IJIEMR Transactions, online available on 19th Dec 2018. Link :

http://www.ijiemr.org/main/index.php?vol=Volume-07&issue=ISSUE-13

Title: DESIGN AND IMPLEMENTATION OF FUSION MAC USING WALLACE AND BOOTH MODELING

Volume 07, Issue 13, Pages: 565–573.

Paper Authors

MS.CH.BHARGAVI, MR.P.ABHISHEK

Ashoka Institute of Engineering and Technology

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

DESIGN AND IMPLEMENTATION OF FUSION MAC USING WALLACE AND BOOTH MODELING

[1]MS.CH.BHARGAVI,[2]MR. P.ABHISHEK
[1]M.tech Scholar, VLSI System Design, Department of ECE, Ashoka Institute of Engineering and Technology
[2]Assistant professor, M.tech(VLSI), Department of ECE,Ashoka Institute of Engineering and Technology
[1]cheralabhargavi@gmail.com,[2]p.abhiemails@gmail.com

Abstract The augmentation activity is available in numerous parts of the advanced framework or computerized PC, most remarkably in flag preparing, designs and logical calculation. With propels in the innovation, different procedures have been proposed to outline multipliers, which offer fast, low power utilization and lesser zone. The primary adage of this task is to actualize an elite Multiplier-and-aggregator (MAC) by utilizing Wallace multiplier and to check the re-enactment results on ISE programming. After the execution of MAC unit Booth augmentation came into presences for simple handling of the duplications. Be that as it may, MAC unit with corner calculation underpins up as far as possible (8bit and 16bit). As the scope of bits increments again the heap will be expanded and the circuit ends up complex. To beat the disadvantages in existing framework we accompanied an alternate methodology i.e., MAC unit by utilizing Wallace multiplier. Macintosh unit with Wallace multiplier works autonomously, so there will be less load on Processor unit. The aggregate outline is coded with Verilog-HDL and the blend is finished utilizing rhythm RTL complier utilizing common libraries of TSMC O.18um innovation. The aggregate MAC unit works at 217 MHz the aggregate power scattering is 177m.732W.

1. Introduction

Presently days, the interest for the fast versatile remote correspondences is quickly developing. The Multiplier-Accumulator (MAC) activity is the key task in DSP applications as well as in interactive media data preparing and different applications. As specified above, MAC unit comprise of multiplier, viper and enlist/gatherer. In this task, we utilized 16 bit Wallace multiplier. The MAC inputs are gotten from the memory area and given to the multiplier square. This will be valuable in 16 bit advanced flag processor. The info which is being sustained from the memory area is 16 bit.The advances in innovation, numerous specialists have attempted and are endeavouring to outline multipliers which offer both of following – fast, low power utilization, normality of design and thus less territory or even blend of them in one multiplier. Along these lines making them reasonable for different fast, low power, and conservative VLSI usage. The basic augmentation technique is include and move calculation. Augmentation is a numerical task that at its easiest is a curtailed procedure of adding a whole number to itself, a predetermined number of times. A number (multiplicand) is added to itself various occasions as determined by another number (multiplier) to shape an outcome (item). Augmentation equipment frequently devours much time and territory contrasted with other math tasks. Computerized flag processors utilize a multiplier unit as an essential building square and the calculations they run are frequently duplicate escalated. Augmentation based activities are as of now actualized in numerous Digital Signal Processing (DSP) applications, for example, convolution, Fast Fourier Transform (FFT), sifting and in chip in its number-crunching and rationale unit. In this section, we examine diverse structures for duplication and the techniques that enhance speed, control as well as region. Likewise, it is imperative to consider these strategies with regards to VLSI outline. It is advantageous to discover structures that are particular and simple to format. A MAC unit comprises of a multiplier and an aggregator containing the entirety of the past Successive items. The MAC inputs are acquired from the memory area and given to the multiplier square. Multipliers are key segments of numerous superior frameworks, for example, FIR channels, microchips, advanced flag processors, and so on. A framework's execution is for the most part dictated by the execution of the multiplier in light of the fact that the multiplier is by and large the slowest component in the framework. In any case, zone and speed are typically clashing limitations with the goal that enhancing speed results generally in bigger zones. Thus, entire ranges of multipliers with various zone speed imperatives have been composed with completely parallel. Consequently with the reasonable decision of the kind of the multiplier the execution of the MAC unit can be improved.

