|| Volume 5 || Issue 8 || August 2020 || ISSN (Online) 2456-0774
INTERNATIONAL JOURNAL OF ADVANCE SCIENTIFIC RESEARCH
AND ENGINEERING TRENDS

DESIGN OF HYBRID APPROXIMATE ADDERS FOR ENERGY-EFFICIENT APPLICATIONS

MAJJI SIVA[1], MANUMULA SRINUBABU[2]

1 M.Tech, ES&VLSI, Kakinada Institute of Technology and Science, Divili
sivamajji29@gmail.com
2 Assistant Professor, Dept of ECE, Kakinada Institute of Technology and Science, Divili
msrinubabu37@gmail.com

Abstarct : Adders are one of the most widely digital components in the digital integrated circuit design and are the necessary part of Digital Signal Processing (DSP) applications. With the advances in technology, researchers have tried and are trying to design adders which offer either high speed, low power consumption, less area or the combination of them. In this paper, the design of various adders such as Simple yet Accuracy Reconfigurable adders are discussed and the performance parameters of adders such as area and delay are determined and compared. Various adders are designed using Verilog in Virtex-6 FPGA devices. Then, they are simulated and synthesized using Xilinx ISE 14.2 for Virtex-6 family device with speed grade -2.

Keywords : Ripple Carry Adder; Carry Skip Adder; Carry Increment Adder; Carry Look Ahead Adder; Carry Save Adder; Carry Select Adder; Carry Bypass Adder.

I. INTRODUCTION

In Arithmetic unit are the essential blocks of digital systems such as Digital Signal Processor (DSP), microprocessors, microcontrollers, and other data processing units. Adders become a critical hardware unit for the efficient implementation of arithmetic unit. In many arithmetic applications and other kinds of applications, adders are not only in the arithmetic logic unit, but also in other parts of processor. Addition operation can also be used in complement operations (1's, 2's, and so on), encoding, decoding and so on. In general, addition is a process which involves two numbers which are added and carry will be generated. The addition operations will result in sum value and carry value. All complex adder architectures are constructed from its basic building blocks such as Half Adder (HA) and Full Adder (FA).In this paper, an attempt has been made to design and simulate the different types of adders using Verilog language and Xilinix ISE 14.2. Then the performance parameters of the various adders are calculated and compared. The rest of the paper is organized as follows: Section I deals with the introduction about adders. In section II, the designs and the features of various adders are discussed. And section III deals with the simulation and synthesis results of adders.

II. EXISTING METHOD

2.1 Approximate Adders

we consider in this study the exploration of the fifth approximate version of AMA [13] and the Error-Tolerant Adder I [11], because they are also explored by related works [22], [23]. The former approximate adder is renamed to "Copy adder" due to its copy function implemented by the buffers. The approximate adders which are explored in this study can be observed in Figure 1. The "Copy adder" in Figure 1(a) has its k bits long approximate part implemented by buffers to copy the operand a to the sum. This procedure has 50% probability of getting the correct sum for each bit position. Furthermore, the carry-in estimation for the precise part is implemented by the more straightforward assignment of the input operand bit b_{k-1}. This procedure has 75% probability of getting a correct carry-in estimation to the precise block. The use of half adders implements the approximate part of the ETAI in Figure 1(b). The sum is performed in the non-conventional direction, (i.e. from the most significant bit $k-1$ to the least significant bit position 0).

