Chapter

Towards Optimised FPGA Realisation of Microprogrammed Control Unit Based FIR Filters

Syed Manzoor Qasim, Mohammed S. BenSaleh and Abdulfattah M. Obeid

Abstract

Finite impulse response (FIR) filter is one of the most common type of digital filter used in digital signal processing (DSP) applications. An FIR filter is usually realised in hardware using multipliers, adders and registers. Field programmable gate arrays (FPGAs) have been widely explored for the hardware realisation of FIR filters using different algorithms and techniques. One such technique that has recently gained considerable attention is the use of microprogrammed control unit (MPCU) in designing FIR filters. In this chapter, we further explore MPCU technique for optimised hardware realisation of digital FIR filter. To evaluate the performance, two different architectures of FIR filter are designed using Wallace tree multiplier. Both the architectures are coded in Verilog hardware description language (HDL). The performance is analysed by evaluating the resource utilisation and timing reports of Virtex-5 FPGA generated by the Synopsys Synplify Pro tool. Based on the implementation results, as compared to conventional design, Wallace tree multiplier using carry skip adder (CSKA) provides optimal digital FIR filter.

Keywords: carry skip adder, field programmable gate array (FPGA), FIR filter, microprogrammed control unit, Wallace tree multiplier

1. Introduction

Digital filters play an important role in many digital signal processing (DSP) applications. These applications range from noise reduction, spectral shaping, equalisation, signal detection and signal analysis, etc. The basic building blocks of digital filter are adder, multiplier and register based delay elements. Based on the application requirement, these blocks are connected to realise a particular architecture of filter. There are several ways to realise digital filters. Two such filters used in different applications are finite impulse response (FIR) and infinite impulse response (IIR) filters. FIR filters are widely preferred for DSP applications because they are always stable, exhibit linear phase properties and provide no feedback. Convolution, the core operation of FIR filter, performed on a window of *N* data samples involves multiplication and addition. For optimal realisation of FIR filter, these arithmetic operation needs to be optimised.

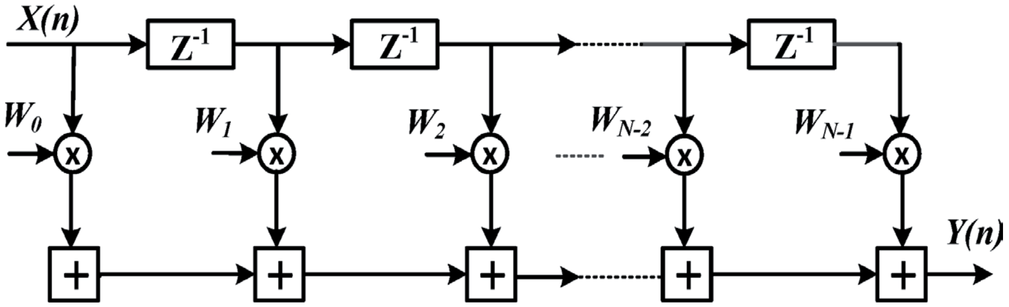

Figure 1.
N-tap direct form FIR filter.

Direct form is the most commonly used FIR filter. As can be seen from **Figure 1**, *N*-tap or (*N-1*)th order FIR filter consist of *N* multipliers, *N-1* adders and *N* shift registers. The tap coefficients, $\{W_0, W_1, W_2,......,W_{N-1}\}$ constitute the filter impulse response. The filter type (low pass, high pass or band pass) is determined by these coefficients.

Different techniques for the field programmable gate array (FPGA) realisation of FIR filter using microprogrammed control unit (MPCU) have been reported in the literature [1–3]. Multipliers and adders play a dominant role in the optimal realisation of FIR filters [4, 5]. The objective of this chapter is to further explore this technique using Wallace tree multiplier with different adder configurations for optimal realisation of FIR filter [6]. The proposed design is modular and scalable which enables realisation of higher-order FIR filter.

