Research Journal of Applied Sciences, Engineering and Technology 11(12): 1404-1409, 2015
DOI: 10.19026/rjaset.11.2247
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: June 24, 2015 Accepted: August 2, 2015 Published: December 25, 2015

Research Article

Implementation of High Speed Pipelined Distributed Arithmetic Based FIR Filter

[1]P.L. Joseph Raj and [2]T. Vigneswaran
[1]Sathyabama University,
[2]VIT University, Chennai, TN, India

Abstract: In the explosive growth of wireless and networking applications, Digital Signal Processing (DSP) operations are extensively used for characterizing and controlling the discrete input signals. For those DSP operations, Finite Impulse Response (FIR) filter is used to filter the unwanted/noise/distorted signals from the discrete input signals. In this study, design of Pipelined Distributed Arithmetic (DA) based FIR filter is implemented through Very Large Scale Integration (VLSI) System design environment for variable word lengths. In general, Multiplication and Accumulation (MAC) unit is the heart of any digital signal processor. But MAC architectures utilize more hardware resources and consume large delay and power. In order to overcome this problem, multiplier-less architecture known as distributed arithmetic multiplication is designed in our work. In this study, we introduce Pipelined DA based FIR filter with the help of Pipelining Registers. To reduce the memory size of DA based multiplication, Look-up Tables (LUTs) are partitioned by using Offset Binary Coding (OBC) technique. Hence speed and power consumption of Pipelined DA based FIR filtering operation is improved than traditional DA based FIR filtering operation, for increasing word length inputs.

Keywords: Distributed Arithmetic (DA) multiplier, Finite Impulse Response (FIR) filter, Multiplication and Accumulation (MAC) unit, Offset Binary Coding (OBC), Pipelined DA based FIR filter, Very Large Scale Integration (VLSI)

INTRODUCTION

Digital Signal Processor (DSP) plays a very important role in the explosive growth of wired and wireless networks as well as multimedia applications. Among different types of signal processing operations, Finite Impulse Response (FIR) digital filtering technique is widely used for signal filtering applications. Throughout the years, large endeavours have realized the FIR filter architectures with the help of Very Large Scale Integration (VLSI) technology. Low power consumption, less silicon area utilization and high speed are the main concern in VLSI System design environment. In general, digital FIR filter consists of delay elements, adders and multipliers for filtering the unwanted/undesired signals. Therefore, performance of a digital FIR filter depends on Multiplication and Accumulation (MAC) unit. Several algorithms/methodologies have been mapped into MAC architecture to improve the performance of digital FIR filter (Sudhakar *et al*., 2012); one of them being Distributed Arithmetic (DA).

Distributed Arithmetic (DA) based multiplier is one of the efficient digital multiplication mechanisms in which Look Up Tables (LUTs) are used to store the pre-computed values and it can be read out easily based on input values. Hence, DA based multiplier structure is also named as multiplier-less architecture. It can be designed to meet various speed requirements. Systolic realization of DA based multiplier and LUT based multiplication is introduced in Meher *et al*. (2008) and Meher (2010) respectively. In this study, decomposition of multiplication is performed by using LUT tables. Fixed pre-computed coefficient values are selected in the first clock cycles based on input values. After selecting the fixed pre-computed values, shifting and addition of multiplied samples are performed in the next clock cycle. Performance analysis of DA based multiplication is described in Murali *et al*. (2014) and Suhasini *et al*. (2012). Ripple Carry Adder (RCA) is used in Suhasini *et al*. (2012) to perform addition operation. Pavel and Richard (2014), efficient fully parallel FIR filter structure is developed for symbol synchronization purpose. Due to accessing of coefficient values, parallel FIR filter consumes large delay and speed of the filtering process becomes slow. In order to overcome this problem, modified DA is developed in Ramesh and Nathiya (2012) and Michal and Mariusz (2012). This architecture is realized through hardware in Eshwararao and Lokeshraju (2012). Further, to improve the performance of digital FIR filter, Block Least Mean Square Algorithm is introduced in Devipriya *et al*. (2014) for DA based