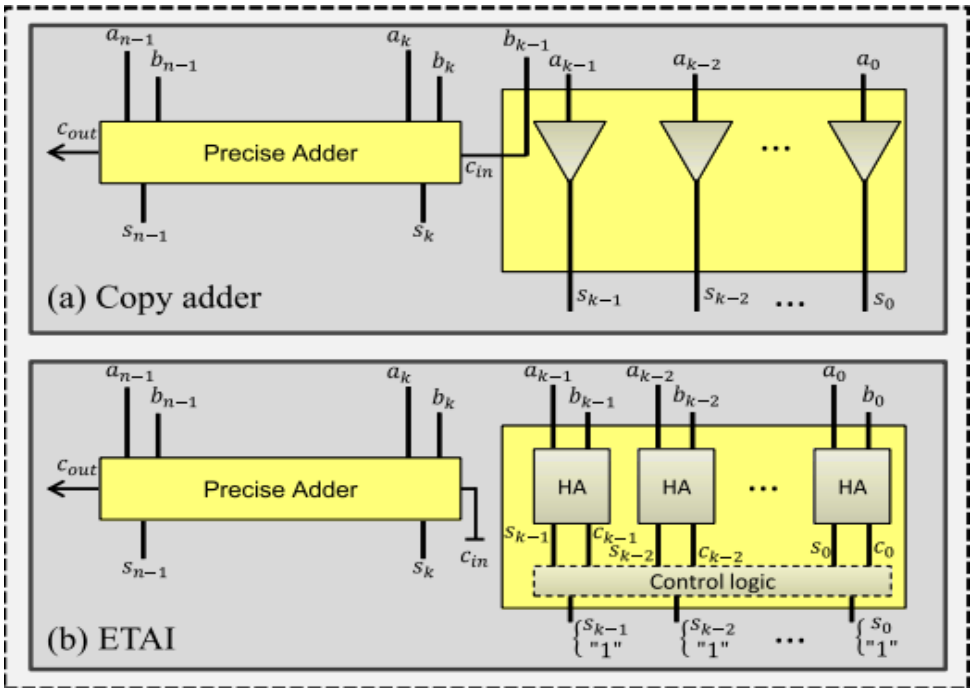


Fig. 1. Approximate adders: (a) Copy adder; (b) ETAI.

