Efficient Multiple Constant Convolution Circuit using Modified Parallel Pipelined Floating Point Addition

Pratibha Singh[1], Prof. Rishi Jha[2]
[1]*Mtech Student,* [2]*Guide*
Department of Electronics and Communication Engg. NIIST, Bhopal

Abstract - An programmable device can be designed and reconstruct for different operations which provide accurate timing and synchronization with concurrent execution of parallel threads, and rapid decision making. Elimination of the multiplication logic is the foremost idea among researchers to reduce the complexity of the logic operations, and this idea to implement is having vary much crucial design to make circuit work. Instead of removing multiplier circuit in the architecture, one can optimize the multiplication logic to reduce the complexity of multiplication logic and improve performance of the design. The similar approach has been carried out to design the MCC circuit in this work and reduce the complexity of the design using parallel pipelined floating point adder instead of other adders. This approach significantly reducing the complexity of the architecture and reduce the power supply requirements. In the synthesis results section all the performance parameters are shown.

Keywords - FP Adder, MCC, Parallel Pipelined design, Power Efficient Design.

I. INTRODUCTION

As the integration scale is continues growing, increasingly modern signal processing frameworks are being implemented on a VLSI chip. These signal processing applications requires complex calculation capacity as well as consume extensive measure of energy. While performance and Area stay to be the two noteworthy design tolls, power consumption has turned into a basic worry in the present VLSI framework structure. The requirement for low-power VLSI framework emerges from two fundamental powers. In the first place, with the unfaltering growth of working frequency and processing capacity per chip, vast flows must be conveyed and the heat because of huge power consumption must be evacuated by legitimate cooling strategies. Second, battery life in versatile electronic gadgets is restricted. Low power configuration specifically prompts delayed task time in these convenient devices.

Multipliers are key segments of numerous superior systems, for example, FIR filters, microprocessors, digital signal processors, and so on. A framework's performance is by and large dictated by the performance of the multiplier in light of the fact that the multiplier is by and large the slowest block in the framework. Moreover, it is by and large the most area consuming. Thus, upgrading the speed and area of the multiplier is a noteworthy structure issue. Nonetheless, area and speed are generally clashing constraints with the goal that enhancing speed results for the most part in bigger areas. Accordingly, an entire range of multipliers with various area-speed constraints has been designed with completely parallel. Multipliers toward one side of the range and completely sequential multipliers at the opposite end. Among the digit serial multipliers where single digits comprising of a few bits are worked on. These multipliers have moderate performance in both speed and area. Be in any case, existing digit consecutive multipliers have been plagued by confounded exchanging frameworks or conceivably inconsistencies in plan. Radix 2^n multipliers which work on digits in a parallel mold rather than bits convey the pipelining to the digit level and stay away from most of'the above issues.

Multiplying a variable by an arrangement of known consistent coefficients is a typical task in numerous digital signal processing (DSP) algorithms. Contrasted with other normal operations in DSP algorithms, for example, expansion, subtraction, utilizing delay components, and so forth. Multiplication is for the most part the most costly. There is a trade– off between the measure of logic assets utilized (i.e. the measure of silicon in the integrated circuit) and how quick the calculation should be possible. Contrasted with the vast majority of alternate operations, multiplication requires additional time given a similar measure of logic assets and it requires more logic assets under the constraint that every task must be finished inside a similar measure of time. To concern above issue a efficient multiple constant convolution circuit using modified parallel pipelined floating point addition is proposed and implemented in Virtex -7 device using Xilinx ISE design suit 13.1.

II. PARALLEL PIPELINED FLOATING POINT ADDITION

Representing and manipulating real numbers efficiently by computers is required in many field of science, engineering, finance and more. There exist several

representations for approximating real numbers: fixed-point, logarithmic, continued fractions, floating-point (FP) and many more. Out of these representations, FP is the most popular in modern computer systems.

Each of these representation formats promises a different compromise between speed, accuracy, dynamic range and implementation cost. In modern computer systems, the FP representation seems to provide the best balance between these requirements. A detailed description of FP arithmetic in modern computer systems.

Floating-point seems to be a good compromise between dynamic range, accuracy and implementation complexity when trying to manipulate real numbers. This is one of the main reasons why FP arithmetic is extensively used in scientific algorithms.

