

IF-THEN Adder Application in Online DALUT Implementation

Mohamed Al Mahdi Eshtawie and Masuri Othman
Institute of Microengineering and Nanoelectronics
National University of Malaysia, 43600 Bangi , Selangor, Malaysia

Abstract: This paper presents the application of a proposed if-then rule based adder in implementing distributed arithmetic online lookup table (DALUT). The online DALUT development and implementation is a continuation of our previous work where we proposed this idea and use it in designing finite impulse (FIR) filter. In our LUT architecture we have been able to overcome the major disadvantage of the basic DA architecture reported as the exponential growth of the LUT size with the number of input variables. The if-then adder was proposed in another work where it shows an efficient performance when compared with the well known ripple carry adder (RCA) and the carry lookahead adder (CLA). The online DALUT with the if-then adder was applied in designing 70-tap finite impulse response pulse shaping filter and some other different order FIR filters. The design was coded with Verilog hardware description language (verilog HDL) and synthesized using the Xilinx technology after being simulated with ModelSim 7.5g. The synthesis report shows that the design preference when using the if-then adder more efficient than in the case of the RCA and CLA adder. The maximum frequency reached with the design using the if-then adder was 85.095MHz, whereas, when using CLA and RCA adders it 77.936MHz and 77.042MHz respectively. Finally the design has been successfully downloaded to Virtex-II FPGA fg456 and tested with the TLAS201 logic analyzer.

Key words: Distributed Arithmetic, Online LUT, Xilinx Technology, FIR Filter, if-then adder.

INTRODUCTION

Distributed arithmetic is an efficient multiplierless technique used for computing inner product when one of the input vectors is fixed [1-5]. Basically, DA is an algorithm that performs multiplication with a look-up table based scheme. Hence it is computational operation that forms an inner (dot) product of a pair of vectors in a single direct step^[6]. For 4-tap FIR filter implemented with distributed arithmetic technique, the lookup table description is given in Fig. 1 below.

Figure 1 show clearly that the original DA architecture consists of three main parts i.e. the shift register unit, lookup table unit, and the shift and add unit. Here the LUT contains 16 locations i.e. 2^4 where 4 is the filter order. Therefore if we need to design and implement 70-tap FIR filter then the LUT size in this case will be 2^{70} . Therefore, if we let the LUT size to be (A) and the number of input variables N, then $(A=2^N)$. This unmasks the major disadvantage of the DA technique i.e. the exponential growth of the LUT size as the number of input variables increase.

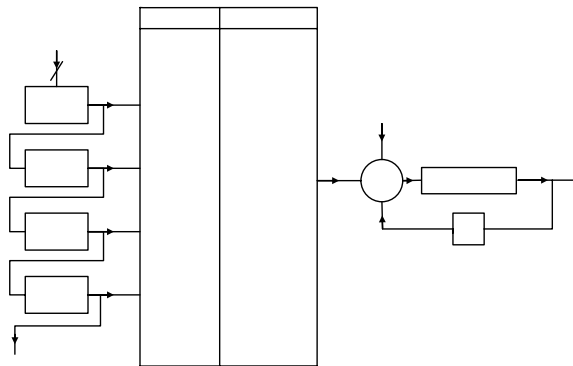


Fig. 1: Original Distributed Arithmetic Architecture [3]

Therefore, in^[6] we proposed a new architecture that we called online DALUT. By online architecture we mean that the needed LUT's locations contents are calculated while processing the input data. Needed locations mean that those locations pointed out with the

shift register unit output during processing the input data.

This is in controversy with the design shown in Fig. 1 above where the LUT is precalculated for all input values possibilities. Therefore, the LUT can be eliminated from the general DA architecture and the new structure will be as in Fig. 2 below.