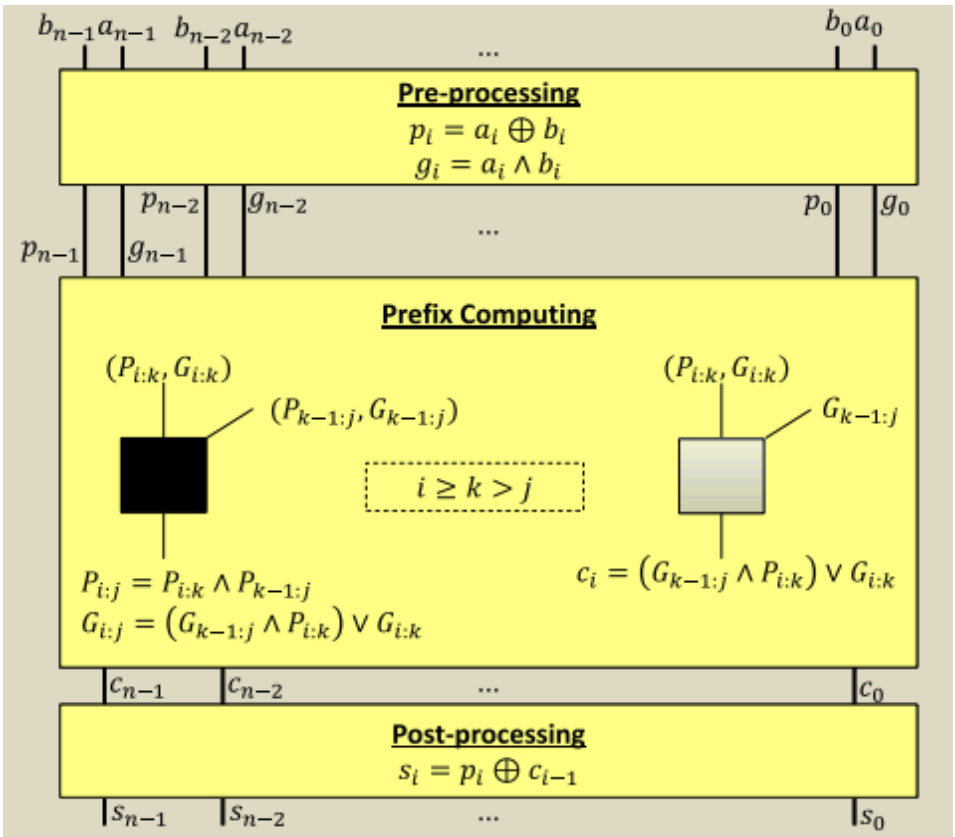


Fig. 2. Parallel prefix adder structure

The control logic block is conceived as follows: if the first carry-generate c is equal to "1," then all the remaining least significant sum bits are set to "1." Otherwise, the sum result is the one computed by the propagate signal. The carry-in to the precise part in ETAI is statically set to "0." This procedure has 50% probability of getting the correct carry-in to the precise part. One can observe that for both the approximate adders, any conventional adder topology can be implemented in the precise part. this work, we explore the

RCA, the high performance PPAs, and the low power adder fully optimized by the commercial tool used in this study. That is why in the next subsection a brief PPA overview is developed.

2.2. Precise Adders

As previously mentioned, adders are fundamental building blocks in a great variety of computational applications. Based on that, many adder topologies have been proposed to deal with the tradeoff between power and computational performance. The RCA topology is characterized to present low values of power consumption, area, and computational performance. Depending on the high-performance application requirements and given that computational complexity is increasing in nowadays tasks, the critical approach is to accelerate the adder's critical path delay (i.e. carry propagation) at the expense of higher area and power dissipation. Based on that, the Parallel Prefix Adders were proposed to deal with high-performance demands [24]. The carry propagation structure in the PPA adders is implemented by simple logic cells which tend to keep a regular connection. Based on that, the sum computation can be divided into pre-processing, prefix computation and post-processing parts, as can be seen in Figure 2. In the pre-processing part, the propagate pi (i.e. ai ⊕bi) and generate gi (i.e ai∧bi) signals are computed based on the input operands ai and bi . In the prefix computation stage, the carry computation is accelerated by the parallel composition of the black cells which implement the group propagate Pi: j and generate Gi: j signals as well as the gray cells which compute the carry ci . Finally, the post-processing step is given by the sum si = pi ⊕ ci−1.

2.3 Drawbacks

At each stage of a carry-save addition,

1. We know the result of the addition at once.
2. We*still do not know* whether the result of the addition is larger or smaller than a given number (for instance, we do not know whether it is positive or negative).

3. PROPOSED METHOD

A. Preliminaries

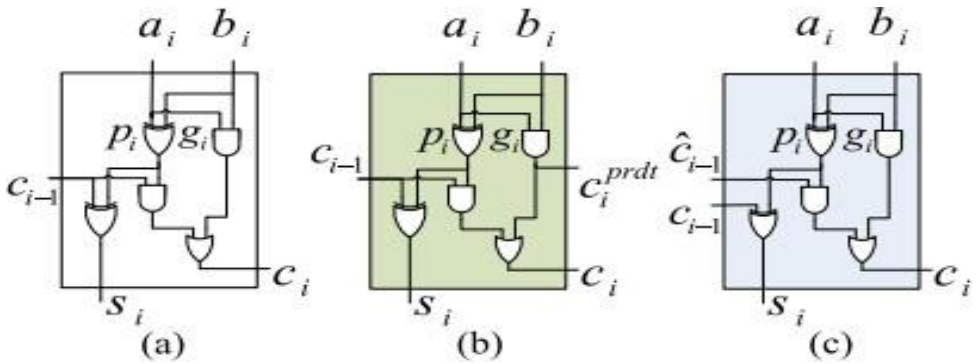


Fig. 3.Conventional full adder. (b) Carry-out selectable full adder. (c) Carry-in configurable full adder.