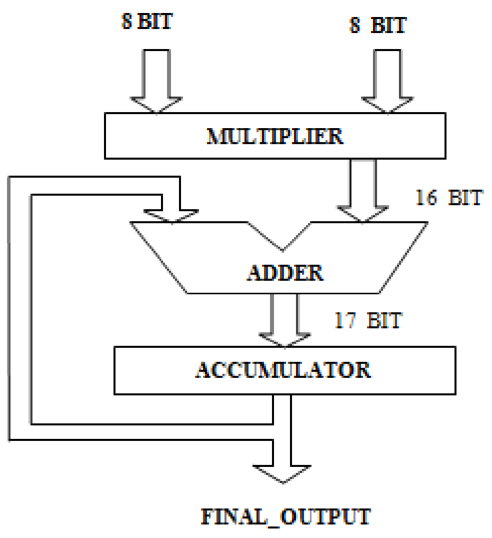


Fig.1 Block diagram of MAC

The multiplier yield is given as the contribution to convey spare viper which performs expansion. The capacity of the MAC unit is given by the accompanying condition.

$$F = I\ PiQi$$

The yield of convey spare viper and one piece is for the convey .Then, the yield is given to the gatherer enroll. The aggregator enroll utilized in this plan is Parallel in Parallel out (PIPO). Since the bits are colossal and furthermore convey spare snake creates all the yield esteems in parallel, PIPO enroll is utilized where the information bits are taken in parallel and yield is taken in parallel. The yield of the gatherer enroll is taken out or nourished back as one of the contribution to the convey spare viper. As appeared in fig 1. the fundamental design of MAC unit.

2. Research Work

In about all computerized IC plans today, the expansion task is a standout amongst the most fundamental and regular activities. Guidance sets for DSP's and universally useful processors incorporate no less than one sort of expansion. Different directions, for example, subtraction and augmentation utilize expansion in their tasks, and their fundamental equipment is comparable if not indistinguishable to expansion equipment. Regularly, a snake or different adders will be in the basic way of the outline, consequently the execution of a plan will be frequently be constrained by the execution of its adders. When taking a gander at different properties of a chip, for example, region or power, the originator will find that the equipment for expansion will be a huge supporter of these territories.In this section we start with the essential building squares utilized for expansion, at that point experience distinctive calculations and name their focal points and drawbacks.

Ripple Carry Adder

The Ripple Carry Adder (RCA) is one of the least complex adders to execute. This snake takes in two N-bit inputs (where N is a positive whole number) and delivers (N + 1) yield bits (a N-bit entirety and a 1-bit carryout).In the most pessimistic scenario (when all the carryout's are 1), this convey bit needs to swell over the structure from the minimum noteworthy position to the most huge position. Thus, the ideal opportunity for this execution of the snake is communicated in beneath eq, where tRCAcarry is the deferral for the carryout of a FA and tRCAsumis the postponement for the total of a FA.

$$\text{Propagation Delay}(t_{RCAprop})=(N-1).\ t_{RCAcarry}+t_{RCAsum}$$

From Equation we can see that the postponement is relative to the length of the snake. A case of a most pessimistic scenario spread defer input design for a 4 bit swell convey viper is the place the info operands change from 1111 and 0000 to 1111 and 0001, bringing about a whole changing from 01111 to 10000. From a VLSI plan point of view, this is the least demanding viper to actualize. One simply needs to outline and format one FA cell, and after that exhibit N of these cells to make a N-bit RCA. The execution of the one FA cell will to a great extent decide the speed of the entire RCA.

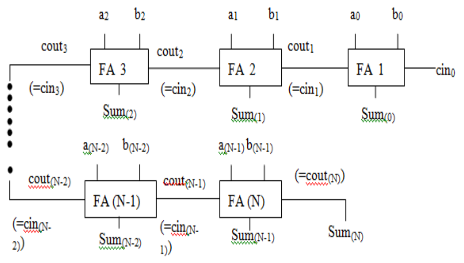


Fig.2 Schematic for an N-bit Ripple Carry Adder

Ripple Carry Drawbacks

- Not extremely productive when huge piece numbers are utilized.
- Delay increments straightly with the bit length.