Scientific computations usually also involve more operations than just additions and multiplications (supported in hardware by most FPUs). They require divisions, trigonometric functions, exponentials, logarithms, square-roots, accumulations and other. One example of such application is circuit modeling in SPICE. These electronic components basic blocks of the SPICE circuit modeling tool. When simulating these circuits using microprocessors, most of the time is spent evaluating elementary functions (log, exp) which are not supported in silicon. Performance drops even more if these are found deep inside in the inner loops of the code. Nowadays, FPGA implementations of these operators can offer the same throughput as for basic operators, offering as significant speedup compared to the microprocessor counterpart when simulating these models. However, this was not always the case.

Pipelining is a technique that shortens the circuit delay by placing a register in a combinational logic path to break the critical path. Pipelining has the advantage to get high throughput of a circuit because the register to register delay is the delay path that sets the clock rate. For the adder structures discussed in this examination, except for the carry save adder, the critical path is always the carry of the adders, that is, from carry-in to carry-out.

For instance, in Figure 2.1, there is an execution unit that is composed of two logic blocks connected in series. By inserting registers (Figure 2.2 between them, the two series blocks are isolated so that the delay for each logic block is 1/2 of the total delay of the previous execution unit.

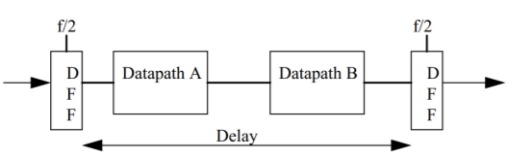

Fig. 2.1 Critical detapath

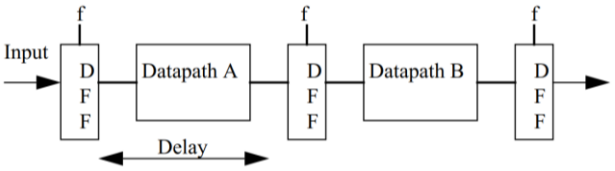

Fig. 2.2 Pipelined datapath.

There is only a small area overhead and a little increase of power consumption of the example above. Both of them are coming from the register inserted in the combinational logic. It is reasonable that an extra register in the circuit will be a little area and power consuming. However, the percent- age of area increase of the register is trivial to the overall area of a large and complicated circuit. Also, the power consumption can be largely reduced because the register works as an isolation to avoid glitch propagation, even there is a little power increase by the register.

III. PROPOSED METHODOLOGY

In this examination an area power and delay efficient circuit has been designed and in Xilinx ISE design suite using Virtex -7 device family. Fig. 3.1 shows the top model of proposed design using multiple constant convolutions using parallel pipelined floating pint adder. In the operation of prefix design the outcome of the operation depends upon the initial value of inputs vectors.

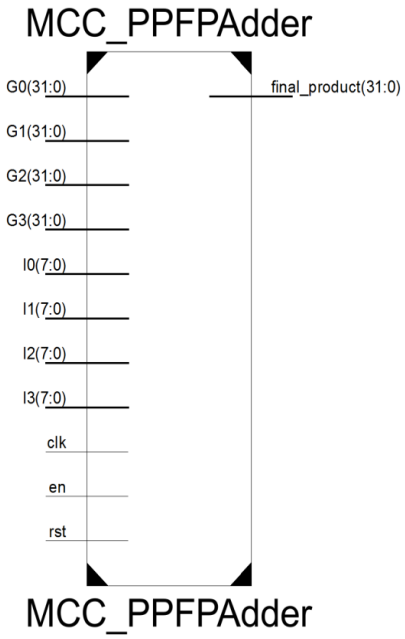

Fig.3.1 Main Top Module of Our Architecture.

The term parallel prefix refers to design of parallel adder using prefix design algorithm. In parallel design of adders the execution of operation of carry propagation involves in parallel by segmentation of whole design. The proposed design provides fast computation because of the processing

achieved in parallel manner. Fig. 3.2 shows the illustration of conventional addition and parallel addition. Fig. 3.2 shows the sub module of multiple constant convolution circuit used to implement proposed design. In the proposed sub module there are two fundamental blocks are used which are Gaussian convolution multiplier and parallel prefix adder block as shown in RTL schematic of proposed design Fig. 3.3.