An N-bit adder operates on two addends A = (aN , aN−1,..., ai,..., a1) and B = (bN , bN−1,..., bi,..., b1). For bit i, its carry-in is ci−1 and its carry-out is ci . Defining the carry generate bit gi = ai · bi, propagate bit pi = ai ⊕ bi , and kill bit ki = ¯ai · b¯i, the conventional full adder computes the sum si and carry ci according to

$$s_i = p_i \oplus c_{i-1} \quad (1)$$

$$c_i = g_i + p_i \cdot c_{i-1}. \quad (2)$$

A gate level schematic of a conventional full adder is provided in Fig. 3(a). A CRA is used to chain N bits of conventional full adders together. By applying (2) recursively, one can get

$$c_i = g_i + p_i\, g_{i-1} + \cdots + g1\ i\ k=2\ pk + c0\ i\ k=1\ pk. \quad (3)$$

This equation implies that ci can be computed directly from g and p of all bits, without waiting for the c of its

lower bits to be computed. This observation is the basis for CLA adder.

c prdt i = gi. (4)

For the LSB of the higher bit subadder, which is bit i + 1, its carry-out ci+1 can be computed using one of two options: either by the conventional ci+1 = gi+1 + pi+1 · ci , or by using the carry prediction as

ci+1 = gi+1 + pi+1 · c prdt i = gi+1 + pi+1 · gi. (5)

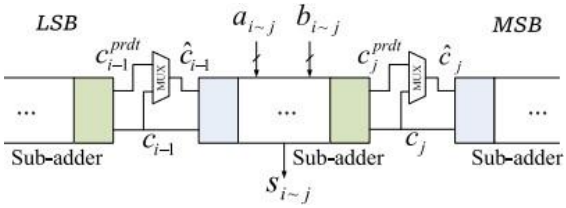


Fig. 4. Design of SARA

The selection between the two options is realized using MUXes as in Fig. 4 and the multiplexer (MUX) selection result is denoted as cˆi . Comparing (5) with (3), we can see that the carry prediction is a truncation-based approximation to carry computation.1 Therefore, cˆi can be configured to either accurate mode or approximation mode, that is cˆi ← c prdt i , if approximation mode ci, if accurate mode. (6)

It should be noted that the carry prediction c prdt i reuses gi in an existing full adder instead of introducing an additional dedicated circuit as in [18] or Fig. 2. This prediction scheme makes a very simple modification to the conventional full adder, as shown in Fig. 3(b). One can connect cˆi to its higher bit i + 1 to compute both carry ci+1 and sum si+1, as in GDA [18] and RAP-CLA [19]. We suggest an improvement over this approach by another simple change as in Fig. 3(c), where si+1 is based on ci instead of cˆi . Such approach can help reduce the error rate in outputs when an incorrect carry is propagated. Because the sum keeps accurate and the carry will not be propagated when addends are exactly the same. Moreover, out of all four configurations of sum/carry calculation by approximate/accurate carry-in, the most meaningful way is to have sum bit calculated by accurate carry and make carry bit configurable.

Therefore, sum si+1 is calculated directly by accurate carry ci without the option of c prdt i . Applying this in SARA as in Fig. 4, in the approximation mode, computing sj+1 from c j can still limit the critical path to be between c prdt i−1 and sj+1, but has higher accuracy than computing sj+1 from cˆj . Compared to sum computation in GDA and RAP-CLA, this technique improves accuracy with almost no additional overhead. Compared to CRA, the overhead of SARA is merely the MUXes, which is almost the minimum possible for configurable adders. Although sj+1 is calculated by accurate carry c j , its delay can still be reduced by approximate carry in lower subadder. In a multibit adder, the delay of sum bit depends on the carry chain propagated from its lower bits. In our SARA structure, even when accurate carry c j is propagated at bit j, the carry chain might be truncated by approximate carry in other lower bits.