Carry Skip Adder

From examination of the RCA, the constraining variable for speed in that snake is the engendering of the cout bit. The Carry Skip Adder (CSKA, otherwise called the Carry Bypass Adder) addresses this issue by taking a gander at gatherings of bits and decides if this gathering has a carryout or not This is refined by making a gathering engender flag (pCSKAgroup) to decide if the gathering (carryin CSKA gathering) will spread over the gathering to the carryout (carryoutCSKAgroup). To investigate the task of the entire CSKA, take a N-bit snake and separation it into N/M gatherings, where M is the quantity of bits per gathering. Each gathering contains a 2-to-1 multiplexer, rationale to figure M entirety bits, and rationale to compute pCSKA gathering. The select line for the mux is essentially the pCSKA assemble flag, and it picks between carryin CSKA gathering or cout4.which demonstrates the equipment for a gathering of 4 bits (M=4) in the CSKA.For the situation where pCSKA amass is 0, something like one of the spread signs is 0. This suggests either an erase or potentially produce happened in the gathering. An erase flag basically implies that the carryout for the gathering is 0 paying little respect to the convey in, and a create flag implies that the carryout is 1 paying little heed to the convey in.No equipment is expected to actualize these two signs on the grounds that the gathering carryout flag will reflect one of the three cases (a d, g or gathering p happened). The extra equipment to understand the gathering carryout refined with a 4-information AND entryway and a 2-to-1 multiplexer (mux). All in all, a M-info AND door and a 2-to-1 mux are required for a gathering of bits, including the rationale to ascertain the total bits.

pCSKAgroup = p0. p1. p2.p3

carryoutCSKAgroup= carryinCSKAgroup . pCSKAgroup

Each mux at that point knows which flag to go as the carryout of the gathering. There are two cases to consider after the mux select line has been decided.In the main case, convey in CSK A gathering will engender to the carryout. This implies pCSKAgroup=1 and the carryout is reliant on the convey in. In the second case, the carryout flag of the most huge snake will turn into the gathering carryout. This implies pCSKAgroup=0 and the carryout is autonomous of the carryin. In the event that we disconnect a specific gathering, the second case (flag cout4) dependably takes longer in light of the fact that the carryout flag must be ascertained through rationale, while the principal case (carryin CSKA gathering) requires just a wire to engender the flag.

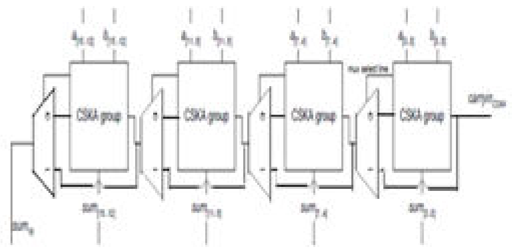


Fig.3 16-bit Carry Skip Adder N=16, M=4

Carry Select Adder

Including two numbers by utilizing excess can speed expansion much further. That is, for any number of entirety bits we can perform two increments, one expecting the carryin is 1 and one accepting the carryin is 0, and afterward pick between the two outcomes once the real carryin is known. This plan, proposed by Sklansky in 1960, is called contingent aggregate expansion. An execution of this plan was first acknowledged by Bedrij and is known as the Carry Select Adder (CSLA).

3. Implementation

3.1 Hybrid Ripple Carry Look Ahead Adder (Hrcla) Architecture

Hybridization has been accomplished by undulating the third convey of a 4-bit CLA. This strategy brought about decrease of land yet at the expense of expanded basic way delay.Furt her, the idea has been stretched out to have two and three full adders undulated in a si milar way.

3.2 Proposed hybrid architecture I

The principle issue in CSA utilizing CLA is, the quantity of bits registered per organize is restricted to four bits. In these new cross breed models the quantity of bits figured per MUX stage can be expanded. CLA is utilized in introductory stages, HRCLA is utilized in 4 phases. In Later stages this technique lessens the including segments, containing just swell adders. By utilizing swell adders, the bits figured per MUX stage can be expanded straightly, prompting expanded speed.

3.3 Proposed hybrid architecture II

The second design depends on the hybridization of CSA by utilizing both RCA and CLA. In this design the 4-bit CLA has been held in each stage. RCA is fell aggregately by one piece to the CLA in each stage (crossover structure). The quantity of phases of mux can be lessened further, when contrasted with the engineering I, thus the basic way delay has been additionally decreased.