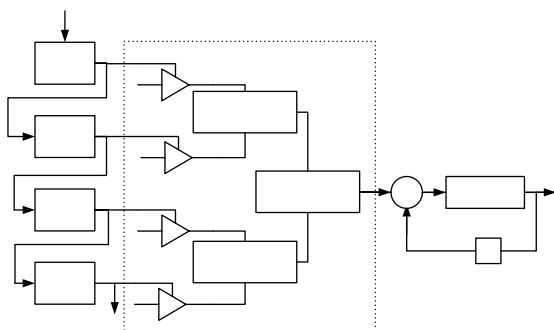


Fig. 2: The online DA LUT architecture^[6]

Serial distributed arithmetic (SDA) is a high speed multiplication technique based on serial processing at bit level ^[7]. The SDA can also be referred to as bit serial word parallel (BSWP) method since the output sample is available after its pipelined bits are processed. Therefore, its high computation rate is helpful for calculating the sum of products for filtering operations ^[8]. BSWP can also be represented as a vector multiplication where each word is converted into an equivalent number of bits and the bits are recorded in such a way that the arithmetic sum becomes distributed throughout the structure ^[9]

Mathematically, the DA can be overviewed with the basic sum of product equation for FIR filters as follows:

$$y = \sum_{k=1}^L h_k x_k \quad (1)$$

Where h_k are the filter coefficients and x_k are the input data words. Input data words can be represented in fractional form provided $|x_k| < 1$

$$x_k = -b_0 + \sum_{n=1}^{N-1} b_{kn} 2^{-n} \quad (2)$$

Where b_{kn} are bit 0 or 1, b_{k0} is a sign bit, $b_{k,N-1}$ is the LSB and N represents the number of bits in a digit or the digit word length. Substituting the value of x_k in Equation (1) we get:

$$y = \sum_{k=1}^L h_k \left[-b_0 + \sum_{n=1}^{N-1} b_{kn} 2^{-n} \right] \quad (3)$$

Equation (3) implies that the block of data bits has a dot product relationship with the filter coefficients. It can also be written as:

$$y = \sum_{n=1}^{N-1} \left[\sum_{k=1}^L h_k b_{kn} \right] 2^{-n} + \left[\sum_{k=1}^L h_k (-b_{k0}) \right] \quad (4)$$

Equation (4) shows that each data bit is multiplied with the filter coefficient bits. This multiplication is equivalent to AND operation.

Proposed Architecture: The architecture proposed in this paper applies the same concept of the online architecture presented in^[6]. However, in this work the if-then adder is used instead of the carry lookahead adder. The if-then rule based adder is designed with the main principle of the fuzzy logic IF-THEN rule base engine. Fuzzy logic IF-THEN rule based system means that the system made its decision with a logic flow that mimics or resembles the human being way of thinking. Figure 3 shows the new online LUT architecture after using the if-then adder.

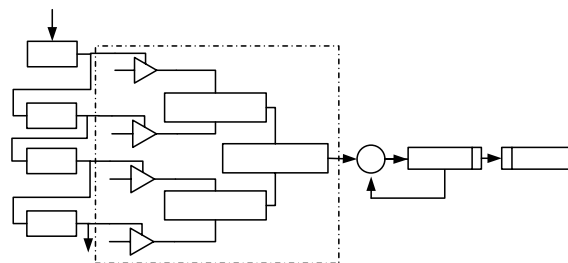


Fig 3: Online DA LUT Architecture with if-then Adder

Figure 3 shows that the input signal is fed to the system in serial fashion. The input word length 8 bits. Therefore after 8 clock pulses, the first input signal's value is loaded into the first register in the shift register unit. On the basis of the signal value, the first active location in the online LUT is derived. This address is produced after shifting the contents of all of the shift registers in the shift register unit to the right position by one bit. As a result of this shifting operation, and after another 8 clock pulses the second input signal's value is loaded into the first register whereas, the first input signal's value is now exist in the second register. On this way of processing the incoming input signal values will be processed. Each new input signal value loaded to the first shift register in the shift register unit. The corresponding filter output according to Equation (1). The partial sums or results obtained with each input

X[n-0]
674

ENB

C₁

X[n-2]

ENB

Carry Look-ahead
Adder

the shift and add unit. The contents of the accumulator are shifted to the right 1 bit at a time and the first output bit from this register is the final result's LSB. The value remained in the accumulator is added to the new partial sum obtained from the online lookup table based on the second input signal bit's value. The second bit shifted out from the accumulator is placed in (LSB+1) bit of the final result's value. When the input signal's MSB bit value is processed, the partial sum obtained from the online LUT is converted to its two's complement equivalent and on the same way is added to the contents of the accumulator register. At this point the contents of the result register together with the accumulator register forms the final filter output value. Figure 4 shows the system block diagram for this arrangement.