Corresponding Author: P.L. Joseph Raj, Sathyabama University, Chennai, TN, India

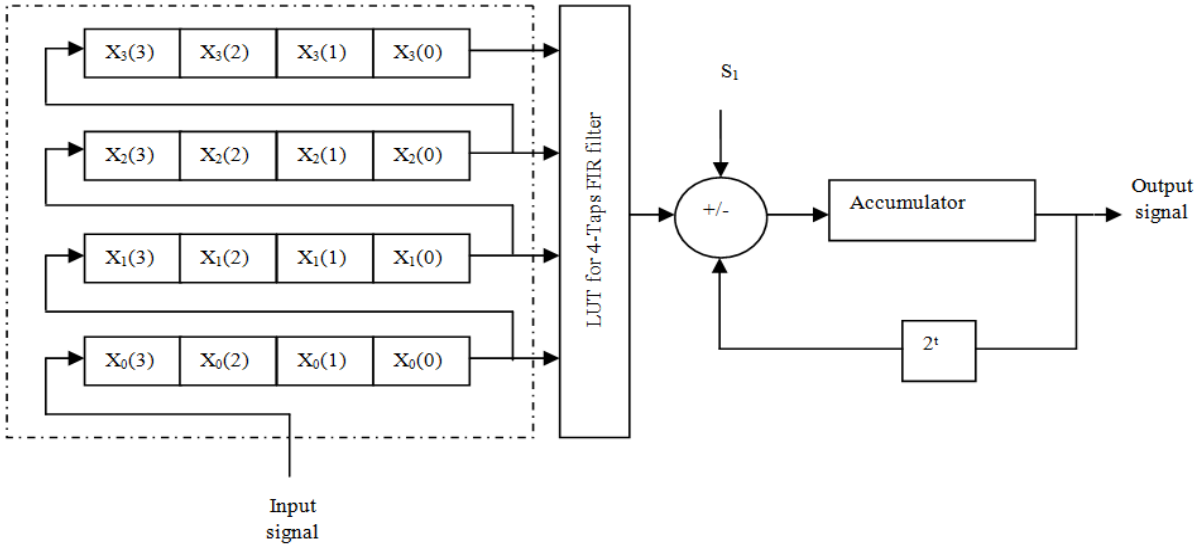

Fig. 1: DA based FIR filter implementation

multiplications. Built-In Self Test (BIST) circuit is designed in Toihria *et al*. (2014) for test a loop filter. Also BIST can be used in DA based FIR filter for testing.

In this study, Pipelined DA multiplication based digital FIR filter is designed through Very Large Scale Integration (VLSI) System design environment. Four tap FIR filter is designed in this study for different lengths such as 8-bit and 16-bit. The performance of variable length pipelined DA based digital FIR filter is compared with traditional DA based digital FIR filter.

DISTRIBUTED ARITHMETIC BASED DIGITAL FIR FILTER

Distributed Arithmetic is an efficient technique for calculation of inner product. The MAC operation is common in Digital Signal Processors (DSP). Dedicated multipliers have been used for MAC operation. Those multipliers are fast but they consume considerable hardware. The following expression represents a multiply and accumulate operation:

$$y = A_1 \times x_1 + A_2 \times x_2 + \ldots\ldots + A_n \times x_n \quad (1)$$

$$y = \sum_{k=1}^{n} A_k x_k \quad (2)$$

To reduce the hardware complexity for multiplier, DA based multiplier is preferred in this study. DA is a technique that is bit-serial in nature. It replaces the explicit multiplications by ROM look-ups where pre-computed values for coefficients are stored. The formulation of DA is described as follows:

In Eq. (2), x_k be an N-bit scaled two's complement number. In other hand, it is denoted as:

$$|x_k| < 1 \quad (3)$$

$$x_k : \{b_{k0}, b_{k1}, b_{k2} \ldots\ldots b_{k(n-1)}\} \quad (4)$$

where, b_{k0} is the sign bit. From Eq. (3) and Eq. (4), x_k can be represented as follows:

$$x_k = -b_{k0} + \sum_{n=1}^{N-1} b_{kn} 2^{-n} \quad (5)$$

When substituting (5) in (2), we get:

$$y = \sum_{k=1}^{n} A_k \left[-b_{k0} + \sum_{n=1}^{N-1} b_{kn} 2^{-n} \right]$$

$$y = \sum_{k=1}^{n} (b_{k0} \bullet A_k) + \sum_{k=1}^{n} \sum_{m=1}^{N-1} (A_k \bullet b_{km}) 2^{-m} \quad (6)$$

Further simplification is made on Eq. (6) based on expansion and compression technique, we get:

$$y = -\sum_{k=1}^{n} A_k \bullet (b_{k0}) + \sum_{m=1}^{N-1} \left[\sum_{k=1}^{n} A_k \bullet b_{km} \right] 2^{-m} \quad (7)$$

Equation (7) can be rewritten as:

$$y = \underbrace{\sum_{m=1}^{N-1} \left[\sum_{k=1}^{n} A_k b_{km} \right] 2^{-m}}_{\text{has only } 2^n \text{ possible}} + \underbrace{\sum_{k=1}^{n} A_k (-b_{k0})}_{\text{has only } 2^n \text{ possible}}$$

Table 1: LUT for 4 taps FIR filter

	b_{1n}	b_{2n}	b_{3n}	b_{4n}	Contents
Y =	0	0	0	0	0
$A_1 \times x_1 + A_2 \times x_2 + \ldots +$	0	0	0	1	A_4
$A_n \times x_n$	0	0	1	0	A_3
	0	0	1	1	A_3+A_4
	0	1	0	0	A_2
	0	1	0	1	A_2+A_4
	0	1	1	0	A_2+A_3
	0	1	1	1	$A_2+A_3+A_4$
	1	0	0	0	A_1
	1	0	0	1	A_1+A_4
	1	0	1	0	A_1+A_3
	1	0	1	1	$A_1+A_3+A_4$
	1	1	0	0	A_1+A_2
	1	1	0	1	$A_1+A_2+A_4$
	1	1	1	0	$A_1+A_2+A_3$
	1	1	1	1	$A_1+A_2+A_3+A_4$

With the sign bit as an input, we can store it in a ROM of size = $2*2^n$. For example, let the number of taps be 4, we need $2^n = 2^4 = 16$-words ROM. The fixed coefficient values for 4-taps are stored in LUT, which is demonstrated in Table 1. DA based FIR filter implementation for 4-taps FIR filter is shown in Fig. 1.

DA based multiplication has accomplished MAC without an explicit multiplier. But, the size of ROM however grows exponentially with each added input address line. For each element in a vector, it requires an address line. To reduce memory size of LUT, Offset Binary Coding (OBC) is introduced in this study. Design of digital FIR filter by using OBC is briefly described in next section.

PIPELINED DA BASED FIR FILTER BY USING OFFSET BINARY CODING

Traditional DA based FIR filter uses the fixed LUT as ROM for storing the pre-computed coefficient values. Due to increasing the length of ROM size, hardware complexity of DA based FIR filter can be increased considerable. Also delay for filtering operation can be increased due to accessing the coefficient through large length of memory. In order to reduce these disadvantages of DA based FIR filter, pipelining mechanism is introduced in this study. Also OBC technique is used to LUT partitioning technique.

In proposed work, pipelined DA based FIR filter is designed by adding the pipelining register in the output of partitioned LUT's. Memory size of ROM can be reduced by OBC technique (LUT partition) and speed of filtering process can be improved by pipelining register. The following expression represents the ROM size reduction technique using OBC:

$$x_k = \frac{1}{2}\left[x_k - \left(-x_k\right)\right] \tag{8}$$

2's complement of x_k Eq. (5) can be represented as:

$$-x_k = -\bar{b}_{k0} + \sum_{m=1}^{N-1} \bar{b}_{km} 2^{-m} + 2^{-(N-1)} \tag{9}$$