3.4 System Architecture

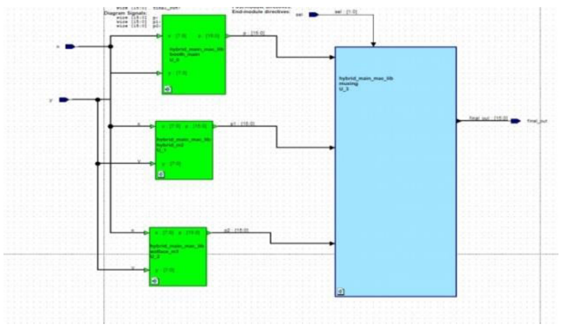


Fig.4 System Architecture

3.5 CLA Principle Concept Analysis

A ripple-carry adder works similarly as pencil-and-paper strategies for expansion. Beginning at the furthest right (minimum huge) digit position, the two comparing digits are included and an outcome got.It is likewise conceivable that there might be a do of this digit position (for instance, in pencil-and-paper strategies, "9 + 5 = 4, convey 1"). In like manner, all digit positions other than the furthest right one have to consider the likelihood of including an additional 1 from a convey that has rolled in from the following position to one side. This implies no digit position can have a completely last an incentive until the point that it has been set up regardless of whether a convey is rolling in from the right. Additionally, if the aggregate without a convey is 9 (in pencil-and-paper strategies) or 1 (in twofold number juggling), it isn't even conceivable to tell regardless of whether a given digit position will pass on a convey to the situation to its left side.

3.6 Carry-Save Adder

Adder is a kind of advanced adder, utilized in PC small scale engineering to figure the entirety of at least three n-bit a convey spare numbers in paired. It varies from other computerized adders in that it yields two quantities of indistinguishable measurements from the information sources, one which is a grouping of halfway aggregate bits and another which is a succession of convey bits.

3.7 Booth Algorithm

Booth augmentation calculation or Booth calculation was named after the innovator Andrew Donald Booth. It very well may be characterized as a calculation or technique for increasing paired numbers in two's supplement documentation. It is a basic strategy to duplicate twofold numbers in which increase is performed with rehashed expansion tasks by following the stall calculation. Again this stall calculation for duplication activity is additionally changed and consequently, named as altered corner calculation.

- **Booth Recoding Table for Radix-4**

The means given underneath speak to the radix-4 stall calculation:

- Extend the sign piece 1 position if important to guarantee that n is even.
- Append a 0 to one side of the slightest critical piece of the stall multiplier.
- According to the estimation of every vector, every fractional item will be 0, +y, - y, +2y or - 2y.

Table.1 Booth recoding

- **Booth's Encoder**

Adjusted booth multiplier's (Z) digits can be characterized with the accompanying condition: Zj = q2j + q2j-1 - 2q2j+1 with q-1 = 0
The figure demonstrates the adjusted stall calculation encoder circuit. Presently, the result of any digit of Z with multiplicand Y might be - 2y, - y, 0, y, 2y.

In any case, by performing left move activity at fractional items age arrange, 2y might be produced. By taking 1's supplement to this 2y, nullification is done, and after that one is included suitable 4-2 blower. One stall encoder appeared in the figure produces three yield motions by taking three back to back piece inputs in order to speak to each of the five potential outcomes - 2X, - X, 0, X, 2X.

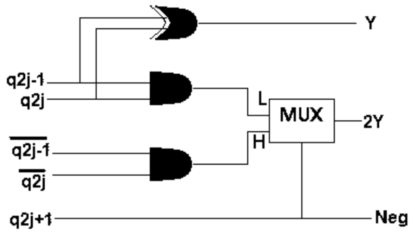


Fig.5 Booth Encoding

3.8 Proposed Wallace Tree Multiplier

In the proposed design, contains an AND exhibit for figuring the incomplete Products. Incomplete item decrease is proficient by the utilization of 4:2, 5:2 blower structures and the last phase of expansion is performed by a convey spare adder.

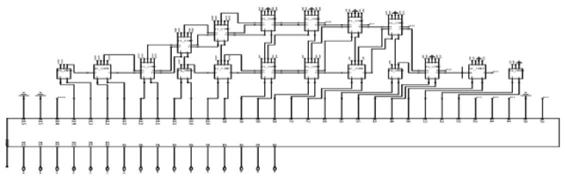


Fig.6 Schematic of novel Wallace tree multiplier

Multipliers Bit Block			Recode 1-bit pair		2 bit Booth	
i+j	i	i-j	i+j	i	Multiplier Value	Partial Product
0	0	0	0	0	0	Mx0
0	0	1	0	1	1	Mx1
0	1	0	1	-1	1	Mx1
0	1	0	1	0	2	Mx2
1	0	0	-1	0	-2	Mx-2
1	0	1	-1	1	-1	Mx-1
1	1	0	0	-1	-1	Mx-1
1	1	0	0	0	0	Mx0