The rest of the chapter is organised as follows. Section 2 presents two different designs of MPCU-based FIR filters. Section 3 describes the design of Wallace tree multiplier using two different adder configuration. Section 4 presents the FPGA implementation results and its analysis. Finally, Section 5 concludes the chapter.

2. MPCU based FIR filter architectures

The FIR filter top-level module as shown in **Figure 2** consists of a datapath unit and a control unit. The control unit is realised using the microprogrammed approach. The MPCU consist of two main parts, the first part addresses the microinstructions stored in the control memory while the second part holds and generates microinstruction for the datapath unit [1–3].

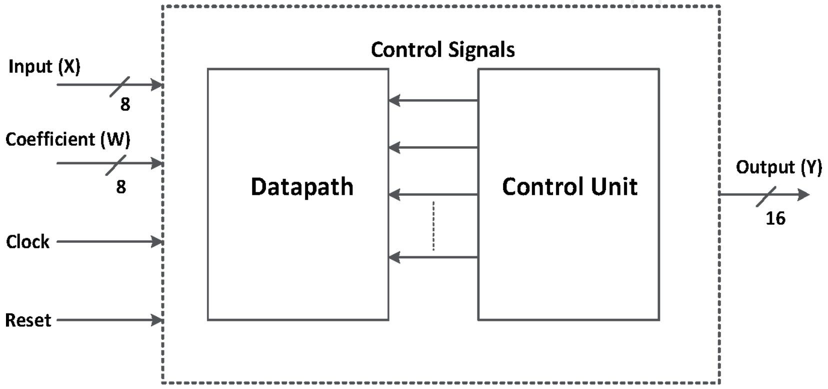

Figure 2.
FIR filter module.