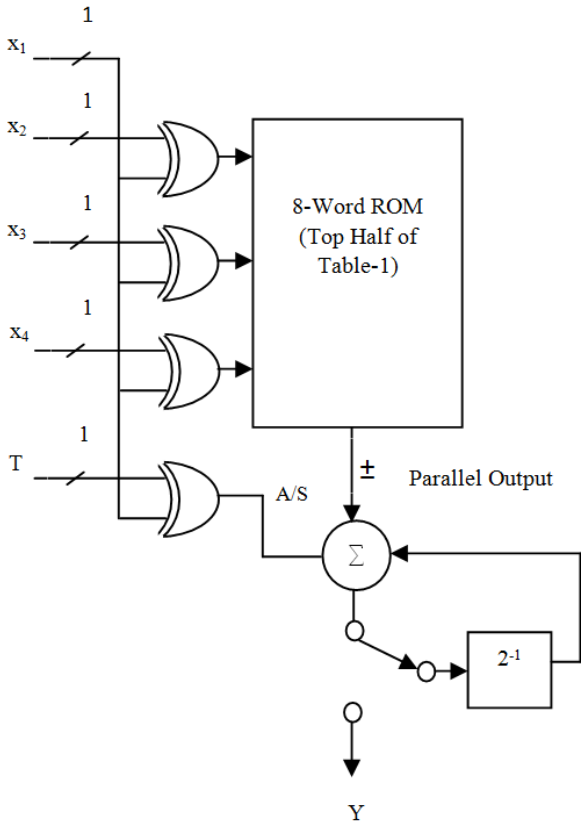

Fig. 2: LUT partitioning technique for 4-taps FIR filter

Re-writing the value of x_k we get:

$$x_k = \frac{1}{2}\left[-\left(b_{k0} - \bar{b}_{k0}\right) + \sum_{m=1}^{N-1}\left(b_{km} - \bar{b}_{km}\right)2^{-m} - 2^{-(N-1)}\right] \tag{10}$$

Offset code can be defined as:

$$c_{kn} = \left\{\begin{matrix} b_{km} - \bar{b}_{km}, m \neq 0 \\ -\left(b_{km} - \bar{b}_{km}\right), m = 0 \end{matrix}\right\} where\, c_{km} \in \{-1, 1\} \tag{11}$$

Finally:

$$x_k = \frac{1}{2}\left[\sum_{m=0}^{N-1} c_{km} 2^{-m} - 2^{-(N-1)}\right] \tag{12}$$

Using the new x_k, we have:

$$y = \sum_{m=0}^{N-1} \frac{1}{2} \sum_{k=1}^{n} A_k c_{km} 2^{-m} - \frac{1}{2} \sum_{k=1}^{n} A_k 2^{-(N-1)} \tag{13}$$

By using the Eq. (13), LUT part can be partitioned by two half. The partitioned LUT for 4-taps FIR filter is illustrated in Table 2. LUT partitioned by using OBC is

