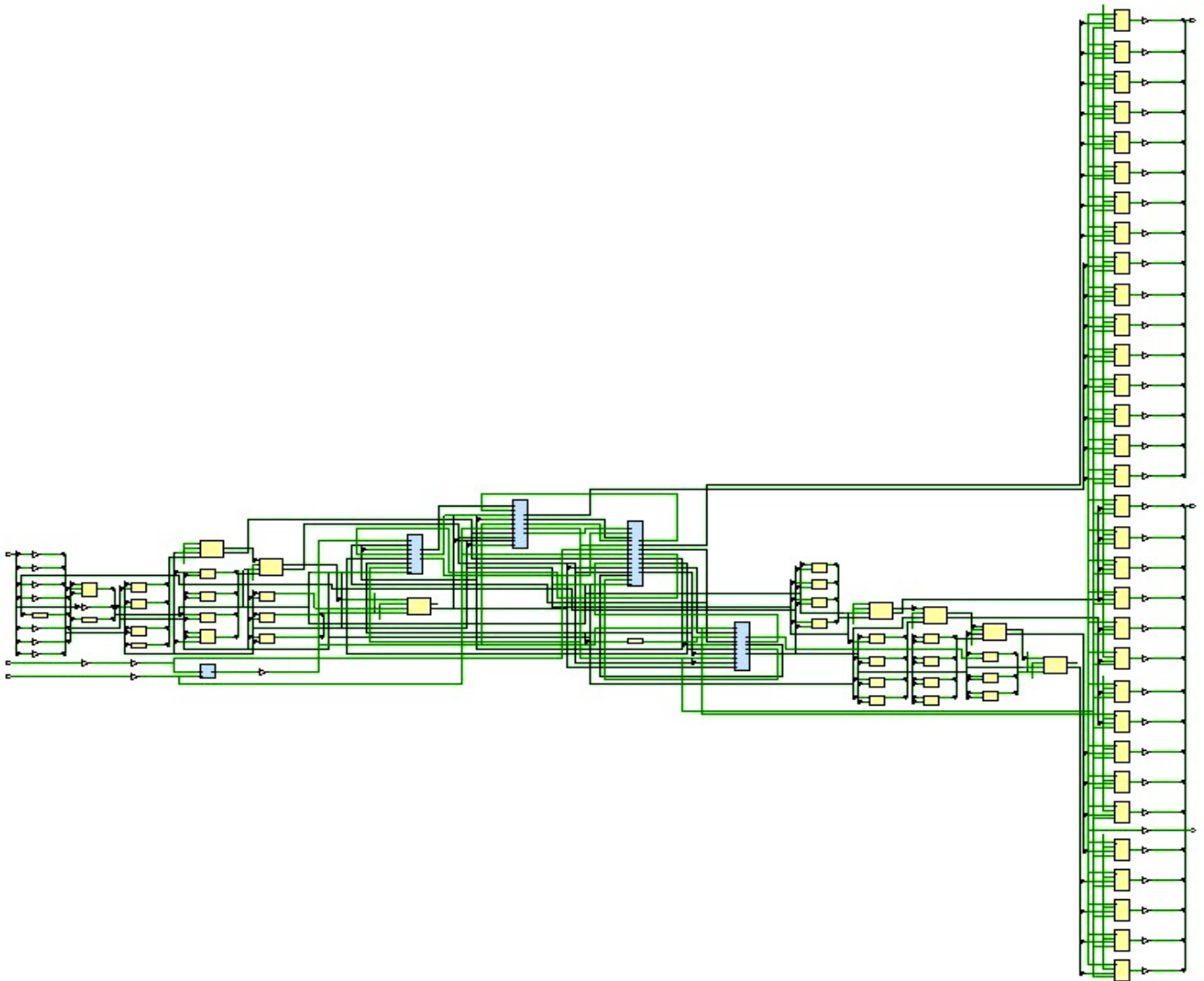


Fig. 20 RTL schematic diagram for 2-parallel 4-tap FIR filter

Table 4. Speed comparison among different FIR filter architectures

Structure	FIR filter type	Delay(ns)
16-bit Vedic multiplier [10]	4-tap micro-programmed sequential FIR filter	10.56
16-bit Wallace tree multiplier[10]	4-tap micro-programmed sequential FIR filter	15.56
Virtex-4 (XC4VFX12)[11]	2-tap serial FIR filter	22.924
Virtex-5 (XC5VLX110T)[11]	2-tap serial FIR filter	16.841
Virtex-6 (XC6VCX75T)[11]	2-tap serial FIR filter	12.024
Virtex-4 (XC4VFX12)[11]	3-tap serial FIR filter	24.648
Virtex-5 (XC5VLX110T)[11]	3-tap serial FIR filter	18.696
Virtex-6 (XC6VCX75T)[11]	3-tap serial FIR filter	17.411
Virtex-4 (XC4VFX12)[11]	2-tap pipelined FIR filter	20.284
Virtex-5 (XC5VLX110T)[11]	2-tap pipelined FIR filter	14.268
Virtex-6 (XC6VCX75T)[11]	2-tap pipelined FIR filter	9.898
Virtex-4 (XC4VFX12)[11]	3-tap pipelined FIR filter	22.012
Virtex-5 (XC5VLX110T)[11]	3-tap pipelined FIR filter	15.928
Virtex-6 (XC6VCX75T)[11]	3-tap pipelined FIR filter	15.456
3 s500efg320-4[12]	2-parallel FIR filter	11.905
3 s500efg320-4[12]	Area-efficient 2-parallel FIR filter	10.880
Proposed Artix7 Basys3(7a35t-cpg236)	2-parallel 11-tap FIR filter	6.689
Proposed Artix7 Basys3(7a35t-cpg236)	2-parallel 2-tap FIR filter	6.673
Proposed Artix7 Basys3(7a35t-cpg236)	2-parallel 4-tap FIR filter	6.735

Table 5. Area utilization (%) comparison among different FIR filter architectures

Structure	FIR filter type	Area Utilization in LUT (%)