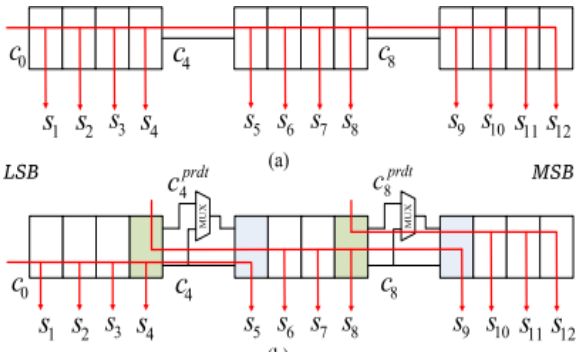


Fig. 5. Implementation of 12-bit adder in (a) CRA and (b) SARA

In Fig. 4, when c prdt i−1 is propagated, the delay of sj+1 is reduced as its path is shorten to be between bit i − 1 and j + 1. We can take the 12-bit adder in Fig. 5 as an example. For 12-bit SARA working in approximate mode, the sum s9 uses the accurate carry c8 from a lower subadder (bits 5 to 8). But c8 is propagated from approximate carry c prdt 4 of another subadder (bits 1 to 4). As shown in the figure, the delay of s9 in SARA is about six stages. Compared with the same bit in CRA, the delay of sum bit s9 in SARA is reduced by three stages. Similar delay reduction can be observed in other sum bits (bits 6 to 12). For sums at bits 1 to 5, their delay is the same as CRA, because they are using an accurate carry c0 from LSB. As a result, the maximum delay in 12-bit SARA is reduced, since for a multibit adder its maximum delay depends on the longest critical path.

4. SIMULATION RESULT

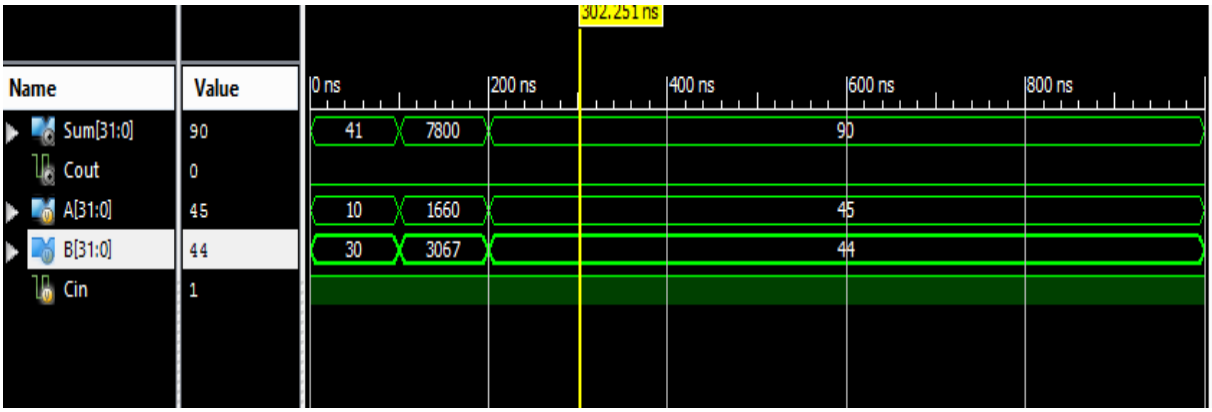


Fig. 6 existing simulation output

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	79	17600	0%
Number of fully used LUT-FF pairs	0	79	0%
Number of bonded IOBs	98	100	98%