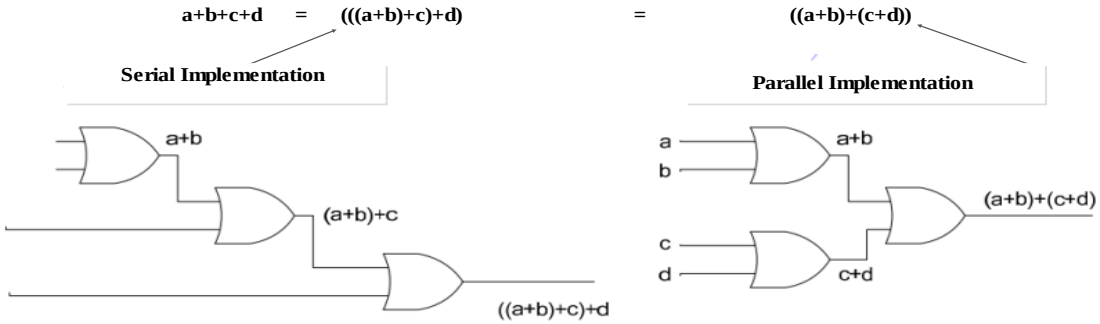

Fig. 3.2 Illustration of serial and parallel implementation of adders.

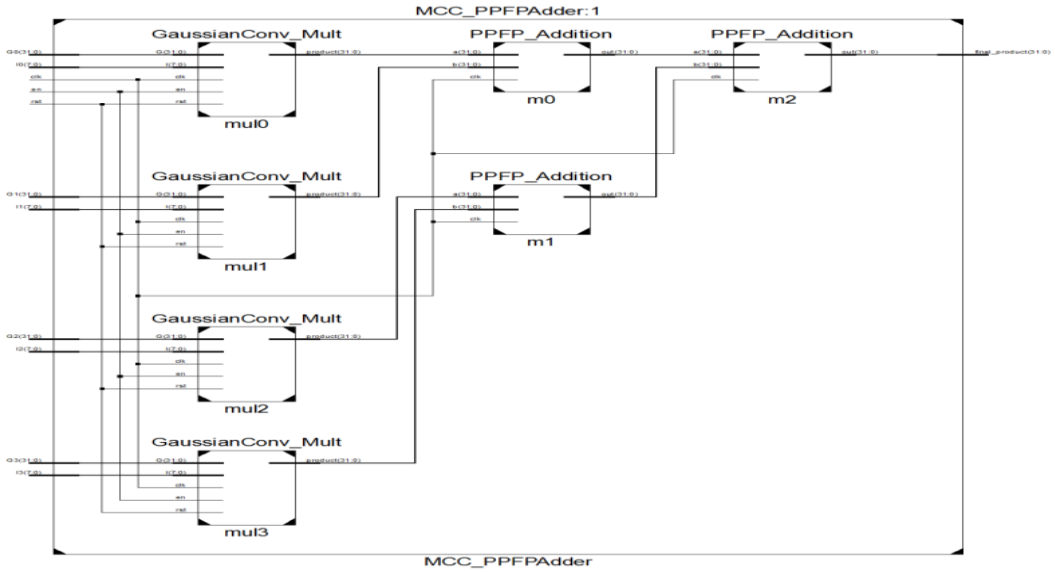

Fig.3.3 RTL Schematic of MCC Multiplier Architecture.

