TRANSPOSED FORM OF FOLDED FIR FILTER

K. Subramanian[1], Dr. R. Prema[2], S. Muthukrishnan[3]

[1-3]*Dept. of Electronics and Communication Systems, Karpagam Academy of Higher Education, Coimbatore, Tamilnadu, India.*

ABSTRACT

The designing method of folded finite-impulse response (FIR) filter on pipelined array based multiplier arrays is presented in this paper. The design is considered at the bit-level of the pipelined multiplier array and internal delays are fully exploited in order to reduce power consumption and hardware complexity, transposed FIR filter forms is considered. The proposed schemes are compared as to the aspect of hardware complexity with a straightforward implementation of a folded FIR filter based on the pipelined Wallace Tree multiplier. The comparison reveals that the proposed schemes requirement is very less hardware. Finally, the efficient implementation of folded FIR filter circuits is presented when constraints in area, clock frequency and power consumption.

Keywords—*Filter, Folded, Transposed, FIR, Adder, Multiplier, Pipeline, Array.*

.

1. INTRODUCTION

Today's, digital signal processing (DSP) is used in a vital area of real-time applications of an important role in the digital revolution. Finite-impulse response (FIR) of digital filters is the most fundamental DSP components. FIR filters have the advantage of easy implementation and stability but the large number of filter taps leads to excessive hardware complexity. Power consumption is an important factor in DSP circuits, especially when used in wireless communication systems.

Folding technique has been proposed to reduce the hardware when the processing throughput required by the application is less than the throughputs at which the circuit can be operate. FIR filters are ideal candidates for folding since they are essentially a repetition addition of multiplications. Reconfigurable folded transposed form FIR filter architecture has been presented in but not detailed implementation is given, although the bit-level design can lead to hardware efficient circuits.

A significant advantage of the FIR folded architectures is that they lead to reduced hardware in comparison with the corresponding unfolded. Thus there is no need to resort to hardware expensive asynchronous solutions.

In this paper, we propose a folded scheme based on pipelined arrays of multiplier for the implementation of FIR filters in transposed form. Arrays of multipliers have canonic structure and can achieve high throughputs since they can be pipelined at the bit by bit and the exploitation of the internal pipelining delays of the array for storing data words and filter coefficients and the result is significant hardware reduction.

The Wallace Tree multiplier pipelined at the bit-level has lower latency than the array multipliers but it does not allow the internal pipelining delays to be exploited. The design of direct-transposed form is considered. The detailed designed circuits based on possible combinations of filter forms and array types are presented by comparing the results from theaspect of hardware complexity. A way to combine the merits of folded and unfolded filters is to cascade a number of folded FIR filter units.

The partially folded filter is an intermediate form between the folded and unfolded form of a filter, featuring higher throughput than the folded and requiring less hardware than the unfolded. In this work the effect of the number of cascaded filter units in the area, power consumption and operation frequency has been studied. A technique for the optimal selection of the number of cascaded filter stages is presented which takes into account the design limitations of the requested filter.

2. VLSI AND DEALS WITH CIRCUITS

VLSI stands for "Very Large Scale Integration". This is the field which involves packing logic devices into smaller and smaller areas This is opened up a wide opportunity to do things that were not possible before. VLSI circuits are everywhere car, brand new state-of-the-art digital camera, computer, the cell-phones. In the electronic market, VLSI has driver towards a very high integration growing. While it grownups on a chip increases, critical concerns arises regarding the power consumption and size of the components on the chip. As CMOS technology scales down, process variation introduces significant uncertainty in power and performance to VLSI circuits and significantly affects their reliability.

VLSI has been around for a long time, there is nothing new about it, but as a side effect of advances in the world of computers, there has been a dramatic proliferation of tools that can be used to design VLSI circuits. Alongside, obeying Moore's law, the capability of an IC has increased exponentially over the years, in terms of computation power, utilization of available area, yield. The combined effect of these two advances is that people can now put diverse functionality into the IC's, opening up new frontiers. Examples are embedded systems, where intelligent devices are put inside everyday objects, and ubiquitous computing where small computing devices proliferate to such an extent that even the shoes you wear may actually do something useful like monitoring your heartbeats! These two fields are kind a related, and getting into their description can easily lead to another article. Digital VLSI circuits are predominantly CMOS based device. The way normal blocks like gates and latches are implemented is different from what have seen so far, but the work process remains the same.