Fig. 7 Existing design summary

```
LUT3:I1->O            2    0.195   0.602  C1/r3/FA6/Cout1 (C1/r3/c6)
LUT3:I1->O            2    0.195   0.602  C1/r3/FA7/Cout1 (C1/r3/c7)
LUT3:I1->O            2    0.195   0.602  C1/r3/FA8/Cout1 (C1/Cout3)
LUT3:I1->O            2    0.195   0.602  C1/r4/FA1/Cout1 (C1/r4/c1)
LUT3:I1->O            2    0.195   0.602  C1/r4/FA2/Cout1 (C1/r4/c2)
LUT3:I1->O            2    0.195   0.602  C1/r4/FA3/Cout1 (C1/r4/c3)
LUT3:I1->O            2    0.195   0.602  C1/r4/FA4/Cout1 (C1/r4/c4)
LUT3:I1->O            2    0.195   0.602  C1/r4/FA5/Cout1 (C1/r4/c5)
LUT3:I1->O            2    0.195   0.602  C1/r4/FA6/Cout1 (C1/r4/c6)
LUT3:I1->O            2    0.195   0.602  C1/r4/FA7/Cout1 (C1/r4/c7)
LUT3:I1->O            1    0.195   0.360  C1/r4/FA8/Cout1 (Cout_OBUF)
OBUF:I->O                  3.957          Cout_OBUF (Cout)
----------------------------------------
Total                     30.199ns (10.967ns logic, 19.232ns route)
                                   (36.3% logic, 63.7% route)
```

Fig. 8 Existing time summary

A	B	C	D	E	F	G	H	I
Device			On-Chip	Power (W)	Used	Available	Utilization (%)	
Family	Virtex4		Logic	0.000	63	10944	1	
Part	xc4vfx12		Signals	0.000	132	---	---	
Package	ff668		DCMs	0.000	0	4	0	
Temp Grade	Commercial		IOs	0.000	97	320	30	
Process	Typical		Leakage	0.165				
Speed Grade	-10		Total	0.165				
Environment			Thermal Properties		Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)	
Ambient Temp (C)	50.0				9.3	83.5	51.5	
Use custom TJA?	No							
Custom TJA (C/W)	NA							
Airflow (LFM)	250							
Characterization								
PRODUCTION	v1.0,02-02-08							