Compressor

The 4-2 blower has 4 inputs X1, X2, X3 and X4 and 2 yields Sum and Carry alongside a Carry-in (Cin) and a Carry-out (Cout) as appeared in Fig 4.4. The information Cin is the yield from the past lower critical blower. The Cout is the yield to the blower in the following critical stage.The 4-2 blower is represented by the fundamental condition

$$x1+x2+x3+x4+Cin = Sum + 2*(Carry + Cout)$$

The 5-2 Compressor square has 5 inputsX1,X2,X3,X4,X5 and 2 yields, Sum and Carry, alongside 2 input convey bits (Cin1, Cin2) and 2 yield convey bits

(Cout1,Cout2) as appeared in Fig. The information convey bits are the yields from the past lesser huge blower square and the yield convey are passed on to the following higher critical compressor block.The essential condition that administers the capacity of the 5-2 blower square is given underneath.

X1+X2+X3+X4+X5+Cin1+Cin2 =Sum+2*(Carry + Cout1 + Cout2)

4. Results

Implementation analysis for proposed design

In this we need to clarify about the proposed configuration result they are for the most part required three investigation.

- Area examination
- Power examination
- Delay/Time examination

Primitive and Black Box Usage:

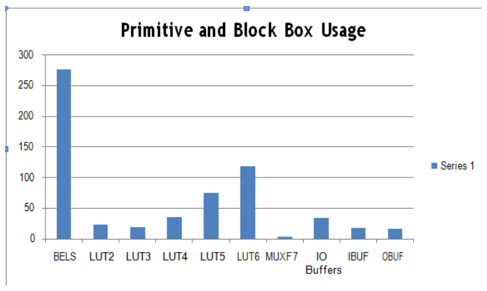


Fig.7 Usage of primitive and block box

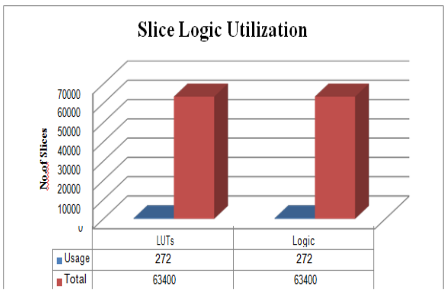
In the above chart is shown in usage of primitive and block boxes. In that how many LUTS, BELS, MUX, IO Buffers, IBUF, OBUF are used to design the above chart.

Slice Logic Utilization:

Number of Slice LUTs: 272 out of 63400

Number used as Logic: 272 out of 63400

In the below chart is shown in utilization of slice logic. This can be described the how many LUTS and logics are used in this chart.



	LUTs	Logic
🟦 Usage	272	272
🟥 Total	63400	63400

Fig.8 Logic Utilization

Timing Details:

All values displayed in nanoseconds (ns)

Total number of paths / destination ports: 93720 / 16

RTL Schematic diagram

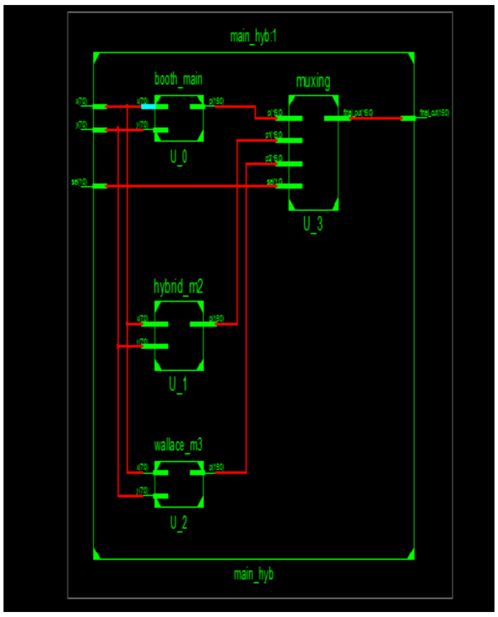


Fig.9 RTL Schematic diagram

In above fig demonstrates the RTL of proposed plan. In that basically four squares are required i.e. Altered stall , crossover viper , Wallace multiplier and mux. The sources of info (operands) are given to the altered corner, half and half viper and Wallace snake/multiplier to play out the activity. Subsequent stage the mux will be chosen to the in view of the determination line to created at the yield results