It is clear from Fig. 4 that the data out register is used to keep the value obtained from the LUT. On the other hand the value remained in the sum register i.e. after shifting its contents to the right one bit is loaded to the temporary register. In order to get the correct filter final output value, the contents of the final result register must be shifted to the right each time it is loaded with new bit. The total number of shifts will be equal to the register length.

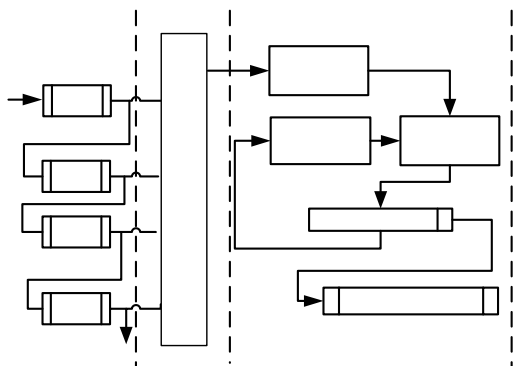


Fig 4: System Block Diagram

RESULTS

The functionality, precision and efficiency of the new online LUT architecture is verified and compared with a same work in this area. 70-tap FIR raised cosine pulse shaping filter design and implementation is one of the ways used to prove the efficiency of the new design and comparing it with the usages of two other different adders. The RCA and the CLA have also been used as arithmetic units in constructing the online LUT and the system performance for the three different cases is examined from the output correctness in one hand and from some other parameters such as speed and area of the design in another hand. The functionality and

precision are verified for each design aspect using the ModelSim 5.7g. For each of them, the expected correct filter's output is obtained. Figure 5 shows the waveform obtained when simulating the design with the ModelSim 5.7g.

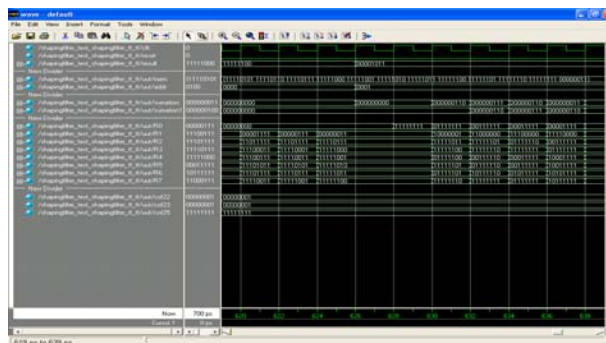


Fig. 5: The output waveform obtained from the ModelSim 5.7g

The first division of Fig. 5 shows the standard output of the unit under test (UUT). It shows the clock waveform, reset signal and the filter output value. However, the second division shows the address and the data stored in the memory used to feed the filter with its input. The third division shows the contents of the summation register. This value is calculated based on each new address got from the shift register unit. Some of the shift register unit registers are shown in the fourth division whereas; three of the filter coefficients are presented in the fifth division.

The results obtained from the Xilinx synthesis report, map report, and the place and rout report are summarized in table 1.

Figure 6 shows the RTL general schematic diagram of the designed system generated after synthesizing. The system clock is supplied from a clock divider generator so that we will be able to notice the output change at the FPGA LED output indicators

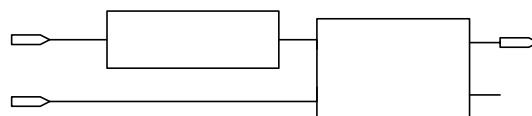


Fig. 6: System Schematic diagram

The TLA5201 logic analyzer has also been used to examine the functionality of the designed filter with its if-then adder. However for this experiment we designed 8-tap FIR filter and downloaded to the Xilinx Virtex II fg456 FPGA proto board. The simulation results obtained from ModelSim is shown in Fig. 7.