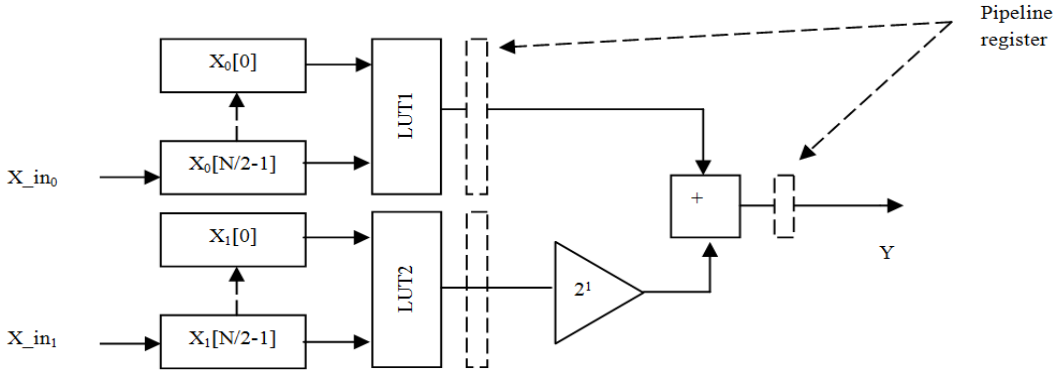

Fig. 3: Pipelined DA based FIR filter implementation

Table 2: Partitioned LUT for 4 taps FIR filter

X_{1n}	X_{2n}	X_{3n}	X_{4n}	C_{1n}	C_{2n}	C_{3n}	C_{4n}	Data
0	0	0	0	-1	-1	-1	-1	$-1/2(A_1+A_2+A_3+A_4)$
0	0	0	1	-1	-1	-1	1	$-1/2(A_1+A_2+A_3-A_4)$
0	0	1	0	-1	-1	1	-1	$-1/2(A_1+A_2-A_3+A_4)$
0	0	1	1	-1	-1	1	1	$-1/2(A_1-A_2+A_3+A_4)$
0	1	0	0	-1	1	-1	-1	$-1/2(A_1-A_2+A_3-A_4)$
0	1	0	1	-1	1	-1	1	$-1/2(A_1-A_2-A_3+A_4)$
0	1	1	0	-1	1	1	-1	$-1/2(A_1-A_2-A_3-A_4)$
0	1	1	1	-1	1	1	1	$-1/2(-A_1+A_2+A_3+A_4)$
1	0	0	0	1	-1	-1	-1	$-1/2(-A_1+A_2+A_3-A_4)$
1	0	0	1	1	-1	-1	1	$-1/2(-A_1+A_2+A_3-A_4)$
1	0	1	0	1	-1	1	-1	$-1/2(-A_1+A_2-A_3-A_4)$
1	0	1	1	1	-1	1	1	$-1/2(-A_1+A_2+A_3+A_4)$
1	1	0	0	1	1	-1	-1	$-1/2(-A_1-A_2-A_3-A_4)$
1	1	0	1	1	1	-1	1	$-1/2(-A_1-A_2+A_3-A_4)$
1	1	1	0	1	1	1	-1	$-1/2(-A_1-A_2-A_3+A_4)$
1	1	1	1	1	1	1	1	$-1/2(-A_1-A_2-A_3-A_4)$

Table 3: Comparison of proposed pipelined DA based FIR filter and traditional FIR filter for variable lengths

Performance analysis of 4-Taps 8-bit DA based FIR filters				
Type	Slices	Flip-flops	Delay (ns)	Power (mW)
DA based FIR filter	38	52	5.619	231
Pipelined DA based FIR filter	35	44	5.089	386
Performance analysis of 4-Taps 16-bit DA based FIR filters				
Type	Slices	Flip-flops	Delay (ns)	Power (mW)
DA based FIR filter	48	65	5.665	1663
Pipelined DA based FIR filter	55	80	3.727	1606

illustrated in Fig. 2. After completing 4-taps iteration output is switched to y in Fig 2. Sw1 represents the switch which handles the iteration of operation. Pipelined DA based FIR filter is illustrated in Fig. 3.

It consists of two numbers of LUTs, delay elements and adder. Dotted lines in Fig. 3 indicated as pipelining register. Pipelining registers are used to provide synchronism for output of two LUTs. Therefore, pipelining mechanism helps to speed of FIR filtering process. The implementation of pipelined DA based FIR filter is done for variable length such as 8-bit and 16-bit. In proposed design, Carry Save Adder (CSA) is used for addition process. CSA is one of the best adders in which carry of each input is saved and finally can be processed with the help of Ripple Carry Adder (RCA).

RESULTS AND DISCUSSION

The results of proposed pipelined DA based FIR filter were validated using Modelsim 6.3c and Synthesis results for pipelined DA based FIR filter were evaluated using Xilinx 12.4i (Family-Spartan 3, Package-Xc3s50, Speed grade is -5) design tool. The simulation result for pipelined DA based FIR filter is illustrated in Fig. 4. In this result, 4-bit input signals act as an address line of LUT and fixed pre-computed coefficient values stored in LUT act as a ROM. Results from coefficient values are finally added using Carry Save Adder (CSA). The comparison of proposed pipelined DA based FIR filter and traditional DA based FIR filter for variable lengths is illustrated in Table 3.

