TEST Engineering & Management

November-December 2019
ISSN: 0193-4120 Page No. 5242 - 5245

Energy Efficient Reconfigurable Fir Filter Using Improved Carry-Bypass Adder on FPGA

Gali Madhavi, P. Jyothi
madhavireddy.jkc@gmail.com

Article Info
Volume 81
Page Number: 5242 - 5245
Publication Issue:
November-December 2019

Article History
Article Received: *5 March 2019*
Revised: *18 May 2019*
Accepted: *24 September 2019*
Publication: *25 December 2019*

Abstract

In recent years there is a tremendous change in the growth of multimedia applications, especially finite impulse response (FIR) filters due to its low power, high speed, and less expensive nature. A high-performance adder is one of the key components in the design of application-specific integrated circuits. In this paper, improved Carry-bypass adder reconfigurable FIR is introduced to perform the filter operations. In this paper, the main intention of the proposed adder is to reduce the carry propagation time by skipping the adder stages in higher stages that results in less power and area. To achieve faster computation and less overhead the introducing of critical path delay and countermeasures have been investigated to alleviate the issues and evaluated other traditional adders.

Keywords: *FIR, Carry Bypass Adder (CBA),FPGA, critical path delay.*

I. INTRODUCTION

Digital signal processing (DSP) is generally primarily used in a wide spectrum of concurrent applications. Digital finite-impulse response (FIR) filters are also one of the key handling components used primarily to improve strength. Digital FIR filter implementation is one of the simplest schemes relative to analog FIR filter design [1]. However, owing to the high amount of multipliers the current FIR filter requires a large area and power consumption.FIR filters are used primarily in mobile communication. Correspondingly, numerous systems are used to develop and implement FIR filters efficiently. The coefficient valuation is continuous in various continuous multiplication (MCM) technique. Before and after the runtime, we should not alter the coefficient significance.However, the coefficient quantity in the programmable change technique (PSM) can be changed during the runtime. During this job, the MCM method centered on binary common sub expression elimination (BCSE) with FFA is used to produce the distinct coefficient by re-use the change and adder amount. Power and pause are therefore smaller than all other standard methods [2].This method is only used for the procedure of a given filter. Because we need to simultaneously conduct more filters, we need to sequentially implement this filter. The gap is therefore large.The re-configurable FIR filter design suggested lowers additional costs and works with faster and supports less complex nature that can used in many signal processing application like filter equalization, matched filtering process, convolution, and several data converters applications. The suggested architecture is introduced and checked with the 8-bit, 12-bit and 16-bit filter inputs on a Spartan-3A family of field-programmable gate array (FPGA). The suggested new FIR-configurable multiplier block architecture provides an excellent region and velocity enhancement as opposed with current FIR-configurable filter designs.The main contributions and organization of this paper are summarized as follows:In section II we describe literature review of several adder FIR filtering techniques.The section III describes System model.The section IVResults and Discussion.Finally, in section V we concluded the paper.

II. RELATED WORKS

In [3], the authors showed a new pipeline design for a reduced energy, high performance and DA-based adjustable filter. With series LUT updates and simultaneous execution of media and weight procedure, the throughput rate of the FIR model was maximized. A rapid Bit-clock for Carry-Save Accumulation (CSA) was enhanced to reduce energy usage but has a much faster clock pace for all other activities.

In [4], the authors provided a high-performance, small capacity and small zone DA-based linear FIR filter. The LeastMean Square (LMS) method is utilized to adjust the suitable weights at the same time decrease the parameter mean square error (MSE) from the present and requested filteringresults. AlsoDA panel with pipeline reduces changing operation and reduces energy. The primary restriction of this article is that it focuses heavily on energy consumption.

In [5], the writers employed the techniquePipelined Modified Booth Multiplier (PMBM) meant forthe application of energy efficient RFIRfiltering. However, this technique is limited by the high delay due to the reduced device speed and output.In [6] the more efficient RFIR structural design was implemented using DA. However it is also evaluated the two types of structures and concluded the design of the direct type requires small quantity of registers relative as the system of the transposed type. Configurable DA-based FIR block filter offers scalability for large block dimensions and filter lengths. However, the restriction of this technique only addressed constructions for 4 blocks.