Table 1: results obtained when synthesizing and implementing the design with Xilinx technology

Type of Adder	Maximum Frequency	Delay of 10 worst nets	Equivalent gate count	Occupied slices	Average connection delay	Maximum pin delay
RCA	77.042MHz	2.952 nsec	11543	770	0.989	5.252 nsec
CLA	77.936MHz	2.737 nsec	10852	714	0.978	5.039 nsec
If-Then Adder	85.095MHz	2.542 nsec	11118	724	0.828	3.146 nsec

Table 2: Time parameters for different filters with the different adders

Adder type	Parameter	Filter Order	64	32	16	8
RCA CLA If-then adder	Max. frequency		112.404	117.171	141.463	152.155
			117.646	113.636	146.800	167.238
			119.782	116.939	146.617	176.991
RCA CLA If-then adder	Average connection delay		1.091	1.056	1.045	0.922
			1.036	1.046	1.083	0.923
			1.042	1.042	1.027	0.932
RCA CLA If-then adder	Max pin delay		2.761	2.995	5.733	3.123
			3.148	3.328	6.240	3.254
			3.005	3.080	5.932	3.612
RCA CLA If-then adder	Average connection delay on the 10 worst net		1.645	1.935	2.015	2.003
			1.617	1.944	2.254	1.707
			1.681	1.847	2.116	1.926

Table 3: Area parameters for different filters with the different adders

Adder type	Design score	64	32	16	8	
RCA CLA If-then adder		142	144	145	132	
		136	143	153	126	
		138	141	145	132	
RCA CLA If-then adder	Total gate count		70638	69,688	68,358	67,812
			70620	69,658	68,406	67,754
			70407	69,592	68,442	67,906
RCA CLA If-then adder	Occupied Slices		454	255	163	129
			451	253	167	122
			421	247	172	140
RCA CLA If-then adder	4 input LUTs		193	164	145	118
			194	159	129	134
			198	148	135	131



Fig. 7: Output waveform obtained from the ModelSim 5.7g

On the other hand, the output waveform obtained from the design analyzer is given in Fig. 8 below. In this case the maximum frequency reached with the design when simulated using ModelSim is 102.981MHz. However the

waveform shown in Fig. 8 which show the same results as Fig. 7 is achieved with a frequency 90MHz. Since both of the waveforms give the same result, this means that the system at a frequency 90MHz is still stable and accurate. This is one of the signs ensure that the designed system can be implemented as VLSI chip with a desired stable performance. Figure 9 shows an snapshot of the hardware connection of the TLA5201 logic analyzer to the Xilinx Virtex II fg456 FPGA proto board.

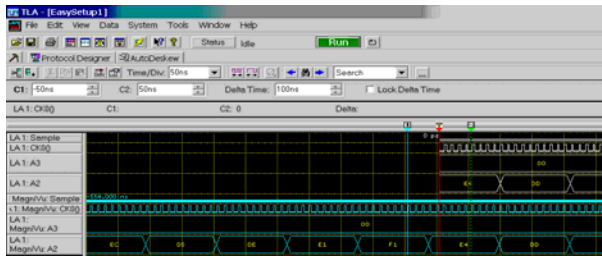


Fig. 8: shows the hardware connection for the FPGA and the logic analyzer.



Fig. 9: The TLA5201 logic analyzer used to test the system

In this work we have also used the online LUT with its if-then adder, and the other two adders i.e. CLA and the RCA to design and implement a number of different orders FIR raised cosine filter. The result obtained from synthesizing and implementing these different designs is summarized in Table 2 and 3 below. Table 2 shows the results concerned with the time parameters starting with the maximum frequency reached by each design which is also shown in Fig. 10.