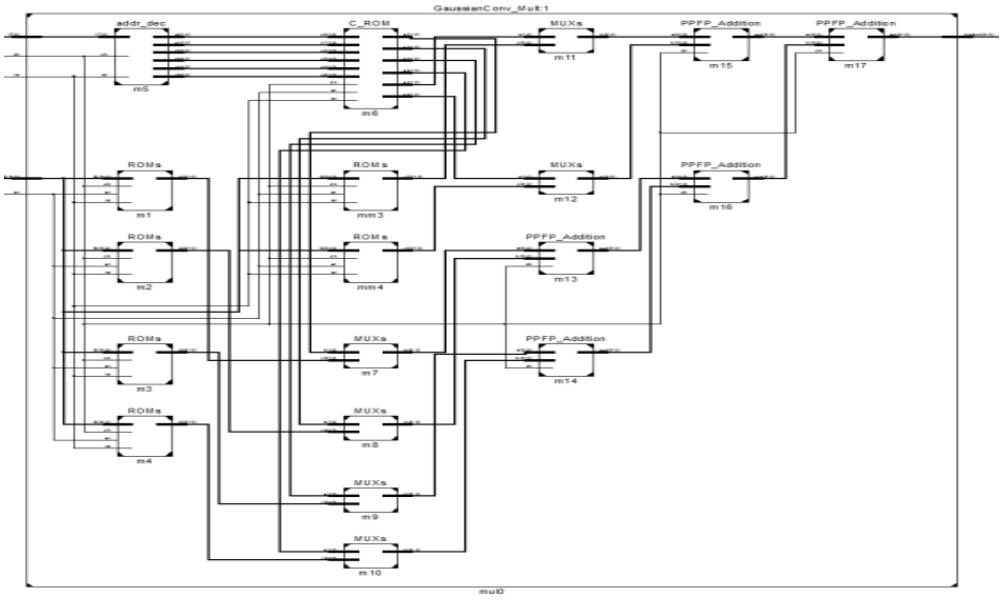

Fig.3.4 Internal Schematic of MCC Multiplier Architecture.

Internal architecture of RTL schematic of MCC sub module is shown in Fig. 3.4. Internal architectures show the design of blocks used in sub module at RTL level. Fig 3.5 shows the schematic of parallel pipelined floating point adder architecture.

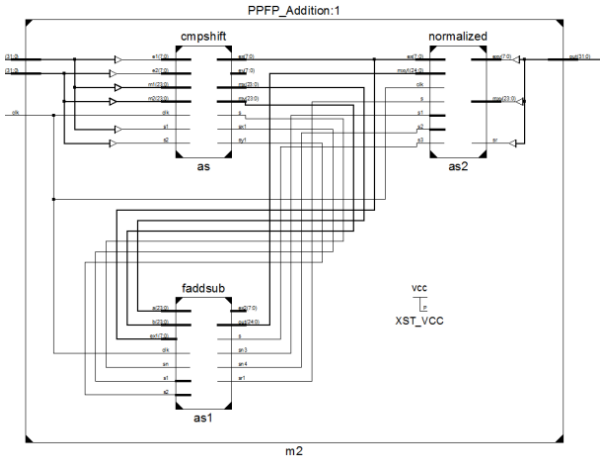

Fig.3.5 Schematic of Parallel Pipelined Floating Point Adder Architecture

IV. SIMULATION RESULTS

Fig. 4.1 shows the Xilinx synthesis screen of proposed design by choosing Virtex 7 device family implemented by Verilog HDL programming language.

The design is synthesized for a large range of pipeline stages to explore latency, area, and delay tradeoffs. This synthesis was performed using Xilinx XST feature. The synthesis of design shows the device utilization summary and timing analysis by timing simulation. The power analysis of proposed design has carried out using Xilinx Xpower analyzer power analysis screen of proposed design has shown in Fig. 4.2.

A comparative analysis of proposed design with existing design has shown in table 1.

Parameters are taken as platform and power in mW both devices are using Virtex 7 device for simulation but the consumption of power during execution of devices are different. Proposed design has better power utilization as compared to previous design.

