



Article Title: **AxPPA: Approximate Parallel Prefix Subtractor**

AxPPA: Approximate Parallel Prefix Subtractor

Jithu M, Abisheka. R.A

Department of Electronics and Communication Engineering,
Marthandam College of Engineering and Technology, Kuttakuzhi, India
Assistant professor, Department of Electronics and Communication Engineering,
Marthandam College of Engineering and Technology, Kuttakuzhi, India

ABSTRACT

Parallel prefix Subtractor is the most flexible and widely used for binary addition/subtraction. Parallel Prefix Subtractor is best suited for VLSI implementation. No any special parallel prefix Subtractor structures have been proposed over the past years intended to optimize area. This paper presents a new approach to new design the basic operators used in parallel prefix architectures which subtract the unit by using modified technique of 2's complement. Verification will also be done using LFSR technique so we don't need to apply any manually input to perform the subtraction process. We can analysis and create the difference in terms of area between parallel prefix Subtractor and BIST architecture of parallel prefix Subtractor. The number of multiplexers contained in each Slice of an FPGA is considered here for the redesign of the basic operators used in parallel prefix tree. The experimental results indicate that the new approach of basic operators make some of the parallel Prefix Subtractor architectures faster and area efficient.

Keywords: Parallel Prefix Adder, 2s complement, Optimize Area, BIST architecture, LFSR Approach

1 Introduction

Parallel prefix is the discriminating component in most computerized circuit plans including advanced sign processors (DSP) and microchip information way units. Thusly, broad exploration keeps on being centered around enhancing the force delay execution of the viper. In VLSI usage, parallel-prefix adders are known to have the best execution. Reconfigurable rationale, for example, Field Programmable Gate Arrays (FPGA) has been picking up in notoriety as of late in light of the fact that it offers enhanced execution regarding speed and control over DSP-based and chip based answers for some handy outlines including versatile DSP and information transfers applications and a noteworthy decrease being developed time and cost over Application Specific Integrated Circuit (ASIC) plans. The force point of interest is particularly imperative with the developing ubiquity of portable and versatile hardware, which make far reaching utilization of DSP

Capacities. Then again, in view of the structure of the configurable rationale and directing assets in Fpgas, parallel-prefix adders will have an alternate execution than VLSI executions. Specifically, most advanced Fpgas utilize a quick convey chain which advances the convey



Article Title: AxPPA: Approximate Parallel Prefix Subtractor

way for the basic Ripple Carry Adder (RCA). In this paper, the viable issues included in planning and actualizing tree-built adders in light of Fpgas are. A proficient testing technique for assessing the execution of these adders is examined. A few tree-based viper structures are actualized and described on a FPGA and contrasted and the Ripple Carry Adder (RCA) and the Carry Skip Adder (CSA). At last, a few conclusions and proposals for enhancing FPGA plans to empower better tree-based viper execution are given

2 Related Work

Xing and Yu noted that postpone models and expense investigation for viper outlines produced for VLSI engineering don't delineate to FPGA plans. They thought about the outline of the swell convey viper with the convey lookahead, convey skip, and convey select adders on the Xilinx 4000 arrangement Fpgas. Just an enhanced type of the convey skip viper performed better than the swell convey snake when the viper operands were over 56 bits. An investigation of adders actualized on the Xilinx Virtex II yielded comparable results [9]. In [10], the creators considered a few parallel prefix adders actualized on a Xilinx Virtex 5fpga. It is observed that the basic RCA viper is better than the parallel prefix outlines on the grounds that the RCA can exploit the quick convey chain on the FPGA. Kogge-Stone the Kogge-Stone tree [22] Figures 1- 5 accomplishes both $\log_2 n$ stages and fan-out of 2 at each one stage. This takes on at the expense of long wires that must be directed between stages. The tree additionally contains more PG cells; while this may not affect the range if the viper design is on a consistent lattice, it will expand power utilization. Regardless of these expense, Kogge-Stone viper is for the most part used for wide adders because it shows the lowest delay among other structures.

2.1 Subtractor 2's Complement

Consider the multiplier data A to be used with the negative partial product factors. To calculate the 2's complement first is to inverse all the bits of the data A denoting them as Abar. Now perform "Exclusive OR" (XOR) operation on Abar(0) with 1'b1, Abar(1) xor Abar(0), Abar(2) xor Abar(1) and so on till a 1'b0 is found while traversing the data bits A(i). Once 1'b0 is arrived keep the remaining bits as it is without any change.

Let's us consider an example where $A=10101000$, then 2's complement of A be denoted as A2_c_bar, then

Step 1: Abar=01010111.