In [7], the writers launched new RFIR filter design centered on the centralized FIR filter architecture for the statistics.

As discussed earlier the FIR filter complexity and speed [8], wants its applications is based on electronic systems [9] forfilter and software-defined radio systems [10]. The laboratory findings have been evaluated by studying the efficiency parameters such as region, velocity and energy of high-order FIR models and found that the suggested RFIR models consuming less resource and power [11], do not address dynamically reconfigurable mechanisms, but improve their efficiency over standard FIR filters used in image and video processing units [12]. In this case communication devices which desires low-power and small area integrated devices [13].

III. SYSTEM MODEL

The FIR-based DA has been used for FPGA application using the CBA-RFIR system as depicted in Fig.1. The LUTs are evolved with the FPGA implementation of the DRAM. The various values of the temporary $S_{l,p}$products are concurrently extracted from the suitable memory as RAM, as it is a LUT quantity is obtained per each stage.

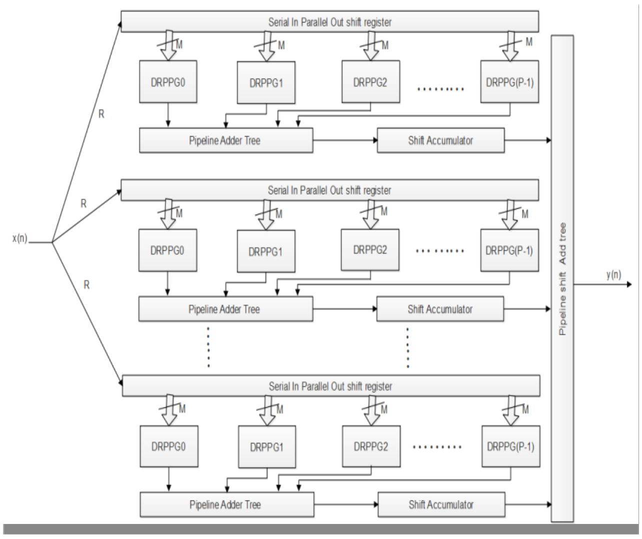

Fig. 1.The structure of reconfigurable FIR filter integrated with DRAM

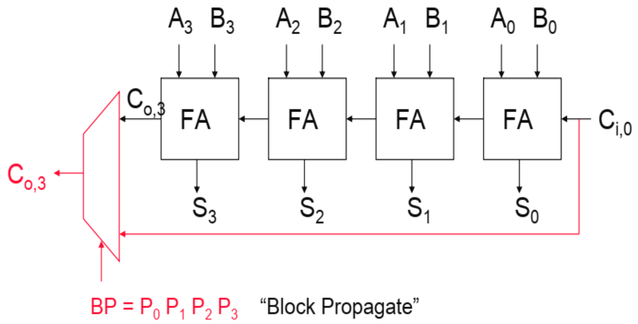

Fig. 2.Carry-Skip (Carry-Bypass) Adder

Fig. 2 illustrates Carry-Bypass Adder when (P_0&P_1&P_2&P_3 = 1) then $C_{0,3}$ = $C_{i,0}$ thenthe block the aforementioned kills or generates the carry internally.

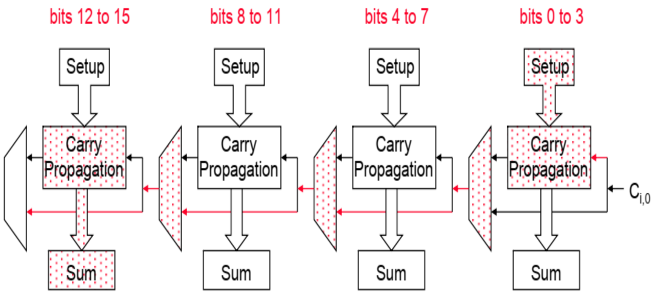

Fig. 3.A 4-bit Block Carry-Skip Adder

Fig. 3 illustrates 4-bit Block Carry-Skip Adderwith worst case delay→carry from bit 0 to bit 15 = carry produced by bit 0, flips through bits 1, 2, and 3, skips top two rows (B is bits in group size), flips from bit 12 to bit 15 in the last cluster.