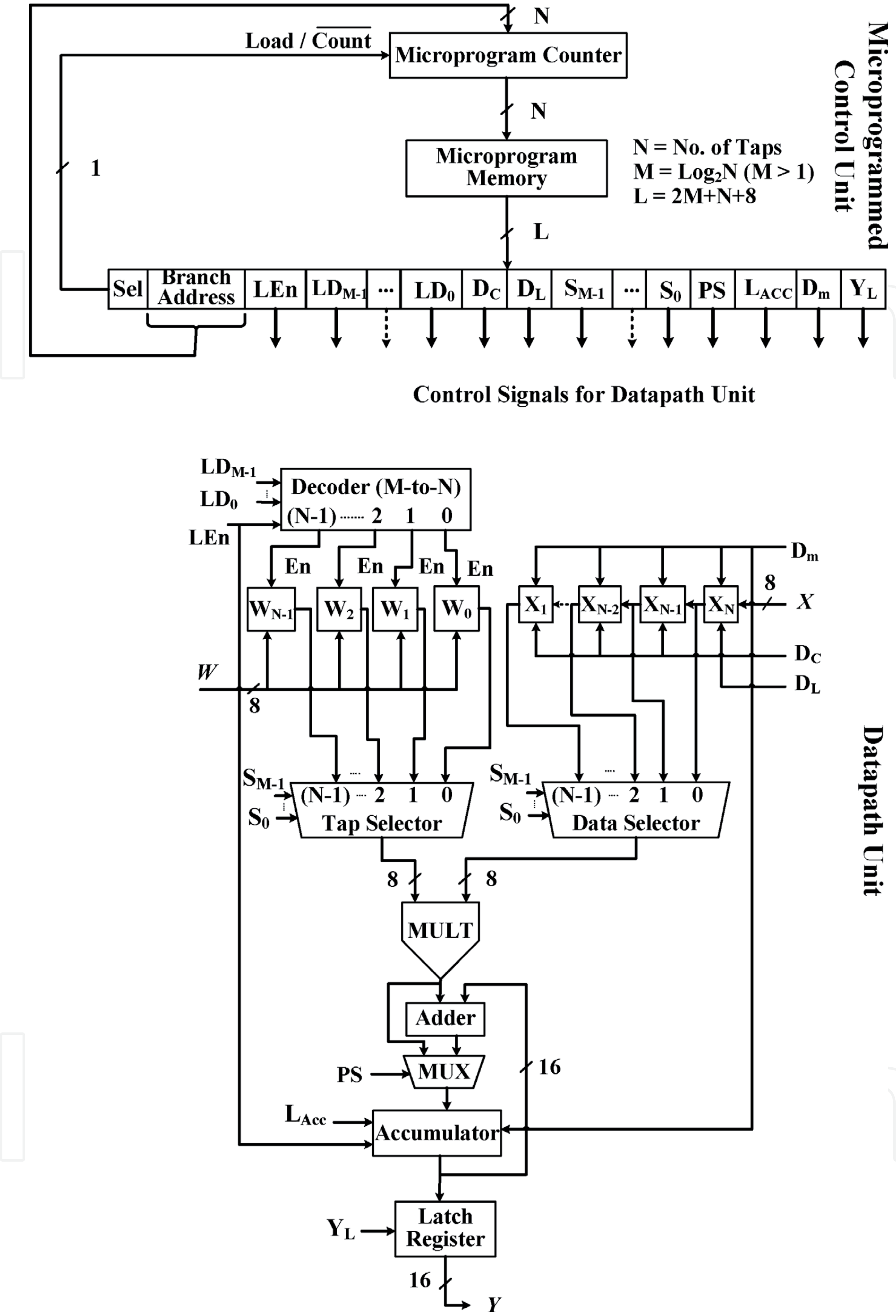

Figure 3.
First architecture of FIR filter [1].

The first architecture of N-tap FIR filter as shown in **Figure 3** comprises of a control and datapath units. The control signals generated by MPCU are fed to the datapath unit. For demonstration, the sequence of operation for a third-order FIR filter is listed in **Table 1** [1].

No.	CS	Branch address	Control functions						
			LEn	LD_1	LD_0	D_C	D_L	D_m	Y_L
1	0	0 0 0 0	1	0	0	0	0	0	0
2	0	0 0 0 0	1	0	1	0	0	0	0
3	0	0 0 0 0	1	1	0	0	0	0	0
4	0	0 0 0 0	1	1	1	1	0	0	0
5	0	0 0 0 0	0	0	0	0	1	0	0
6	0	0 0 0 0	0	0	0	0	0	1	0
7	0	0 0 0 0	0	0	0	0	0	0	1
8	1	0 1 0 0	0	0	0	0	0	0	0

Table 1.
Control signals for third-order FIR filter (Architecture-1).

The datapath consist of the following modules:

- 8-bit data registers
- 8-bit coefficient registers
- *N*-to-*M* decoder
- *N*:1 multiplexer (MUX) for data selection
- *N*:1 MUX for tap selection
- Multiplier
- Adder
- 2:1 MUX for dataflow control
- 16-bit accumulator
- 16-bit latch register.

For (*N-1*)th order FIR filter, the datapath unit of second architecture uses *N* multipliers and *N-1* adders. In addition to the multiplier and adder, the datapath also need the following modules for proper functioning of FIR filter as illustrated in **Figure 4**.

- 8-bit data registers
- 8-bit coefficient registers
- *M*-to-*N* decoder
- One 16-bit latch register

For illustration, the sequence of operation for third-order FIR filter in this case is listed in **Table 2** [2].