Step 2:

$$A2_c_bar(0) = 1 \text{ xor } 1 = 0$$

$$A2_c_bar(1) = 1 \text{ xor } 1 = 0$$

$$A2_c_bar(2) = 1 \text{ xor } 1 = 0$$

$$A2_c_bar(3) = 1 \text{ xor } 0 = 1$$

$$A2_c_bar(4) = A'4 = 1$$



Article Title: AxPPA: Approximate Parallel Prefix Subtractor

$$A2_c_bar(5) = A'5 = 0$$

$$A2_c_bar(6) = A'6 = 1$$

$$A2_c_bar(7) = A'7 = 0$$

2.2 Parallel Prefix Subtractor

Given figure 4 represents the subtraction part using parallel prefix Subtractor using modified approach of 2's compliment. Output analysis of this approach will be explained in detail in this paper in result section. The force point of interest is particularly imperative with the developing ubiquity of portable and versatile hardware, which make far reaching utilization of DSP capacities. Then again, in view of the structure of the configurable rationale and directing assets in Fpgas, parallel-prefix adders will have an alternate execution than VLSI executions Specifically, most advanced Fpgas utilize a quick convey chain which advances the convey way for the basic Ripple Carry Adder (RCA).

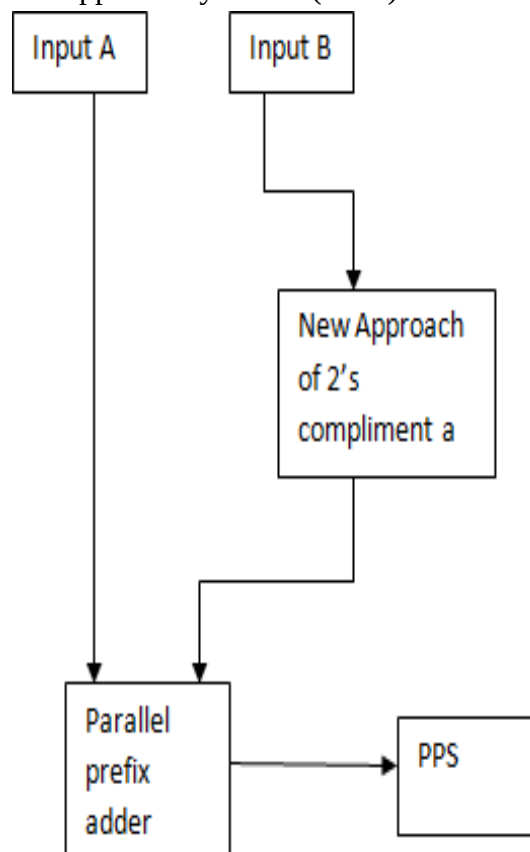


Figure 1: *Parallel Prefix Subtractor*

2.3 Bist Approach

Built in self-test architecture, which analysis the on chip verification of Circuit under Test (CUT). We do not need to apply any input for any input drivers. Figure 5 & Figure 6



Article Title: AxPPA: Approximate Parallel Prefix Subtractor

represent the logical architecture bist capability using LFSR technique for any circuit in vlsi design.

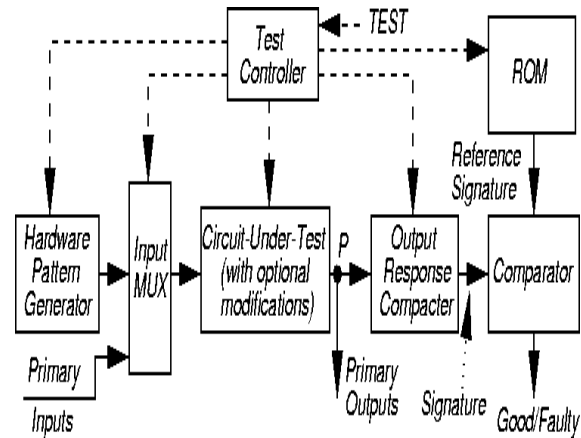


Figure 2: BIST Approach

2.4 BIST approach of Subtractor

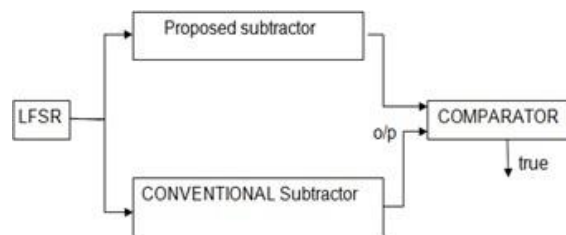


Figure 3: BIST with PPS

4 Result and Analysis

Proposed parallel prefix Subtractor

Input a= 0001000100010001(4'h1111)

Input b= 0001000100010001(4'h1111)

Output: = 0000000000000000(4'h0000)