Fig. 9 Existing power summary

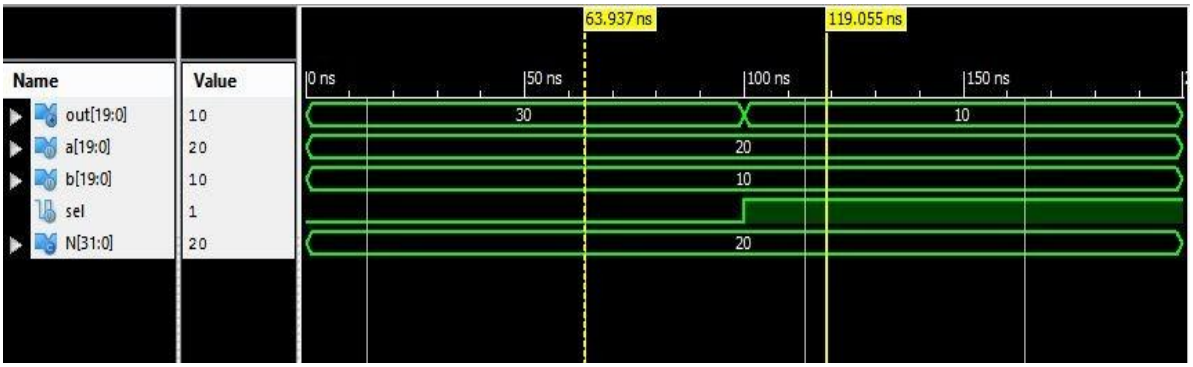


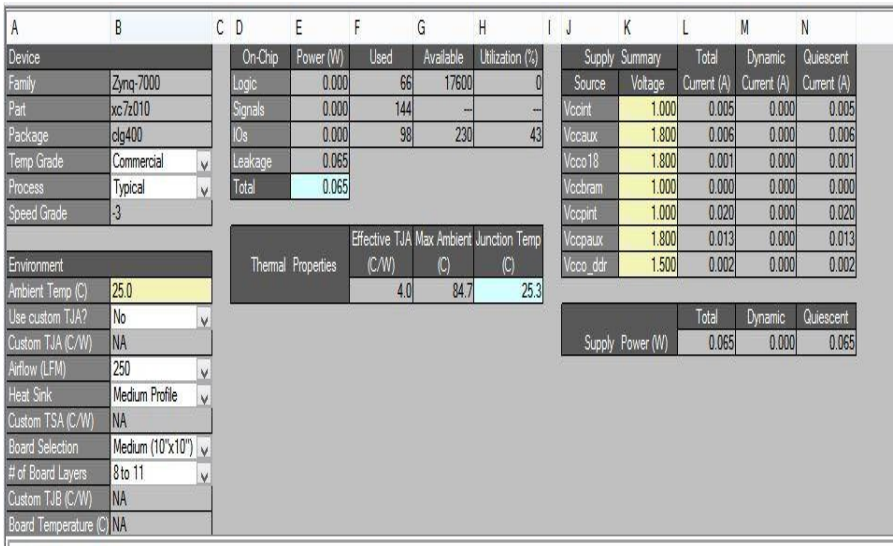
Fig. 10 Proposed simulation output

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	36	5472	0%
Number of 4 input LUTs	63	10944	0%
Number of bonded IOBs	97	320	30%

Fig. 11 Proposed design summary

```
----------------------------------------  ------------
 IBUF:I->O              5   0.000   0.561  a_1_IBUF (a_1_IBUF)
 LUT6:I1->O             3   0.043   0.353  s1/s1/c4/Mxor_sout_xo<0>11 (s1/s1/c4/Mxor
 LUT6:I4->O             3   0.043   0.353  s1/s2/c1/cout1 (s1/s2/c_t<0>)
 LUT5:I3->O             3   0.043   0.353  s1/s2/c3/cout1 (s1/s2/c_t<2>)
 LUT6:I4->O             3   0.043   0.353  s1/generate_N_bit_Adder[8].s3/c1/cout1 (s
 LUT5:I3->O             3   0.043   0.353  s1/generate_N_bit_Adder[8].s3/c3/cout1 (s
 LUT6:I4->O             3   0.043   0.353  s1/generate_N_bit_Adder[12].s3/c1/cout1 (
 LUT5:I3->O             3   0.043   0.353  s1/generate_N_bit_Adder[12].s3/c3/cout1 (
 LUT6:I4->O             3   0.043   0.353  s1/generate_N_bit_Adder[16].s3/c1/cout1 (
 LUT5:I3->O             3   0.043   0.353  s1/generate_N_bit_Adder[16].s3/c3/cout1 (
 LUT6:I4->O             3   0.043   0.353  s1/generate_N_bit_Adder[20].s3/c1/cout1 (
 LUT5:I3->O             3   0.043   0.353  s1/generate_N_bit_Adder[20].s3/c3/cout1 (
 LUT6:I4->O             3   0.043   0.353  s1/generate_N_bit_Adder[24].s3/c1/cout1 (
 LUT5:I3->O             3   0.043   0.353  s1/generate_N_bit_Adder[24].s3/c3/cout1 (
 LUT6:I4->O             3   0.043   0.353  s1/s4/c1/cout1 (s1/s4/c_t<0>)
 LUT5:I3->O             2   0.043   0.433  s1/s4/c3/cout1 (s1/s4/c_t<2>)
 LUT5:I2->O             1   0.043   0.279  Mmux_out261 (out_32_OBUF)
 OBUF:I->O                  0.000          out_32_OBUF (out<32>)
----------------------------------------
Total                       6.904ns (0.688ns logic, 6.216ns route)
                                    (10.0% logic, 90.0% route)
```

Fig. 12 Proposed time summary



Device	
Family	Zynq-7000
Part	xc7z010
Package	clg400
Temp Grade	Commercial
Process	Typical
Speed Grade	-3

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	250
Heat Sink	Medium Profile
Custom TSA (C/W)	NA
Board Selection	Medium (10"x10")
# of Board Layers	8 to 11
Custom TJB (C/W)	NA
Board Temperature (C)	NA