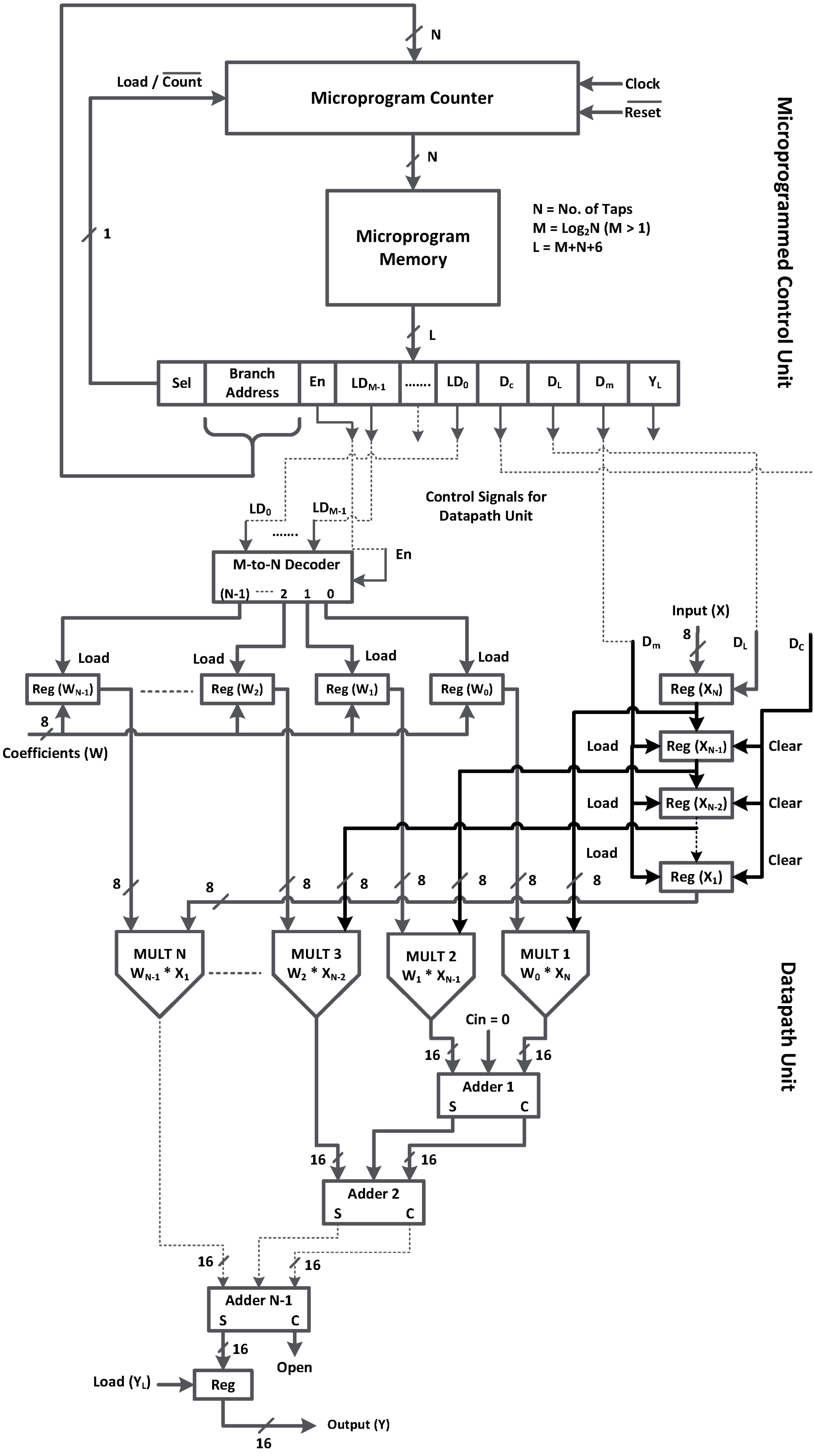

Figure 4.
Second architecture of FIR filter [1].

No.	CS	Branch address				LE	LD$_1$	LD$_0$	Dc	D$_L$	S$_1$	S$_0$	Ps	Lacc	D$_m$	Y$_L$
1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
3	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
7	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
8	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0
9	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
12	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0

Table 2.
Control signals for third-order FIR filter (Architecture-2).

3. Wallace tree multiplier

To overcome the drawbacks associated with conventional array multiplier, tree multiplier is considered. Wallace tree is one such implementation of adder tree that results in high speed. A conventional Wallace tree multiplier uses half and full adders to multiply two numbers in three steps as shown in **Figure 5** [4]. First step is to multiply each bit of n-bit multiplicand with every bit of n-bit multiplier to yield n^2 results. Each bit carry different weights based on the position of the generated bits. The second step involves reduction of partial products using full and half adders. This process continues until two layer of partial products remain. In the last step, the remaining two layers of partial product are added using conventional adder [5].