Virtex-4 (XC4VFX12)[11]	2-tap serial FIR filter	3
Virtex-4 (XC4VFX12)[11]	3-tap serial FIR filter	6
Virtex-4 (XC4VFX12)[11]	2-tap pipelined FIR filter	3
Virtex-4 (XC4VFX12)[11]	3-tap pipelined FIR filter	6
Proposed Artix7 Basys3(7a35t-cpg236)	2-parallel 11-tap FIR filter	2.45
Proposed Artix7 Basys3(7a35t-cpg236)	2-parallel 2-tap FIR filter	0.50
Proposed Artix7 Basys3(7a35t-cpg236)	2-parallel 4-tap FIR filter	0.53

Table 6. Power consumption (in watt) comparison among different FIR filter architectures

Structure	FIR filter type	Power Consumption(in watt)
Xc7a200tfbg676[6]	3-tap, 3-unfolded FIR filter	34.928
Xc7a200tfbg676[6]	3-tap 2-unfolded FIR filter	22.857
Proposed Artix7 Basys3 (7a35t-cpg236)	2-parallel 11-tap FIR filter	39.983
Proposed Artix7 Basys3 (7a35t-cpg236)	2-parallel 2-tap FIR filter	29.402
Proposed Artix7 Basys3 (7a35t-cpg236)	2-parallel 4-tap FIR filter	31.677

Table 4 shows that the proposed 2-parallel 2-tap FIR filter architecture has less delay as compared to 2-parallel 4-tap, 2-parallel 11-tap and existing FIR filter architectures. Table 5 shows that the proposed 2-parallel 2-tap FIR filter architecture utilized less area in terms of LUTs, as compared to 2-parallel 4-tap, 2-parallel 11-tap and existing FIR filter architectures. Table 6 shows that the proposed 2-parallel 2-tap architecture consumes less power as compared to 2-parallel 4-tap and 2-parallel 11-tap FIR filter architectures. The proposed approach also takes less power in comparison with the existing 3-tap, 3-unfolded FIR filter.