On-Chip	Power (W)	Used	Available	Utilization (%)
Logic	0.000	66	17600	0
Signals	0.000	144	--	--
IOs	0.000	98	230	43
Leakage	0.065			
Total	0.065			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	4.0	84.7	25.3

Supply Summary		Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.000	0.005	0.000	0.005
Vccaux	1.800	0.006	0.000	0.006
Vcco18	1.800	0.001	0.000	0.001
Vccbram	1.000	0.000	0.000	0.000
Vccpint	1.000	0.020	0.000	0.020
Vccpaux	1.800	0.013	0.000	0.013
Vcco_ddr	1.500	0.002	0.000	0.002

	Total	Dynamic	Quiescent
Supply Power (W)	0.065	0.000	0.065

Fig. 13 Proposed power summary

Table 1: comparison table

parameter	Existing	proposed
Number of slices	79	36
Time	30.199	6.904
Power	0.165	0.065

5. CONCLUSION

In this paper, we propose an SARA design. It has significantly lower power/EDP than the latest previous work when comparing at the same accuracy level. In addition, SARA has considerable lower area overhead than almost all the previous works. The accuracy-power-delay efficiency is further improved by a DAR technique. We demonstrate the efficiency of our adder in the applications of multiplication circuits and DCT computing circuits for image processing.

REFERENCES

[1] Kuldeep Rawat, Tarek Darwish and Magdy Bayoumi, "A low power and reduced area Carry Select Adder", 45th Midwest Symposium on Circuits and Systems, vol.1, pp. 467-470, March 2002.

[2] Y. Kim and L.-S. Kim, "64-bit carry-select adder with reduced area," Electron. Lett. vol. 37, no. 10, pp. 614- 615, May 2001.

[3] J. M. Rabaey, Digtal Integrated Circuits-A Design Perspective.Upper Saddle River, NJ: Prentice-Hall,2001.

[4] Cadence, "Encounter user guide, " Version 6.2.4, March 2008.

[5] R. Priya and J. Senthil Kumar, " Enhanced area efficient architecture for 128 bit Modified CSLA", International Conference on Circuits, Power and Computing Technologies,2013.

[6] Shivani Parmar and Kirat pal Singh,"Design of high speed hybrid carry select adder",IEEE ,2012.

[7] I-Chyn Wey, Cheng-Chen Ho, Yi-Sheng Lin, and Chien-Chang Peng," An Area-Efficient Carry Select Adder Design by Sharing the Common Boolean Logic Term", Proceedings of the International MultiConference of Engineers and Computer Scientist 2012 Vol II,IMCES 2012,HongKong,March 14-16 2012.

[8] B. Ramkumar and Harish M Kittur," Low-Power and Area-Efficient Carry Select Adder", IEEE

Transactions on Very Large Scale Integration (VLSI) Systems, VOL. 20, NO. 2, February 2012.
[8] Ms. S.Manjui, Mr. V. Sornagopae," An Efficient SQRT Architecture of Carry Select Adder Design by Common Boolean Logic",IEEE, 2013.
[9] Youngjoon Kim and Lee-Sup Kim, "64-bit carry-select adder with reduced area", Electronics Letters, vol.37, issue 10, pp.614-615, May 2001.

Author Profile:

MAJJI SIVA received the B Tech. degree in electronics and communications engineering and the M Tech degree in **Embedded Systems& VLSI** from the college of KAKINADA INSTITUTE OF TECHNOLOGY AND SCIENCE, DIVILI, respectively.

MANUMULA SRINUBABU received the B Tech. degree in electronics and communications engineering and the M Tech degree in Embedded Systems& VLSI respectively. He is presently an Assistant Professor in the Department of ECE, from the college of KAKINADA INSTITUTE OF TECHNOLOGY AND SCIENCE, DIVILI. His current research and teaching interests are in design for testability, VLSI built-in self-test, VLSI design, fault tolerant computing, and computer architecture, antenna designs and developments.