3. FINITE IMPULSE RESPONSE FILTER

A filter removes the unwanted parts of the signal in signal processing, like as random noise, or extracts the parts of the signal, like as the components lying within a certain frequency range. There are many instances in which an input signal to a system contain extra unnecessary additional noise or content, which can be degrade the quality of the desired portion. In such case may remove or filter out the useless samples. In this way allowed the desired frequency of pass-band and blocked undesired frequency of stop-band.

Finite Impulse Response, filter is one of the primary types of filters used in DSP. FIR filter is said to be finite because they do not have any feedback in the system. If send an impulse through the system (a single spike) then the output will invariably become zero as soon as the impulse runs through the filter.

3.1. EXISTING FIR FILTER STRUCTURES

The transpose-form structures are most commonly used to implement FIR filters. For certain special filters, recursive implementations require less computation. Lattice and cascade structures are occasionally also used.

There are no closed loops (no feedback) in this structure, so it is called a non-recursive structure. Since any FIR filter can be implemented using the direct-form, non-recursive structure, it is always possible to implement an FIR filter non-recursively. However, it is also possible to implement an FIR filter *recursively*, and for some special sets of FIR filter coefficients this is much more efficient.

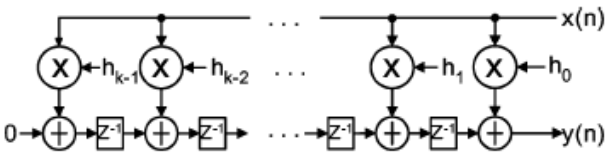

Fig. 3.1 FIR – Transposed Form

The flow-graph-reversal theorem says that if one changes the directions of all the arrows, and inputs at the output and takes the output from the input of a reversed flow-graph, the new system has an identical input-output relationship to the original flow-graph.

4. PIPELINED FIR FILTER ARCHITECTURES

4.1. UNFOLDED FIR FILTER ARCHITECTURES

In Fig.4.1, the structure of an unfolded FIR filter in transposed form is shown. The corresponding folded filter architecture is shown in Fig. 4.2.

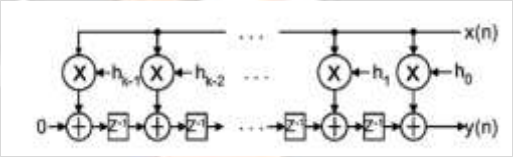

Fig. 4.1. Structure of an unfolded k-tap FIR filter in transposed form.

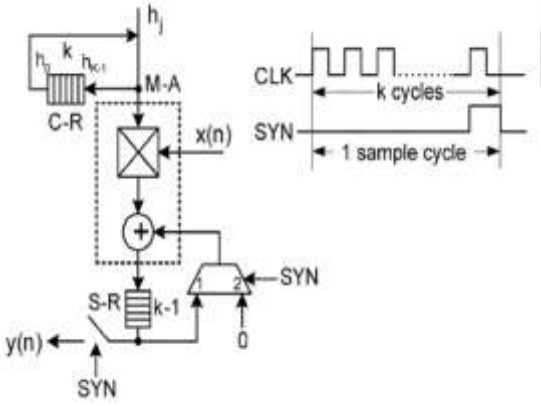

Fig. 4.2. Folded architecture of a k-tap FIR filter in transposed form.

The basic difference between direct and transposed unfolded forms is the position of the delay elements. In the second scheme are located in the sum line instead of the input data line. This difference is also reflected in the corresponding folded architectures. The timing diagram shown in Fig. 4.4 clarifies the operation of the circuit. The computation cycle lasts k clock cycles. The input sample x(n) enters the circuit at the first clock of the operation cycle and remains constant during the next k clocks of the same cycle.

Thus, the sum computed at the first clock cycle is the final result of the current computation cycle but appears at the output during the last clock of the same computation cycle because it shifts through the C-R. Thus, this scheme is also of immediate response.

