Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 7 (19): 2908-2910, 2007
ISSN 1812-5654

Design of Digital FIR Filter Based on Dynamic Distributed Arithmetic Algorithm

T. Vigneswaran and P. Subbarami Reddy
Department of ECE, SRM University, Chennai-603203, India

Abstract: This research presents a method for implementing high speed Finite Impulse Response (FIR) filters using just adders, Look Up Tables (LUTs) and shifters. The extensive use of a Dynamic Distributed Arithmetic (DDA) algorithm eliminates the multiplier unit which requires more number of adders. Xilinx Spartan III devices is used for optimization. It is observed that up to 56.75% reduction in the number of slices, upto 75% reduction in flip flops and up to 53.2% reduction in the number of LUTs is achieved. The speed of the DDA is improved by 31%.

Key words: Distributed arithmetic, FIR filter, multiplier less arithmetic unit, high speed, VLSI

INTRODUCTION

Digital signal processing algorithms are increasingly employed in modern wireless communications and multimedia consumer electronics, such as cellular telephones and digital cameras. Traditionally, such algorithms are implemented using programmable DSP chips for low-rate applications (Smith, 1997), or VLSI Application Specific Integrated Circuits (ASICs) for higher rates (Seals and Whapshott, 1997), or using adders, shifters and LUTs (Wirthlin, 2004). However, advancements in Field Programmable Gate Arrays (FPGAs) provide a new vital option for the efficient implementation of the DSP algorithms (Seals and Whapshott, 1997). FPGAs are being increasingly used for a variety of computationally intensive applications, mainly in the realm of Digital Signal Processing (DSP) and communications (Underwood and Hemmert, 2004). Due to rapid increases in the technology, current generation of FPGAs contain a very high number of Configurable Logic Blocks (CLBs) and are becoming more feasible for implementing a wide range of applications (Zhuo and Prasanna, 2005).

MATERIALS AND METHODS

The output of an N tap FIR filter, which is the convolution of the latest L input samples, is given in Eq. 1. L is the number of coefficients h(k) of the filter and x(n) represents the input time series.

$$Y[n] = \Sigma\, h[k]\, X[n-k]\ k = 0, 1, ..., N-1 \qquad (1)$$

The conventional tapped delay line realization of this inner product is shown in Fig. 1. This implementation

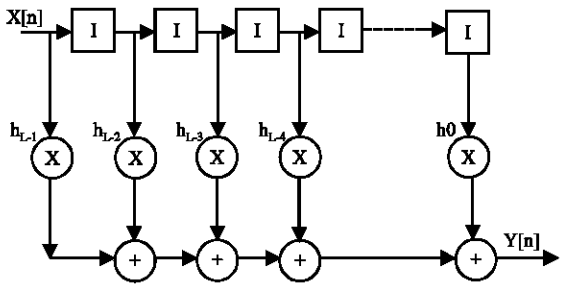

Fig. 1: L-Tap FIR filter

translates to L multiplications and L-1 additions per sample to compute the result. This can be implemented using a single Multiply Accumulate (MAC) engine, but it would require L-MAC cycles, before the next input sample can be processed. A general purpose multiplier occupies a large area on FPGAs (Yokota *et al*., 2002). Since all the multiplications are with constants, the full flexibility of a general purpose multiplier is not required and the area can be vastly reduced using techniques developed for constant multiplication like distributed algorithm. Though most of the current generation FPGAs such as. Xilinx Spartan III have embedded multipliers to handle these multiplications, the number of these multipliers is typically limited. The ideal implementation would involve a sharing of the Combinational Logic Blocks (CLBs) and these multipliers.

In this research, we present a Dynamic Distributed Arithmetic (DDA) technique that is better than conventional techniques for implementation on the CLBs. DDA which is a well known method to save resources like slices, flipflops and LUTs. The DDA architectures make extensive use of look-up tables, which make them ideal for implementing digital signal processing functions on Xilinx

Corresponding Author: Mr. T. Vigneswaran, Department of ECE, SRM University, Chennai-603203, India

FPGAs, whose architectures are based on look-up tables. Moreover, distributed architectures are suitable for low power portable applications, because they replace the costly multipliers with shifts and look-up tables (Ali and Haj, 2005).