In this chapter, two different variants of Wallace tree multiplier are realised. First variant uses conventional full and half adders, while a carry skip adder (CSKA) is used in the second variant.

Carry look ahead adder (CLA) provides high-speed computation but at the cost of high power and high area. To overcome the drawbacks of CLA, CSKA is used

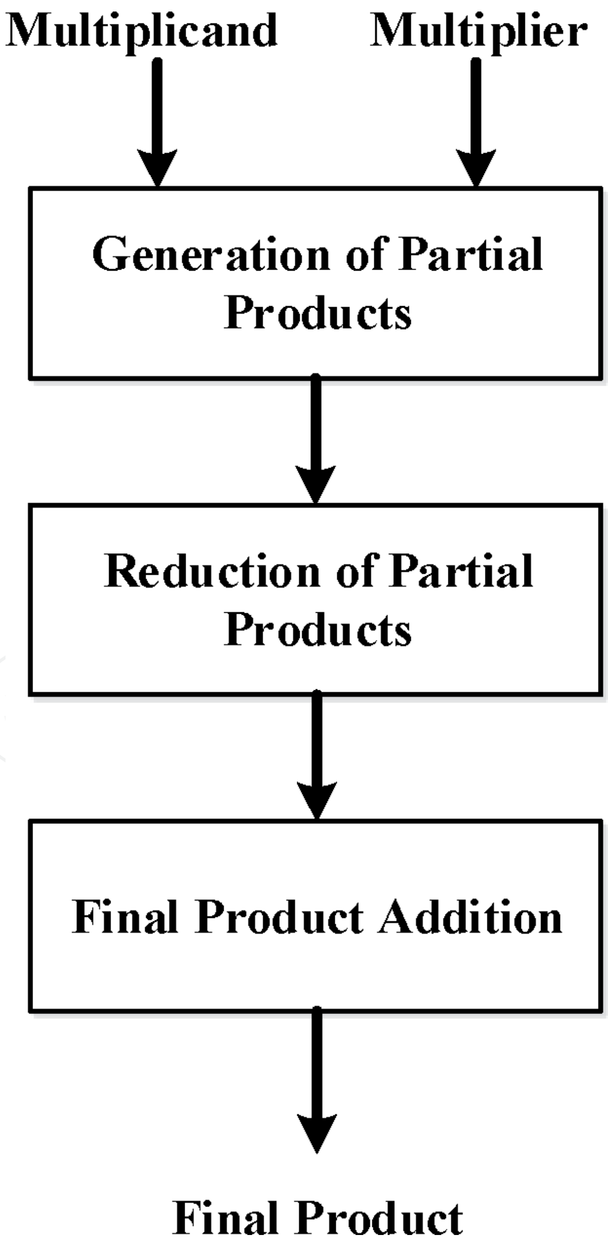

Figure 5.
Block diagram of Wallace tree multiplier.