	Msgs	
/CONVENTIONAL_PARALLE_SUBT/A	16'h1111	16'h1111
/CONVENTIONAL_PARALLE_SUBT/B	16'h1111	16'h1111
/CONVENTIONAL_PARALLE_SUBT/SUM	16'h0000	16'h0000
/CONVENTIONAL_PARALLE_SUBT/COU	1'h1	
/CONVENTIONAL_PARALLE_SUBT/comp_B	16'heeff	16'heeff
/CONVENTIONAL_PARALLE_SUBT/G	16'h0001	16'h0001
/CONVENTIONAL_PARALLE_SUBT/P	16'hffff	16'hffff
/CONVENTIONAL_PARALLE_SUBT/H	16'hffe	16'hffe
/CONVENTIONAL_PARALLE_SUBT/C	16'hffff	16'hffff
/CONVENTIONAL_PARALLE_SUBT/S	16'h0000	16'h0000
/CONVENTIONAL_PARALLE_SUBT/K	16'hffff	16'hffff
/CONVENTIONAL_PARALLE_SUBT/I	32'h00000010	32'h00000010

Figure 4: Waveform of Proposed Design



Article Title: AxPPA: Approximate Parallel Prefix Subtractor

4.1 Area Analysis

```

Number of errors:      0
Number of warnings:    0
Logic Utilization:
  Number of 4 input LUTs:      54 out of  3,840   1%
Logic Distribution:
  Number of occupied Slices:      33 out of  1,920   1%
    Number of Slices containing only related logic:      33 out of   33 100%
    Number of Slices containing unrelated logic:          0 out of   33   0%
    *See NOTES below for an explanation of the effects of unrelated logic
Total Number of 4 input LUTs:      54 out of  3,840   1%
Number of bonded IOBs:      48 out of   141  34%

Total equivalent gate count for design: 324
Additional JTAG gate count for IOBs: 2,304
  
```

Figure 5: Bottom level design

4.2 Timing analysis

```

Minimum period: No path found
Minimum input arrival time before clock: No path found
Maximum output required time after clock: No path found
Maximum combinational path delay: 24.091ns

Timing Detail:
-----
All values displayed in nanoseconds (ns)

-----
Timing constraint: Default path analysis
Delay:      24.091ns (Levels of Logic = 16)
Source:     B<0> (PAD)
Destination: SUM<15> (PAD)

Data Path: B<0> to SUM<15>

Cell:in->out      fanout  Gate  Net
                  Delay    Delay  Logical Name (Net Name)
-----
IBUF:I->O          5    1.492  0.658  B_0_IBUF (B_0_IBUF)
LUT2:T0->Y0        4    0.720  0.629  W0v3q3q1 (N3q41)
  
```

Figure 6: Waveform of Proposed Design

4.3 BIST Result

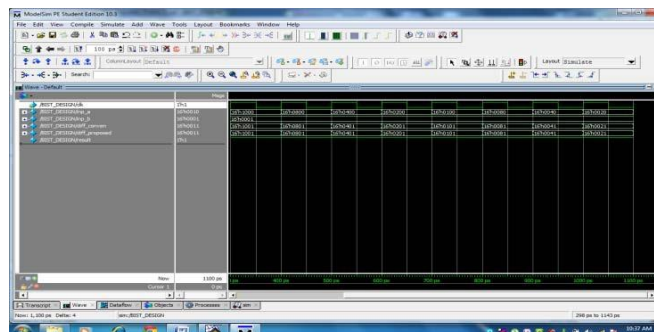


Figure 7: Waveform of bistarchitecture



Article Title: AxPPA: Approximate Parallel Prefix Subtractor

Waveform of bistarchitecture: result is done and set on 1 while expected output and proposed output is same

5 Conclusion

In this paper we approach Parallel prefix Subtractor using prefix algorithm. Proposed design based on modified 2's complement method whereas area reduced by approx. 12% and delay reduced by 13%. BIST architecture also introduced for on chip verification using LFSR technique. In future this work can be extend to multiplication part for modulo arithmetic operation.

References

1. Power-conscious syntheses of parallel prefix adders under bitwise timing constraints, Proc. the Workshop on Synthesis and System Integration of Mixed Information technologies (SASIMI), Sapporo, Japan, October 2007, pp. 7–14
2. Stefania Perri; Pasquale Corsonello; Giuseppe Cocorullo, Year: 2014, "Area-Delay Efficient Binary Adders in QCA" IEEE Trans. Very Large Scale Integration (VLSI) Syst., Vol: 22, no. 5, pp. 1174–1179, May 2014.
3. Maheshwaran. K, Year: 2016. CMOS Design of Low Power High Speed Hybrid Full Adder. International Journal of MC Square Scientific Research, Vol: 8, no: 1, pp.16-22.
4. T.Matsunaga; Y. Matsunaga, Year: 2007, Timing-constrained area minimization algorithm for parallel prefix adders, IEICE Transactions on Fundamentals, Vol: E90-A, no: 12, pp. 2770–2777, December 2007
5. L.Lu; W. Liu; M. O'Neill; E. E. Swartzlander, Jr., QCA systolic array design," IEEE Trans. Compute., Vol: 62, no. 3, pp. 548–560, Mar. 2013.
6. V. Pudi; K. Sridharan, Year: 2012, "Low complexity design of ripple carry and Brent–Kung adder in QCA," IEEE Trans. Nanotechnol., Vol: 11, no: 1, pp. 105–119, Jan. 2012.