Assuming coefficients C[n] are known constants, Eq. 1 can be rewritten as follows:

$$y[n] = \Sigma C[n] \cdot X[n] \quad n = 0, 1, \ldots, N-1 \qquad (2)$$

Variable X[n] can be represented by:

$$X[n] = \Sigma X_b[n] \cdot 2^b \quad b = 0, 1, \ldots, B-1 \qquad (3)$$
$$X_b[n] \in [0, 1]$$

Where, $X_b[n]$ is the bth bit of X[n] and B is the input width.

Finally, the inner product can be rewritten as follows:

$$\begin{aligned} y &= \Sigma C[n] \Sigma X_b[k] \cdot 2^b \\ &= C[0](X_{B-1}[0]2^{B-1} + X_{B-2}[0]2^{B-2} + \ldots + X_0[0]2^0) \\ &+ C[1](X_{B-1}[1]2^{B-1} + X_{B-2}[1]2^{B-2} + \ldots + X_0[1]2^0) \\ &+ \ldots \\ &+ C[N-1](X_{B-1}[N-1]2^{B-1} + X_{B-2}[0]2^{B-2} + \ldots + X_0[N-1]2^0) \\ &= (C[0]X_{B-1}[0] + C[1]X_{B-1}[1] + \ldots + C[N-1]X_{B-1}[N-1])2^{B-1} \\ &+ (C[0]X_{B-2}[0] + C[1]X_{B-2}[1] + \ldots + C[N-1]X_{B-2}[N-1])2^{B-2} + \ldots \\ &+ (C[0]X_0[0] + C[1]X_0[1] + \ldots + C[N-1]X_0[N-1])2^0 \end{aligned}$$

$$y = \Sigma 2^b \Sigma C[n] \cdot X_b[k] \qquad (4)$$

Where:
n = 0, 1… N-1 and
b = 0, 1… B-1

The coefficients in most of the DSP applications for the multiply accumulate operation are constants. The partial products are obtained by multiplying the coefficients C_i by multiplying one bit of data x_i at a time in AND operation. These partial products should be added and the result depends only on the outputs of the input shift registers. The AND functions and adders can be replaced by Look Up Tables (LUTs) that gives the partial product.

Proposed work: The proposed Dynamic Distributed Arithmetic (DDA) Algorithm is suitable for designing digital fir filter with varying co-efficient as compared to the conventional Distributed Arithmetic Algorithm (DAA) based fir filter design where the filter co-efficient

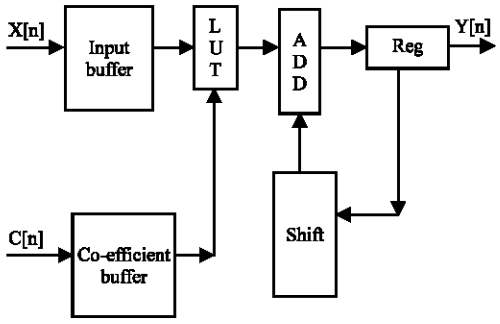

Fig. 2: Block diagram of dynamic distributed algorithm based FIR filter

are constant. Whenever there is a change in filter co-efficient, the co-efficient buffer updated dynamically and then performs the defined operations. In this research, the input sequence is fed into the input buffer register at the input sample rate. The co-efficient are also fed to the corresponding buffer. The LUTs are updated whenever there is change in C(n) values through buffer. The serial output is presented to the RAM based shift registers The RAM based shift register stores the data in a particular address. The outputs of registered LUTs are added and loaded to the scaling accumulator from LSB to MSB and the result which is the filter output will be accumulated on to the output register over the time. For an n bit input, n+1 clock cycles are needed for a symmetrical filter to generate the output. if there is any change in h[n], it will be updated and the resultant content is stored in the LUTs (Fig. 2).

Xilinx Spartan III are programmed using Verilog HDL; a popular hardware description language (Palnitkar, 1996). The language has capabilities to describe the behavioral nature of a design, the data flow of a design, a design's structural composition, delays and a waveform generation mechanism. Models written in this language can be verified using a Verilog simulator. As a programming and development environment, Xilinx ISE Foundation Series tools have been used to produce a physical implementation for the Xilinx Spartan III.