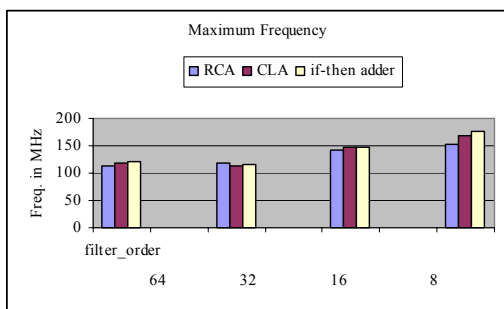


Fig. 10: Maximum Frequency reached by each design

DISCUSSION

When discussing the results obtained after simulating the design of the 70-tap raised cosine pulse shaping filter using the ModelSim 5.7g, performing the synthesis with XST technology and downloading the design to FPGA and examining the output with the

logic analyzer we found that: the filter output obtained from the ModelSim when using the different adders is the same and is the expected correct result. Furthermore, after synthesizing the design with XST technology and performing the design implementation, the results achieved from the place and rout and the map report show that the maximum frequency reached with the design when using the if-then adder is 85.095MHz, whereas when using the RCA and the CLA the maximum frequency reached is 77.042MHz and 77.936MHz respectively. Table 1 also shows that parameters related to the design speed are more optimized in the case of if-then adder than in the other two cases. In contrast, the equivalent gate count and the occupied slices for this design when using the if-then adder is greater than the case when using the CLA adder but less than when using the RCA.

The results presented in Table 2 and 3 show that we did the design for four other FIR filter i.e. 8-tap, 16, 32, and 64-tap and we recognized that the results for the if-then adder competes the other two adders and is better in most of the cases.

CONCLUSION

In this work we present the application of a proposed if-then rule based adder in designing and implementing FIR filter based distributed arithmetic online lookup table technique. The concept of this adder is to use the fuzzy logic rule based engine. The logic flow applied to get the adder output resembles the way of human thinking in this aspect. The filter output obtained when simulating the designed system is the expected correct result. The 70-tap raised cosine pulse shaping filter with its if-then adder has the advantage of high speed frequency than the case when using the RCA and the CLA adder. In addition, 8-tap FIR filter is designed and is successfully downloaded to Xilinx Virtex II fg456 FPGA proto board and examined with the TLA5201 logic analyzer. The maximum frequency achieved when simulating this filter with ModelSim is 102.981MHz whereas after downloaded to the FPGA and examined with the logic analyzer the same output waveform is obtained when the operating frequency is 90MHz.

REFERENCES

1. Sungwook Yu and Earl E Swartzlander Jr. 2001 "DCT Implementation with distributed arithmetic". IEEE transaction on Computers, Vol. 50 No. 9, September 2001.

2. A. Peled and B. Liu, 1974 A new hardware Realization of Digital Filters, IEEE Transactions. Acoustics, Speed and Signal Processing, vol. 22, pp: 456-462
3. Stanley A. White 1989. Applications of distributed arithmetic to digital signal processing: A tutorial review, IEEE ASSP magazine.
4. W.P. Burlison and L. L. Scharf, 1991 A VLSI Design Methodology for Distributed Arithmetic, J. VLSI Signal Processing, vol. 2 pp: 235-252.
5. S. Wolter, A. Schubert, H. Matz, and R. Laur 1993. On the Comparison between Architectures for the Implementation of Distributed Arithmetic,” Proc. IEEE Int’l Symp. Circuits and Systems, vol. 3 pp: 1829-1832.
6. Mohamed A. Eshtawie and Masuri Othman, 2006. On-Line DA-LUT Architecture for High-Speed High-Order Digital FIR Filters, paper status is published in the tenth IEEE international conference on communication systems (IEEE ICCS 2006), Singapore.
7. T. Salim, J. Devlin, and J Whittington, 2004. FPGA Implementation of Digital Upconversion using Distributed Arithmetic FIR Filters,” ICFPT, pp:335-338
8. The Role of Distributed Arithmetic in FPGA Based Signal Processing, Xilinx Publications.
9. K. K. Parhi, 1999. VLSI Digital Signal Processing Systems: Design and Implementation,” Published by John Wiley and Sons.