Optimal Block Size and Time:

$$T_{add} = t_{setup} + B t_{carry} + \left(\left(\frac{N}{B}\right) - 1\right) t_{skip} + B t_{carry} + t_{sum} \quad (1)$$

Supposing one stage of ripple (t_{carry}) possess identical delayas one skip logic stage (t_{skip}) and equally are 1.

$$T_{CSkA} = 1 + D + \left(\frac{M}{D} - 1\right) + D + 1 \qquad (2)$$

t_{setup} ripple in skips ripple in t_{sum} block 0 last block

$$= 2D + \frac{M}{D} + 1 \qquad (3)$$

For compensation the block size is to be optimum that can be formulated as

$$l\,\frac{T_{CSkA}}{dB} = 0 \Rightarrow \sqrt{\left(\frac{M}{2}\right)} = D^{opt} \qquad (4)$$

To obtain the final time for case of optimal and the optimal time is

$$Optimal\ T_{CSkA} = 2\left(\sqrt{(2M)}\right) + 1 \qquad (5)$$

The main scenario for a Ripple-Carry Adder (RCA) is when the LSB is generating a performance and the whole adder is rippled from bit 0 to bit (*N*-1). One instance is 00000001 + 11111111. Bits 7-1 are "Propagators" in the adder terminology, and parts 0 is "Generators." The critical path is between the LSB carry-out and the MSB carry-out in turn each adder is in the critical path.

The concept behind a Carry-bypass Adder (CBA) is to shorten the duration of this critical path by providing a shortcut to the carry path if every bit in a row spreads the hold. A block-wide propagation frequency is quite simple to calculate, and each block can concurrently calculate its own propagation signal. The worst situation is still the same situation, but it appears a lot distinct what occurs. Consider the same 0000......001 + 0111..... 111 issue. The first stage calculates a carry in the first bit and propagate the carry through bits 1, 2, and 3. The first block carry-out signal is applicable at this stage. The propagated signal is already applicable since there are 2-3 gate delays and 4 gate delays. The carry-in multiplexer for bits 8-11 receives the carry signal from bit 3, because bits 4-7 would propagate a carry signal. It is clear that this requires 1 gate delay, and 4 gate delays are required for a standard RCA. Each unit adds 1 gate delay to the carry signal.

Uncertainty the MSB killed is carry propagation, the last CSA unit would rip the input, resulting in another 4 gate delay. The new worst situation is the configuration of an LSB generator and an MSB kill. The critical path source is the same from the RCA to the CSA, but the critical path is distinct. When an indefinite block generates a carry on its own, it will always propagate to the next block. However, if the second bank produces or kills by itself, then this is the start of the critical path. If the second section propagates the carry, the CSA architecture is of benefit to us. When using the word "critical path," it also usually means that we are choosing a number of outputs that trigger the worst-case error. The situations provide indicate "hideous" instances, but this is not the maximum error.

Let's say it have 3 stages in this by-pass adder. And each stage has 4 bits. If there is any bit generate or delete the carry, it will not need the carry from the 1st stage. In this case, it wouldn't need to wait the result of carry from the previous stage. So, the worst case happens when it should wait for the 1st stage carry. And let me put it in another way. The worst case delay = setup time + the whole carry time in the 1st stage + 2*by-pass time + (4-1)*carry time for each bit. The carry from input in propagate from the 1st bit to the last bit in the 1st stage. At the same time, the second stage is whether waiting for the carry from 1st stage or start to generate or delete its own carry. And the worst case would be waiting. So, the worst case is the second stage decides to propagate the 1st stage's carry.

Let's claim that this by-pass adder has three stages. And every stage has 4-bits. If the carry is generated or deleted, it does not need the carry from the first point. In this situation, the results of the transfer from the earlier point would not have to delay. The worst scenario occurs when the first stage is to be carried. The worst interval is setup time + totaltime in stage 1 + 2*by-pass period+ (4-1)* for each bit. The inputcarry from the first bit to the last bit in the first stage. At the same time, the second stage is whether we are waiting for the first stage to carry or start generating or deleting the own carry. And the worst situation is pending. The worst situation therefore is that the second stage chooses to propagate the first stage.