RESULTS AND DISCUSSION

The goal of our experiments was to compare the number of resources consumed by the DDA method with that produced by other conventional methods. For our experiments, we considered 4 tap FIR filters and targeted the Xilinx Spartan III device. The constants were normalized to 4 digit of precision and the input samples were assumed to be 4 bits wide. For the DDA method, we decomposed all the constant multiplications into

Table 1: Resources utilization for the various 4-Tap digital FIR filters

Method/parameters	Direct	Braun	Wallace	Array	DDA
Slices (3584)	51	74	41	69	32
Flip flops (7168)	16	16	14	16	4
LUT (7168)	24	126	62	118	59
IOB (141)	29	29	67	29	9
MUX	4	22	31	22	46

Table 2: In-built device utilization of 4-Tap digital FIR filters using Xilinx Spartan III

Method/parameters	Direct	Braun	Wallace	Array	DDA
Registers	8	8	7	8	0
Adders/sub tractors	6	6	5	6	35
Multipliers	8	0	0	0	0
Shifters	0	0	0	0	31

Table 3: Delay comparison of various types of various 4-Tap digital FIR filter design

Method/parameters	Direct	Braun	Wallace	Array	DDA
Delay (ns)	14.8	17.2	16.8	19.7	13.2

Table 4: Memory utilization of different methods of various 4-Tap digital FIR filter design

Method/parameters	Direct	Braun	Wallace	Array	DDA
Memory (MB)	74.5	70.6	71.5	71.5	68.7

additions, LUTs and shifts and optimized the expressions using the same algorithm. We used the Xilinx Integrated Software Environment (ISE) for performing synthesis and implementation of the designs. All the designs were synthesized for maximum performance.

The reduction in the number of resources, in terms of the number of Slices, LUTs and the number of FFs. It is observed that the number of slices is reduced by 56.75% and LUTs by 53.2%. The number of flip flops is also reduced by 75%. The number of Input Output Block (IOB) requires is only 9 out of 141 (Table 1).

It is noted that the direct method only requires multiplier units which occupy more area than adder unit. All other methods requires only adder-subtracter and shifters. Except the DDA based filter, all other filters utilize in built registers that reduces the speed of the filters (Table 2).

It shows that the delay of the filter implemented by the DDA is reduced by 31% than others (Table 3).

It is observed that the total memory required for the DDA is reduced by 8.22% than other types of implementations (Table 4).

CONCLUSION

This research work presented a multiplier less technique, based on the DDA method for low area and high speed implementations of FIR filters. The validation carried out over Xilinx Spartan III devices where we observed significant speed improvement and area reductions over traditional methods.

REFERENCES

Ali, M. and A. Haj, 2005. An FPGA-based parallel distributed arithmetic implementation of the 1-D discrete wavelet transform. J. Inform., 29: 241-247.

Palnitkar, S., 1996. Verilog HDL. SunSoft Press.

Seals, R. and G. Whapshott, 1997. Programmable Logic: PLDs and FPGAs. Macmillan. UK.

Smith, M., 1997. Application-specific integrated circuits. Addison Wesley Longman. USA.

Underwood, K.D. and K.S. Hemmert, 2004. Closing the Gap: CPU and FPGA trends in sustainable floating-point BLAS performance. Proceeding of International Symposium on Field-Programmable Custom Computing Machines, California, USA.

Wirthlin, M.J., 2004. Constant coefficient multiplication using look-up tables. J. VLSI Signal Proc., 36: 7-15.

Yokota, T., M. Nagafuchi, Y. Mekada, T. Yoshinaga, K. Ootsu and T. Baba, 2002. A Scalable FPGA-Based Custom Computing Machine for Medical Image Processing. Proceeding of International Symposium on Field-Programmable Custom Computing Machines (FCCM).

Zhuo, L. and V.K. Prasanna, 2005. Sparse matrix-vector multiplication on FPGAs. Proceeding of International Symposium on Field Programmable Gate Arrays (FPGA), Monterey, CA.