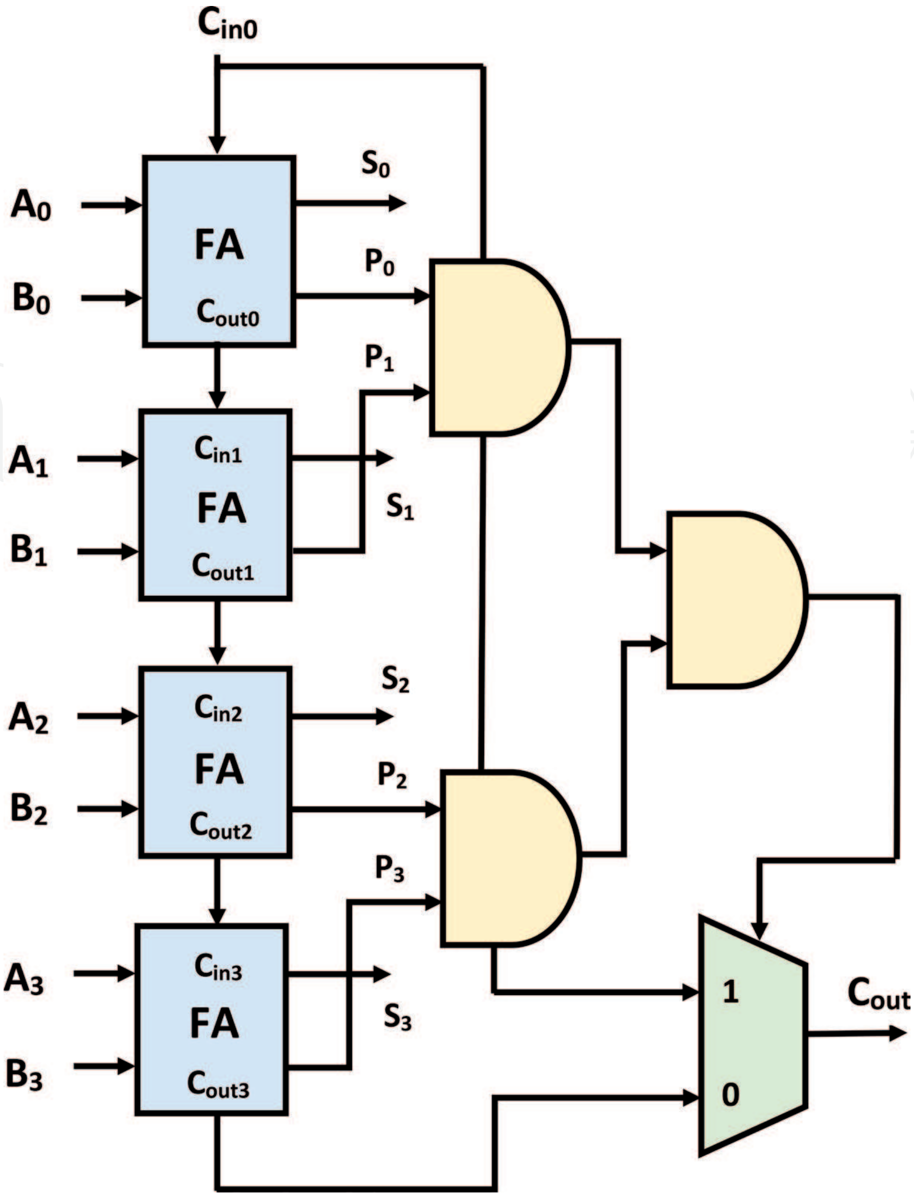

Figure 6.
Block diagram of 4-bit CSKA.

which provides a balanced implementation [5]. A CSKA comprises of a basic ripple carry adder with a distinctive speed-up carry chain referred to as a skip chain. As shown in **Figure 6**, a skip chain comprises of AND gate and 2:1 MUX.

4. Results and analysis

All the top-level modules and sub-modules described in this chapter are coded in Verilog HDL using top-down hierarchical design methodology. The proposed designs are synthesised and implemented in Virtex-5 (xc5vlx50t-1ff1136) FPGA device using Synplify pro electronic design automation (EDA) tool [7]. The results are evaluated based on the slice look-up tables (LUTs), minimum period and maximum clock frequency of the target FPGA. **Tables 3** and **4** summarise the implementation results for both the FIR filter architectures.

It can be inferred that, generally, the Wallace tree multiplier using conventional full and half adder consumes less FPGA slice LUT (area) but at the cost of higher minimum period (delay). In the first architecture, Wallace tree multiplier using CSKA

No. of taps	Filter order	Wallace tree using FA/HA			Wallace tree using CSKA		
		Slice LUTs	Min. period (ns)	Max. freq. (MHz)	Slice LUTs	Min. period (ns)	Max. freq. (MHz)
4	3	147	12.093	82.7	142	10.143	98.6
8	7	181	12.479	80.1	147	11.448	87.4
16	15	184	12.166	82.2	180	10.491	85.3
32	31	212	12.616	79.3	155	10.300	97.1
64	63	209	12.319	81.2	154	11.044	90.5

Table 3.
FPGA resource utilisation for first FIR filter using Wallace tree multiplier.