Simulation Results

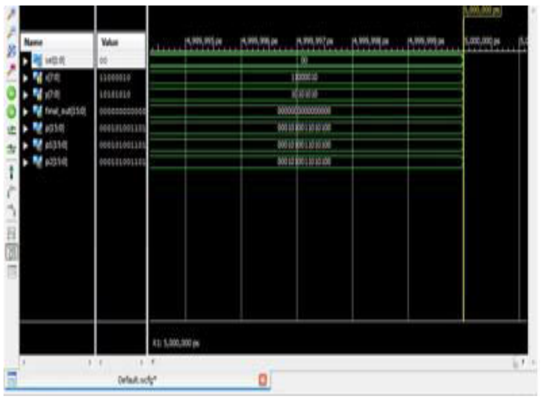


Fig.10 Initialization of input (00)

In this recreation result, the sources of info are given and the mux will be chosen to choice line is "00".in this condition there is no yield why on the grounds that the mux will be doesn't perform in "00"condition, that is the reason the yield is zero.

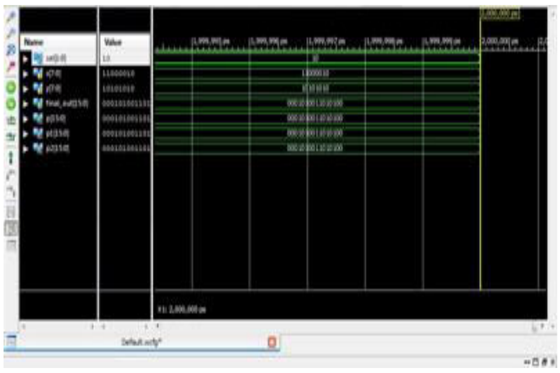


Fig.11 Initialization of input (10)

Considering above simulation results the selection line of mux is “10”.in this condition the hybrid adder output will be selected. then the mux output is hybrid adder output.

5.CONCLUSION

Presently, from the idea of the current outline adders plan we have proposed CSA,CLA based snake outline modules to execute combination MAC squares are:

i. Modified Booth
ii. Hybrid adder
iii. Wallace multiplier/adder

Directly we broke down and assessed the proposed plan modules as specified above utilizing verilog. The estimation of every viper is arranged and modules as charts appeared in results and exchanges. Every module are integrated and recreated utilizing Xilinx 14.2.the relative examination of proposed modules with "Cushion" and existing outline are being organized as demonstrated as follows.

6. References

[1]S. Goel, A. Kumar, M.A. Bayoumi, Design of robust, energy-efficient full adders for deep-submicrometer design using hybrid-CMOS logic style, IEEE Trans. Very Large Scale Integr. (VLSI) Syst, 14 (12) (2006).

[2]P. Bhattacharyya, B. Kundu, S. Ghosh, V. Kumar, A. Dandapat, Performance analysis of a low-power high-speed hybrid 1-bit full adder circuit(2014)

[3]Z. Abid, H. El-Razouk, D. El-Dib, Low power multipliers based on new hybrid full adders Microelectr. J., 39 (12) (2008).

[4]S. Goel, M. Elgamel, M. Bayoumi, Y. Hanafy, Design methodologies for high-performance noise-tolerant XOR-XNOR circuits.

[5]K. Navi, M. Maeen, V. Foroutan, S. Timarchi, O. Kavehei

[6]K. Navi, V. Foroutan, M. Rahimi Azghadi, M. Maeen, M. Ebrahimpour, M. Kaveh, et al.A novel low-power full-adder cell with new technique in designing logical gates based on static CMOS inverter Microelectr.

[7] I. Brzozowski, A. KosDesigning of low-power data oriented addersMicroelectr.

[8]C.-K. Tung, Y.C. Hung, S.H. Shieh, G.S. HuangA low-power high-speed hybrid CMOS full adder for embedded system, Proceeding in Design and Diagnostics of Electronic Circuits and Systems(2007)

[9]H.T. Bui, Y. Wang, Y. JiangDesign and analysis of low-power 10-transistor full adders using novel XOR-XNOR gatesIEEE Trans. Circuits Syst. II Analog Digit. Signal Process.

[10] M. Alioto, G. Palumbo, Analysis and comparison on full adder block in submicron technology IEEE Trans. Very Large Scale Integr. (VLSI) Syst.