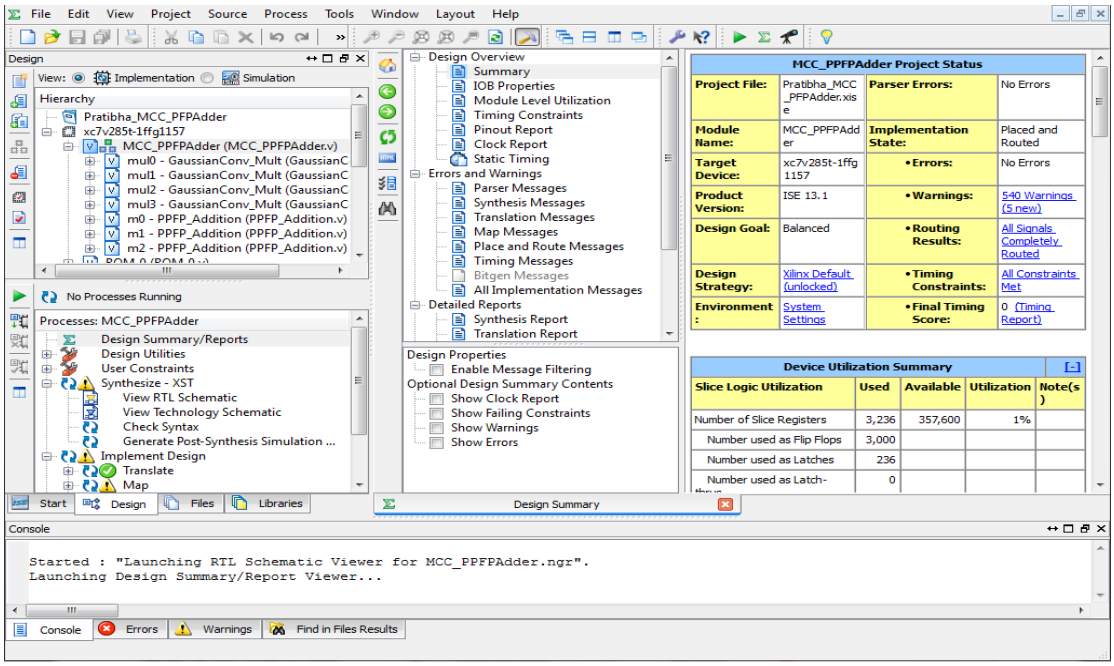

Fig.4.1 Synthesis Screen Shots of the Design.

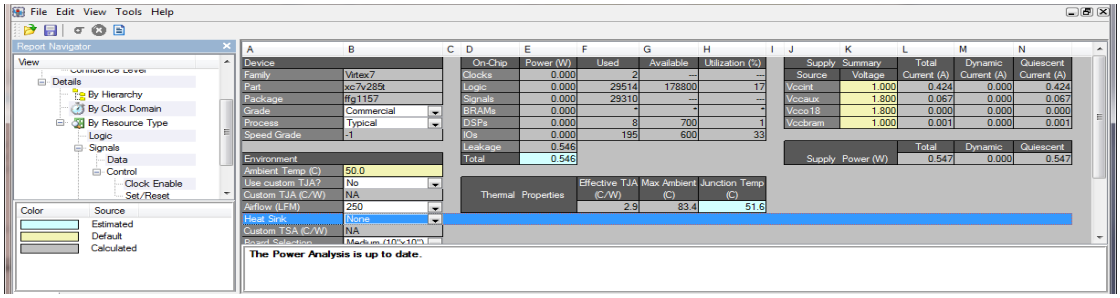

Fig. 4.2 Power Utilization Details of Proposed Design.

Table 2 shows device utilization summary of proposed design. In terms of Slice Logic Utilization, Number of Slice LUTs and Number used as Logic. Tmming summary of proposed design has shown in Table 3 the proposed design have Minimum period: 11.432ns at (Maximum Frequency: 87.471MHz). Minimum input arrival time before clock: 7.127ns and Maximum output required time after clock: 0.726ns.

Table 1: Power Utilization Comparison

Parameters	Previous Architecture	Proposed Architecture
Platform	Virtex 7	Virtex 7
Power (W)	2.317	0.547

Table 2: Device Utilization Summary

Device utilization summary:

```
---------------------------
Selected Device : 7v285tffg1157-1

Slice Logic Utilization:
 Number of Slice LUTs:                 3236  out of  357600     0%
 Number of Slice Registers:           28037  out of  178800    15%
    Number used as Logic:             28037  out of  178800    15%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:  28969
   Number with an unused Flip Flop:   25733  out of  28969     88%
   Number with an unused LUT:           932  out of  28969      3%
   Number of fully used LUT-FF pairs:  2304  out of  28969      7%
   Number of unique control sets:        29

IO Utilization:
 Number of IOs:                        195
 Number of bonded IOBs:                195  out of    600     32%

Specific Feature Utilization:
 Number of Block RAM/FIFO:               2  out of    410      0%
    Number using Block RAM only:         2
 Number of BUFG/BUFGCTRLs:               2  out of     32      6%
 Number of DSP48E1s:                     8  out of    700      1%
```