4. Conclusion

Unfolding can be used for sample period reduction, word level parallel processing, and bit-level parallel processing for any given DSP program. In this paper, an application of unfolding named word level parallel processing is done for low pass word serial 11-tap, 2-tap and 4-tap broadcast type FIR filter architectures. Broadcast type FIR filter architectures have a high throughput value as compared to non-broadcast

type FIR filter architectures. The unfolding factor value is two for designing 2-parallel FIR filter architectures. Serial-to-parallel converter circuit is constructed for generating input samples $x(2k)$ and $x(2k+1)$ at the clock frequency of 2400KHz. Output samples $y(2k)$ and $y(2k+1)$ are generated at the clock frequency of 1200KHz. The data path delay for the implemented 2-parallel 11-tap FIR filter is 6.689ns. The data path delay for the implemented 2-parallel 2-tap and 4-tap FIR filter architecture is 6.673ns and 6.735ns, respectively. Among proposed 2-parallel FIR filter architectures, 2-tap FIR filter architectures have higher speed. Proposed 2-parallel FIR filter architectures have higher speeds compared with existing serial, pipelined and parallel FIR filter architectures. Area utilization (%) for the implemented 2-parallel 11-tap FIR filter is 2.45. Area utilization (%) for the implemented 2-parallel 2-tap and 4-tap FIR filter architecture is 0.50 and 0.53, respectively. Among proposed 2-parallel FIR filter architectures 2-tap FIR filter architectures have consumed less area. Proposed FIR filter architectures also consume less area compared with existing serial and pipelined FIR filter architectures.

References

- [1] Keshab K. Parhi, *VLSI Digital Signal Processing Systems Design and Implementation*, Wiley, pp. 1-808, 2007. [[Google Scholar](#)] [[Publisher Link](#)]
- [2] Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman, *Compilers, Principles, Techniques, and Tools*, Addison-Wesley Publishing Company, pp. 1-796, 1986. [[Google Scholar](#)] [[Publisher Link](#)]
- [3] K.K. Parhi, and D.G. Messerschmitt, "Pipeline Interleaving and Parallelism in Recursive Digital Filters. I. Pipelining Using Scattered Look-Ahead and Decomposition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no.7, pp. 1099-1117, 1989. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [4] K.K. Parhi, and D.G. Messerschmitt, "Pipeline Interleaving and Parallelism in Recursive Digital Filters. II. Pipelined Incremental Block Filtering," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 7, pp. 1118-1135, 1989. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [5] Anantha P. Chandrakasan, Samuel Sheng, and Robert W. Brodersen, "Low-Power CMOS Digital Design," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 4, pp. 473-484, 1992. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [6] Manoj Kumar, and Karni Ram, "Implementation of Word Level Parallel Processing Unfolding Algorithm Using VHDL," *International Journal of Engineering and Advanced Technology*, vol. 8, no. 6, pp. 664-667, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [7] S.K. Manikandan et al., "High Throughput LFSR Design for BCH Encoder Using Sample Period Reduction Technique for MLC NAND Based Flash Memories," *International Journal of Computer Applications*, vol. 66, no. 10, pp. 33-39, 2013. [[Google Scholar](#)] [[Publisher Link](#)]
- [8] Sangeeta Singh et al., "VLSI Implementation of Parallel CRC Using Pipelining, Unfolding and Retiming," *IOSR Journal of VLSI and Signal Processing*, vol. 2, no. 5, pp. 66-72, 2013. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [9] Daimond Soibam Singh, and Manoj Kumar, "Design and Implementation of Third Order Low Pass Digital FIR Filter Using Pipelining Retiming Technique," *International Journal of Engineering and Advanced Technology*, vol. 10, no. 4, pp. 178-184, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [10] Tamli Dhanraj Sawarkar, Lokesh Chawle, and N.G. Narole, "Implementation of 4-Tap Sequential and Parallel Micro-Programmed Based Digital FIR Filter Architecture Using VHDL," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 4, no. 4, pp. 6906-6913, 2016. [[Google Scholar](#)] [[Publisher Link](#)]
- [11] R. Saranya, "FPGA Synthesis of Reconfigurable Modules for FIR Filter," *International Journal of Reconfigurable and Embedded Systems*, vol. 4, no. 2, pp. 63-70, 2015. [[Google Scholar](#)] [[Publisher Link](#)]
- [12] L. Kholee Phimu, and Manoj Kumar, "VLSI Implementation of Area Efficient 2-Parallel FIR Digital Filter," *International Journal of VLSI design & Communication Systems*, vol. 7, no. 5/6, pp. 17-24, 2016. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [13] N. Sankarayya, K. Roy, and D. Bhattacharya, "Algorithms for Low Power and High Speed FIR Filter Realization Using Differential Coefficients," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, no. 6, pp. 488-497, 1997. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]