Critical path for Carry-Skip adder:

In a typical adder that mixes e.g. $X[3\ldots0]+Y[3..\ 0]+C_{in}$ to produce $Z[3..0]$ and C_{out} are all equal to $X[0]$, $Y[0]$, C_{in}; all of them will be given the same $C_{out\text{-}delay}$. If multiple adder stages are combined, the overall propagation time from the initial stages $X[0]$ and $Y[0]$ to their ends plus the times from the subsequent stages of $\mathrm{C_{in}}$ to their endings is the same. If we connect eight identical 4-bit sections, for example, then the time from $X[0]$ or $Y[0]$ to C_{out} will be 1/7 more than from C_{in} to C_{out}. Essentially, if all of the outputs come together, a ripple-carry adder would have three equal critical paths for the input from $X[0]$, $Y[0]$ and C_{in}. If the carry input stabilizes only a long time before every other input is done, it is the critical path with any combinational adder. A carry skip adder tries to minimize the time between stabilizing the input and stabilizing the output if the other inputs have remained stable for some time, but if all inputs become simultaneously stable, the critical paths from $X[0]$ and $Y[0]$ for the output remain the same. Moreover, a cascade-based adder can provide $O(\mathrm{sqrt}(N))$ moment spread with $O(N)$ circuits with higher order phases which are higher than shorter ones. If a N-bit carry-skip stage has a time of $2N+2$ gate time delayed for its X and Y input, and $2N$-bit carry-skip stage delayed for any input, then if a 32-bit adder from the 4-bit carry-skip stage, and 4, 5, 6, 7 or 8-bit carry-skip stages is built, the carry from the first stage stabilizes eight delays after$x$ and y. The N-bit carry-skip stage has a time of $2N+2$ delays. The second phase will maintain 10 delays after X and Y, or 2 periods after input (both occur together). The following phase stabilizes 12 units after X and Y, or 2 units after input, etc. This is not the quickest path to do more, but a good balance between speed and circuitry. We can decrease propagation time to $O(\log N)$, but this involves $O(N\log N$ circuit).

IV. RESULTS AND DISCUSSION

In this section the FPGA performance was analysed for several devices of Virtex-6, Virtex-7, by using Xilinx ISE design suite.

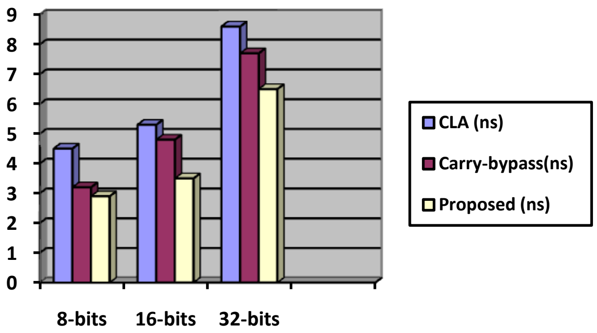

Fig. 4. Assessment of the proposed adder with other adders in terms of critical path delay in a typical speed

Fig.4 illustrates comparison of CLA, carry-bypass adder with our proposed adder for 8, 16 and 32-bit sizes. It is clear from the graph that proposed adder is having very less critical path delay due to its optimal skipping of carries at higher stages.

V. CONCLUSION

We have suggested in this paper a new adder that produces higher bits with a less logical duration. It thus has the improvement of higher computational speed and reducing delays. Further enhanced the efficiency by combining a regular shift/add accumulation with an improved conditional accumulation carry-bypass system for the filter output computation. With less hardware than other adders which just enhanced computational speed, we minimized both delays and power consumption. PDP, including both delay and power, is used as a comparative criteria. Using reconfigurable FIR filter integrated with DRAM on FPGA the proposed adder achieved 2.8ns, 3.9ns and 6.2ns delay which outperformed compared to conventional adders.