No. of taps	Filter order	Wallace tree using FA/HA			Wallace tree using CSKA		
		Slice LUTs	Min. period (ns)	Max. freq. (MHz)	Slice LUTs	Min. period (ns)	Max. freq. (MHz)
4	3	278	13.876	72.1	284	11.026	90.7
8	7	687	19.756	50.6	699	17.552	57.0
16	15	1268	31.662	31.6	1371	29.150	34.3
32	31	2304	54.787	18.3	1670	52.377	19.1
64	63	4352	101.39	9.9	3319	98.678	10.1

Table 4.
FPGA resource utilisation for second FIR filter using Wallace tree multiplier.

has the lowest minimum period. It is therefore concluded that the first FIR filter architecture using Wallace tree multiplier with CSKA provides optimal result. It is also observed that more FPGA resources are utilised as we increase the order of the filter.

5. Conclusion

In this chapter, we further explored the design of MPCU-based digital FIR filters. MPCU is a promising technique that could be utilised for optimal realisation of digital filters used in DSP systems. The overall performance of the FIR filter depends on the multiplier and adder used in the multiply-accumulate unit. Two different architectures of FIR filter were designed using Wallace tree multiplier employing two variants of adder, one using conventional full/half adders and the other using CSKA. All the designs were realised in Xilinx Virtex-5 FPGA using Synplify pro EDA tool. Based on the reports generated by the EDA tool, it is concluded that the design of first FIR filter using the Wallace tree multiplier with CSKA provides optimal result in comparison to the one using conventional full and half adders.

Acknowledgements

The authors gratefully acknowledge the support provided by King Abdulaziz City for Science and Technology (KACST) under the National Electronics, Communication and Photonics research program.

Author details

Syed Manzoor Qasim*, Mohammed S. BenSaleh and Abdulfattah M. Obeid
National Center for MEMS Technology, King Abdulaziz City for Science and Technology (KACST), Riyadh, Saudi Arabia

*Address all correspondence to: syed.manzoor@ieee.org

References

[1] BenSaleh MS, Qasim SM, AlJuffri AA, Obeid AM. Scalable design of microprogrammed digital FIR filter for sensor processing subsystem. IEICE Electronics Express. 2014;**11**:1-7. DOI: 10.1587/elex.11.20140474

[2] Qasim SM, BenSaleh MS. Hardware implementation of microprogrammed controller based digital FIR filter. IAENG Transactions on Engineering Technologies. 2014;**247**:29-40. DOI: 10.1007/978-94-007-6818-5_3

[3] Qasim SM, BenSaleh MS, Bahaidarah M, AlObaisi H, AlSharif T, Alzahrani M, et al. Design and FPGA implementation of sequential digital FIR filter using microprogrammed controller. In: Proceedings of IV IEEE International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT'12); 3-5 October 2012. St. Petersburg, Russia: IEEE; 2012. pp. 1002-1005

[4] Waters R, Swartzlander E. A reduced complexity Wallace multiplier reduction. IEEE Transactions on Computers. 2010;**59**:1134-1137. DOI: 10.1109/TC.2010.103

[5] Xing S, Yu WWH. FPGA adders: Performance evaluation and optimal design. IEEE Design and Test of Computers. 1998;**15**:24-29. DOI: 10.1109/54.655179

[6] Aljuffri AA, Badawi AS, BenSaleh MS, Obeid AM, Qasim SM. FPGA implementation of scalable microprogrammed FIR filter architectures using Wallace tree and Vedic multipliers. In: Proceedings of 3rd IEEE International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE'15); 1 May 2015. Beirut, Lebanon: IEEE; 2015. pp. 155-158

[7] Cosoroaba A. Achieve higher performance with Virtex-5 FPGAs. XCell. 2006;**59**:16-18