4.2. FOLDED FIR FILTER TRANSPOSED FORM CODING

```
library IEEE;

use IEEE.STD_LOGIC_1164.ALL;
 use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_SIGNED.ALL;
entity transpose_fold is Port ( clk,reset: in std_logic; synch:in std_logic;

xin : in std_logic_vector(3 downto 0); yout : out std_logic_vector(7
downto 0)

);
end transpose_fold;

architecture bhv of transpose_fold is

signal add:std_logic_vector(7 downto 0):=(others=>'0'); signal mul :std_logic_vector(7 downto 0):=(others=>'0');
--signal r0,r1,r2,r3,r4,r5,r6,r7:std_logic_vector(3 downto 0):=(others=>'0'); signal hi:std_logic_vector(3 downto
0):=(others=>'0');
type coff is array(7 downto 0) of std_logic_vector(0 to 3); signal coff_rom:coff:=
                        (0 => "0001", 1 => "0010", 2 => "0011", 3 => "0100", 4 => "0101", 5 => "0110", 6 => "0111", 7
                         => "1000"
                        );
begin
process(clk,coff_rom,reset)

--variable hi:std_logic_vector(3 downto 0);
begin
   if clk'event and clk='1' then if reset='1' then
mul<=(others=>'0');

add<=(others=>'0');
hi<=(others=>'0'); else
coff_rom<=coff_rom(0) & coff_rom(7 downto 1); hi<=coff_rom(0);
mul<=xin * hi; add<=add + mul; end if; --yout<=add; end if;
end process; process(clk,synch) begin
if synch='1' then yout<=add; else
yout<="00000000"; end if;
end process;

end bhv;
```

Structure of an unfolded k-tap FIR filter in direct-transposed form

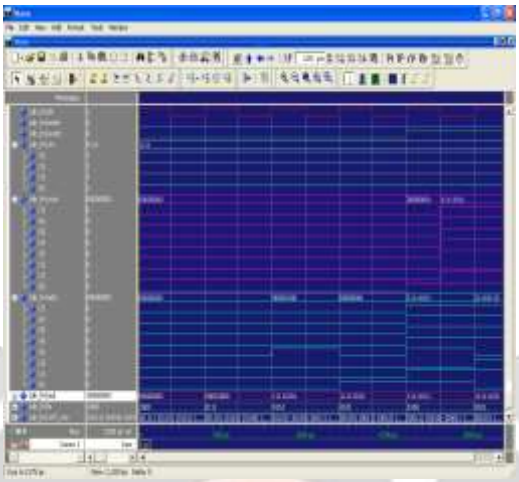

Fig.1 Output Waveform for Direct-transposed Form Architecture

Folded architecture of a k-tap FIR filter in transposed form

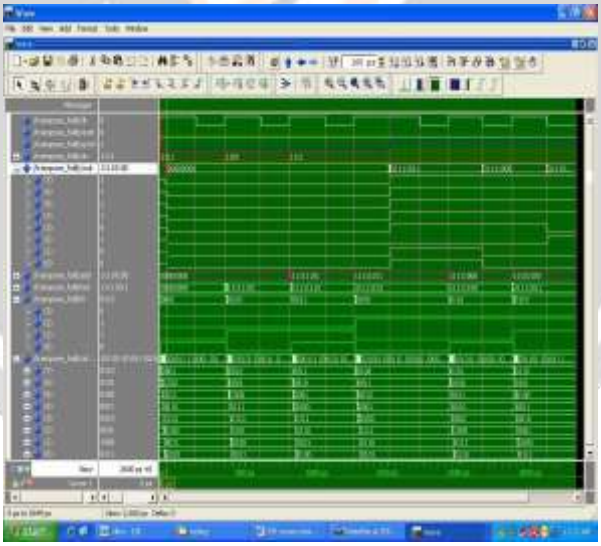

Fig. 2 Output Waveform for Direct Form Folded Architecture

5. CONCLUSION

In this paper, high throughput folded FIR filters schemes based on bit-level pipelined multiplier arrays are proposed. The suggested exploitation of the internal pipelining registers has less hardware than a straightforward implementation based on a Wallace Tree multiplier. The proposed folded FIR filters can be cascaded to implement partially folded FIR filters. In this case, the choice of the most suitable of the proposed schemes depends on the number of filter taps and the requirements for the operational frequency and hardware complexity. For an implementation under speed and transistors number constraints the designer can make use of graphs that permit the

optimal choice of the degree of folding and the architecture of the proposed schemes.The above ideas can be also applied to the implementation of folded IIR filters. The operation of all the circuits presented in this project has been verified by extensive simulation in Xilinx Spartron3E.