Table 3 shows that 8-bit Pipelined DA based FIR filter offers 15.38% reduction in silicon area and 9.43% reduction in delay. Similarly, 16-bit Pipelined DA based FIR filter offers 34.21% reduction in delay and 3.42% reduction in power consumption. In 8-bit Pipelined DA based FIR filter, more power has been consumed to implement the filtering operation. But in 16-bit Pipelined DA based FIR filter, power consumption has been reduced due to LUT partitioning,

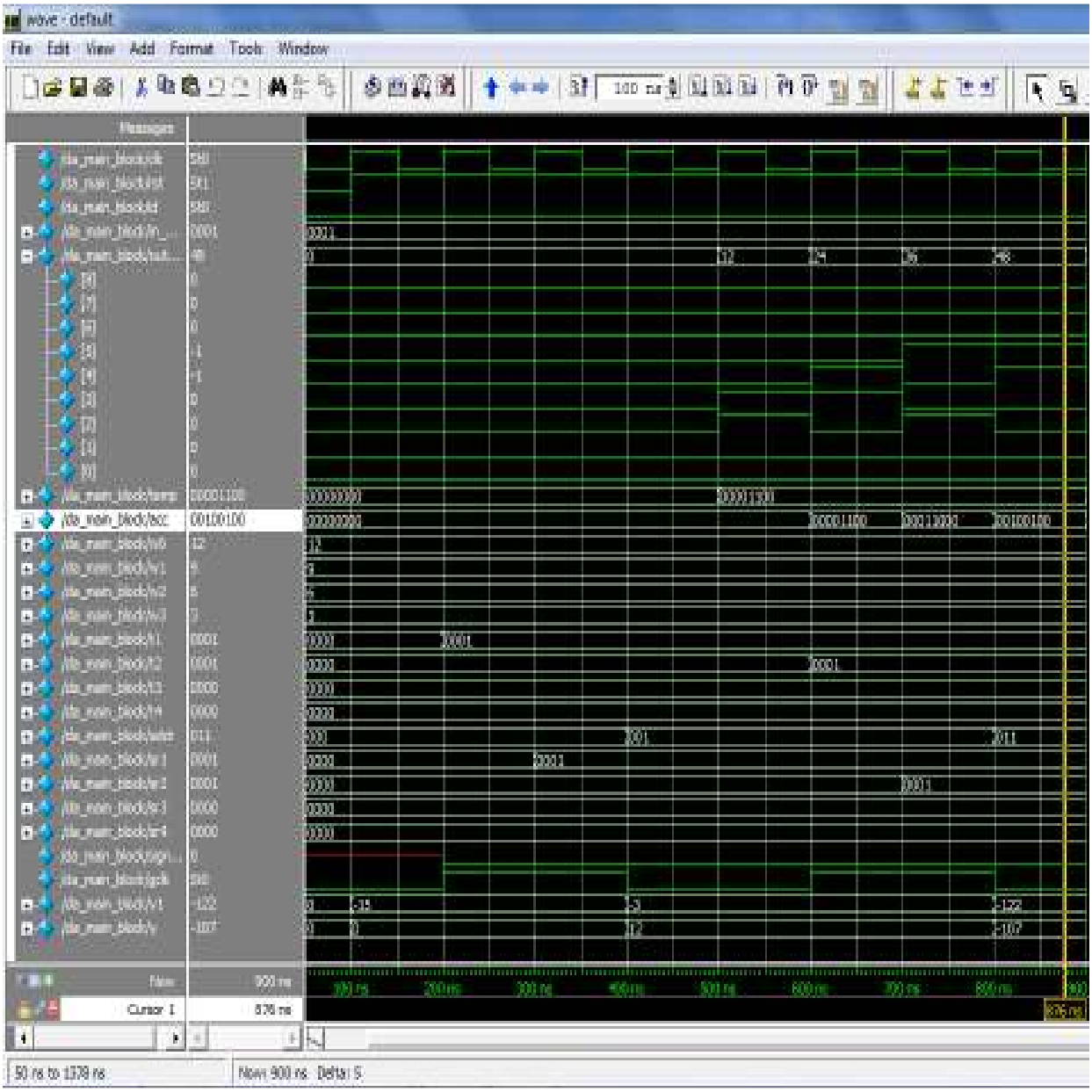

Fig. 4: Simulation results for pipelined DA based FIR filter implementation

instead silicon area is increased significantly. Hence, speed of FIR filter has improved when more number of bit lengths has used. Therefore, the proposed FIR filter design is absolutely suitable for high speed mobile telecommunication networking applications.

CONCLUSION

In this study, Pipelined DA based FIR filter is designed using Verilog Hardware Description Language (Verilog HDL) for wireless and mobile telecommunication networking applications. LUT is used to store the fixed coefficient values in DA based FIR filter. In order to reduce the memory size of the LUT, Offset Binary Coding (OBC) technique is used in this study. Further, Pipelining technique is used for improve the speed of DA based FIR filter. Performances of both traditional DA based and Pipelined DA based FIR filter are analyzed for variable lengths. 8-bit proposed Pipelined DA based FIR filter offers 15.38% reduction in silicon area and 9.43% reduction in delay whereas 16-bit proposed Pipelined DA based FIR filter offers 34.21% reduction in delay and 3.42% reduction in power consumption. Hence, this study concludes that speed of Pipelined DA based digital FIR filer is increased, if the input word-length of digital FIR filter is increased. In the future, proposed pipelined DA based digital FIR filter will be helpful for

wireless communication/Tele-communication and networking applications for control and communicate the data services and eliminate the unwanted noise in communication signals.