Table 3: Timing Summary

Timing Summary:

```
---------------
Speed Grade: -1
   Minimum period: 11.432ns (Maximum Frequency: 87.471MHz)
   Minimum input arrival time before clock: 7.127ns
   Maximum output required time after clock: 0.726ns
```

V. CONCLUSION AND FUTURE SCOPE OF WORK

In this examination and implementation work a new architecture of efficient multiple constant convolution circuit using modified parallel pipelined floating point addition. The whole design has carried out in Xilinx ISE design suite 13.1 using Verilog HDL language. The pipeline structure and parallel design lead to have high speed and less area. The performance of proposed design has examined in terms of area speed and power consumption and a comparative analysis of proposed work with existing work has done it is found that proposed design outperforms against previous one considered from base work. The verification of proposed design has done in Virtex-7 based on Simulation. In future proposed design may be implemented in FPGA hardware to test its real time functionality.

REFERENCES

[1] G. D. Licciardo, C. Cappetta, L. Di Benedetto and M. Vigliar, "Weighted Partitioning for Fast Multiplierless Multiple-Constant Convolution Circuit," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 64, no. 1, pp. 66-70, Jan. 2017.

[2] X. Lou and W. Ye, "Low complexity and low power multiplierless FIR filter implementation," 2017 IEEE s12th International Conference on ASIC (ASICON), Guiyang, 2017, pp. 596-599.

[3] G. J. Dolecek and A. Fernandez-Vazquez, "Multiplierless two-stage comb structure with an improved magnitude characteristic," 2016 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), Jeju, 2016, pp. 607-610.

[4] K. H. Dangra and G. S. Gawande, "Efficient design and implementation of multiplierless FIR filter," 2016 International Conference on Computing Communication Control and automation (ICCUBEA), Pune, 2016, pp. 1-5.

[5] G. R. Teja, R. S. Sruthi, K. S. Tomar, S. Sivanantham and K. Sivasankaran, "Verilog implementation of fully pipelined and multiplierless 2D DCT/IDCT JPEG architecture," 2015 Online International Conference on Green Engineering and Technologies (IC-GET), Coimbatore, 2015, pp. 1-5.

[6] W. B. Ye, X. Lou and Y. J. Yu, "Design of high-speed multiplierless linear-phase FIR filters," 2015 IEEE International Symposium on Circuits and Systems (ISCAS), Lisbon, 2015, pp. 2964-2967.

[7] D. N. Milić and V. D. Pavlović, "A New Class of Low Complexity Low-Pass Multiplierless Linear-Phase Special CIC FIR Filters," in IEEE Signal Processing Letters, vol. 21, no. 12, pp. 1511-1515, Dec. 2014.

[8] S. Y. Park and P. K. Meher, "Low-Power, High-Throughput, and Low-Area Adaptive FIR Filter Based on Distributed Arithmetic," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 60, no. 6, pp. 346-350, June 2013.

[9] S. L. Chen, “VLSI implementation of an adaptive edge-enhanced image scalar for real-time multimedia applications,” IEEE Trans. Circuits Syst. Video Technol., vol. 23, no. 9, pp. 1510–1522, Sep. 2013.

[10] F. C. Huang, S. Y. Huang, J. W. Ker, and Y. C. Chen, “High perfor- mance SIFT hardware accelerator for real-time image feature extraction,” IEEE Trans. Circuit Syst. Video Technol., vol. 22, no. 3, pp. 340–351, Mar. 2012.

[11] G. D. Licciardo, A. D’Arienzo, and A. Rubino, “Stream processor for real- time inverse tone mapping of full-HD images,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 11, pp. 2531–2539, Nov. 2015.

[12] M. Vigliar and G. D. Licciardo, “Hardware coprocessor for stripe-based interest point detection,” US Patent 20 130 301 930, Nov. 14, 2013.

[13] K. K. Parhi, VLSI Signal Processing Systems: Design and Implementation. New York, NY, USA: Wiley, 2007.