6. REFERENCES

[1] M. Kahrs et al., "The past, present and future of audio signal processing," IEEE Signal Process. Mag., vol. 14, no. 5, pp. 30–57, Sep. 1997.

[2] T. H. Meng, A. C. Hung, E. K. Tsen, and B. M. Gordon, "Low-power signal processing system design for wireless applications," IEEE Pers. Commun., vol. 5, no. 3, pp. 20–31, Jun. 1998.

[3] B. G. Haskell, P. G. Howard, Y. A. LeCun, A. Puri, J. Ostermann, M. R. Civanlar, L. Rabiner, L. Bottou, and P. Haffner, "Image and video coding–Emerging standards and beyond," IEEE Trans. Circuits Syst. Video Technol., vol. 8, no. 7, pp. 814–837, Nov. 1998.

[4] P. G. Paulin, C. Liem, T. C. May, and S. Sutarwala, "DSP design Requirements for embedded systems: A telecommunications industrial perspective," J. VLSI Signal Process., vol. 9, pp. 23–47, Jan. 1995.

[5] A. V. Oppenheim and R. W. Schafer, Discrete-Time Signal Processing. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[6] K. Subramanian, "Implementation of Folded FIR Filter Direct Form Using VHDL Programming", IJSART,vol.2(10),pp.154-159,Oct. 2016.

[7] K. K. Parhi, C.-Y.Wang, and P. Brown, "Synthesis of control circuits in folded pipelining DSP architectures," IEEE J. Solid-State Circuits, vol. 27, no. 1, pp. 29–43, Jan. 1992.

[8] K. K. Parhi, "Calculations of minimum number of registers in arbitrary life time chart," IEEE Trans. Circuits Syst. II, vol. 41, no. 6, pp. 434–436, Jun. 1994.

[9] K. K. Parhi, VLSI Digital Signal Processing Systems: Design and Implementation. New York: Wiley-Interscience, 1999.

[10] S. Ramanathan, S. K. Nandy, and V. Visvanathan, "Reconfigurable filter coprocessor architecture for DSP applications," J. VLSI Signal Process., vol. 26, pp. 333–359, Nov. 2000.

[11] S. Hauck, "Asynchronous design methodologies: an overview," Proc. IEEE, vol. 83, no. 1, pp. 69–93, Jan. 1995.

[12] W. Kuang, J. S. Yuan, R. F. DeMara, D. Ferguson, and M. Hagedorn, "A delay-insensitive FIR Filter for DSP applications," in Proc. 9th Annu. NASA Symp. VLSI Design, Albuquerque, NM, Nov. 2000, pp. 135–165.

[13] S. Y. Kung, "On supercomputing with systolic/wavefront arrays," Proc. IEEE, vol. 72, no. 7, pp. 867–884, Jul. 1984.

[14] K. Subramanian, "Implementation of Folded FIR Filter Direct Form Using VHDL Programming", IJSART,vol.2(10),pp.154-159,Oct. 2016.

[15] C. R. Baugh and B. Wooley, "A two's complement parallel array Multiplication algorithm," IEEE Trans. Computers, vol. C-22, no. 12, pp.

[16] C. S. Wallace, "A suggestion for fast multiplier," IEEE Trans. Electron. Comput., vol. EC-13, pp. 14–17, Feb. 1964.

[17] N. H. E. Weste and K. Eshraghian, Principles of CMOS VLSI Design, 2nd ed. Burlington, MA: Addison-Wesley, 1993.

[18] K. Shanmugasundaram T. Karuppiah, K. Subramanian, N. Nandakumar, "Analysis and Study of Robotics in Different Types of Application",vol.3(2),pp.7530-7535,Feb.2014.

[19] R. Amirtharajah and A. P. Chandrakasan, "Self-powered low power signal processing," in IEEE Symp. VLSI Circuits Dig. Tech. Papers, Jun.1997, pp. 25–26.

[20] Dr. R. Prema, K. Subramanian, Mr. B. Senthilvel,"Automatic Aircraft Parking Systems",vol.3(3),pp.358-361, Apr. 2017.

[21] K. Subramanian, "A Study of Advanced Wireless Networks Technology Implementation", vol.5(2),pp.691-697,Feb.2016.