REFERENCES

Devipriya, M., V. Saravanan and N. Santhiyakumari, 2014. Power efficient and high throughput of FIR filter using block least mean square algorithm in FPGA. Int. J. Res. Eng. Technol. (IJRET), 03(02): 01-06.

Eshwararao, B. and V. Lokeshraju, 2012. Hardware realization of FIR filter implementation through FPGA. Int. J. Adv. Res. Electron. Commun. Eng. (IJARECE), 1(6): 25-30.

Murali, L., D. Chitra and T. Manigandan, 2014. Low power distributed arithmetic based fir filter. Middle-East J. Sci. Res., 22(4): 532-536.

Meher, P.K., C. Shrutisagar and A. Abbes, 2008. FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic. IEEE T. Signal Proces., 56(7): 3009-3017.

Meher, P.K., 2010. New approach to look-up-table design and memory-based realization of FIR digital filter. IEEE T. Circuits Syst., 57(3): 592-603.

Michal, S. and R. Mariusz, 2012. Application of modified distributed arithmetic concept in FIR filter implementations targeted at heterogeneous FPGAs. Przegkad Elektrotechniczny, 88(6): 240-246.

Pavel, F. and L. Richard, 2014. High performance VHDL FIR filter structure for symbol timing system implemented on FPGA. Proceeding of the IEEE Conference of Telecommunications, pp: 477-480.

Ramesh, R. and R. Nathiya, 2012. Realization of FIR filter using modified distributed arithmetic architecture. Signal Image Process. Int. J. (SIPIJ), 3(1): 83-94.

Sudhakar, V., N.S. Murthy and L. Anjaneyulu, 2012. Area efficient pipelined architecture for realization of FIR filter using distributed arithmetic. Proceeding of the International Conference on Industrial and Intelligent Information (ICIII, 2012), 31: 169-173.

Suhasini, M., K. Prabhu Kumar and P. Srinivas, 2012. Multiplier design and performance estimation with distributed arithmetic algorithm. Int. J. Comput. Commun. Technol. (IJCCT), 3(4): 07-12.

Toihria, I., R. Ayadi and M. Masmoudi, 2014. Modeling and simulation of a new BIST circuit with VHDL-AMS: Application for loop filter testing. Int. Rev. Modell. Simulat. (IREMOS), 7(4): 748-759.