REFERENCES

[1] David A. Parker and Keshab K. Parhi “Area-Efficient Parallel FIR Digital Filter Implementations” 1063 6862/960. 1996 IEEE.

[2] Lavina Magdalene Mary.J and Dhanasekar.B, “Area Efficient Parallel Fir Digital Filter Structures Based On Fast FIR Algorithm” International Journal of Engineering Research and

Applications (IJERA) Vol. 3, Issue 1, January -February 2013, pp.2042-2046.

[3] S.Y. Park, and P.K. Meher, “Low-power, high throughput, and low-area adaptive FIR filter based on distributed arithmetic”, *IEEE Transactions on Circuits and Systems II: Express Briefs,* Vol.60, No.6, pp.346-350, 2013.

[4] S. Ramanathan, G. Anand, P. Reddy, and S.A. Sridevi, “Low Power Adaptive FIR Filter Based on Distributed Arithmetic”, *Int. Journal of Engineering Research and Applications,* Vol.6, No.5, pp.47-51, 2016.

[5] N. Sriram and J. Selvakumar, “A Reconfigurable FIR Filter Architecture to Trade off Filter Performance for Dynamic Power Consumption”, *Int. J. Adv. Comput. Theor.Eng. (IJACTE),* Vol.2, No.1, pp.112-119, 2013.

[6] K.M. Basant, P.K.Meher, S.K. Singhal, and M.N.S. Swamy, “A high-performance VLSI architecture for reconfigurable FIR using distributed arithmetic”, *Integration, the VLSI Journal,* Vol.54, pp.37-46, 2016.

[7] R. Jia, H.G. Yang, C.Y. Lin, R. Chen, X.G. Wang, and Z.H. Guo, “A Computationally Efficient Reconfigurable FIR Filter Architecture Based on Coefficient Occurrence Probability”, *IEEE Transactions on Computer-Aided Designof Integrated Circuits and Systems,* Vol.35, No.8, pp.1297-1308, 2016.

[8] A. Bonetti, A. Teman, P. Flatresse, and A. Burg, “Multipliers-Driven Perturbation of Coefficients for Low-Power Operation in Reconfigurable FIR Filters”, IEEE Transactions on Circuits and Systems I: Regular Papers, Vol.64, No.9, pp. 2388 – 2400, 2017.

[9] J. Chen, J. Tan, C.H. Chang, and F. Feng, “A new cost-aware sensitivity-driven algorithm for the design of FIR filters”, IEEE Transactions on Circuits and Systems I: Regular Papers, Vol.64, No.6, pp.1588-1598, 2017.

[10] N. Bhagyalakshmi, K.R. Rekha, and K.R. Nataraj, “Design and implementation of DA-based reconfigurable FIR digital filter on FPGA”, In: Proc. of International Conf. on Emerging Research in Electronics, Computer Science and Technology (ICERECT), pp.214-217, 2015.

[11] A. Liacha, A.K. Oudjida, F. Ferguene, M. Bakiri, and M.L. Berrandjia, “Design of high-speed, low-power, and area-efficient FIR filters”, IET Circuits, Devices & Systems, Vol.12, No.1, pp.1- 11, 2017.

[12] M. Alawad and M. Lin, “Fir filter based on stochastic computing with reconfigurable digital fabric”, In: Proc. of the International Conf. on Field-Programmable Custom Computing Machines (FCCM), pp.92-95, 2015.

[13] A. Rasekh and M.S. Bakhtiar, “Design of Low Power Low-Area Tunable Active RC Filters”, IEEE Transactions on Circuits and Systems II: Express Briefs, Vol.65, No.1, pp.6-10, 2018.

AUTHOR DETAILS

GALI MADHAVIis working as Asst.ProfDeptof ECE in Mallareddy institute of technology, secunderabad. She completed her undergraduate and postgraduate degrees from JNTUH University. Her research areas are VLSI system design and low power VLSI.

P.JYOTHIis working as Asst.ProfDeptof ECE in Pallavi engineering of College, Nagole. She completed her undergraduate and postgraduate degrees from JNTUH University. Her research areas are VLSI system design and low power VLSI.