International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

DESIGN A SIMPLE YET EFFICIENT ACCURACY CARRY CONFIGURABLE ADDER DESIGN

1. KOVIRI DEMUDU, 2. K.KANTHI KINNERA, 3.B.VIJAYA LAKSHMI

[1]M.TECH(VLSI DESIGN),DEPT. OF ECE GONNA INSTITUTE OF INFORMATION TECHNOLOGY AND SCIENCES(JNTU K) VISAKHAPATNAM,INDIA

[2]ASSOCIATE PROFESSOR,DEPT. OF ECE GONNA INSTITUTE OF INFORMATION TECHNOLOGY AND SCIENCES, VISAKHAPATNAM,INDIA.

[3]ASSISTANT PROFESSOR,DEPT. OF ECE GONNA INSTITUTE OF INFORMATION TECHNOLOGY AND SCIENCES, VISAKHAPATNAM,INDIA

ABSTRACT:

High computational complexity of multimedia applications on the portable devices demands a high speed and energy efficient processing cores. The performance of arithmetic unit within these cores significantly affects the overall performance of the devices. Therefore, this paper presents carry look ahead (CLA) adder based two efficient accuracy reconfigurable (AR) adder designs namely AR-CLA-I and AR-CLA-II. The AR-CLA-I design is developed using 4-bit CLA segments with novel approximate sum generation approach. Whereas, the AR-CLA-II design is developed using a novel complementary logic based CLA segment which utilizes proposed new logic formulation. The proposed AR-CLA designs can be reconfigured to achieve high energy-efficiency at cost of acceptable loss in quality. The synthesis results on TSMC 65nm CMOS technology library show that the proposed AR-CLA-I (AR-CLA-II) provides on average 76.52% (80.62%) and 71.65% (74.16%) less area and energy respectively over the best available CLA based approximate adder. The quality metrics of the proposed AR-CLA adder designs as standalone unit demonstrates significant improvement over the existing approximate adder designs. Finally, Sobel edge detectors (SED) embedded with proposed AR-CLA adders provide 71.89 dB higher PSNR over SED embedded with best known approximate adder.

Keywords: CLA, SED, AR-CLA, power efficiency, accuracy, CMOS.

1. INTRODUCTION

The electronics industry has achieved a phenomenal growth over the last few decades, mainly due to the rapid advances in large scale integration technologies and system design applications. With the advent of very large scale integration (VLSI) designs, the number of applications of integrated circuits (ICs) in high-performance computing, controls, telecommunications, image and video processing, and consumer electronics has been rising at a very fast pace. The current cutting-edge technologies such as high resolution and low bit-rate video and cellular communications provide the end-users a marvelous amount of applications, processing power and portability. This

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

trend is expected to grow rapidly, with very important implications on VLSI design and systems design.

Approximate computing has emerged as a potential solution for the design of energy-efficient digital systems. Applications such as multimedia, recognition and data mining are inherently error-tolerant and do not require a perfect accuracy in computation. For these applications, approximate circuits may play an important role as a promising alternative for reducing area, power and delay in digital systems that can tolerate some loss of accuracy, thereby achieving better performance in energy efficiency. As one of the key components in arithmetic circuits, adders have been extensively studied for approximate implementation (see [1] for a review). New methodologies to model, analyze and evaluate the approximate adders have been discussed in [2]–[4]. However, there has been relatively less effort in the design of approximate multipliers. A multiplier usually consists of three stages: partial product generation, partial product accumulation and a carry propagation adder (CPA) at the final stage. Considers using approximate adders to generate the radix-8 Booth encoding 3x with error reduction. Approximate partial products are computed using inaccurate 2 × 2 multiplier blocks, while accurate adders are used in an adder tree to accumulate the approximate partial products.

2. RELATED STUDY

As the scale of integration keeps growing, more and more sophisticated signal processing systems are being implemented on a VLSI chip. These signal processing applications not only demand great computation capacity but also consume considerable amount of energy. While performance and Area remain to be the two major design tolls, power consumption has become a critical concern in today's VLSI system design[]. The need for low-power VLSI system arises from two main forces. First, with the steady growth of operating frequency and processing capacity per chip, large currents have to be delivered and the heat due to large power consumption must be removed by proper cooling techniques. Second, battery life in portable electronic devices is limited. Low power design directly leads to prolonged operation time in these portable devices. Addition usually impacts widely the overall performance of digital systems and a crucial arithmetic function. In electronic applications adders are most widely used. Applications where these are used are multipliers, DSP to execute various algorithms like FFT, FIR and IIR. Wherever concept of multiplication comes adders come in to the picture. As we know millions of instructions per second are performed in microprocessors. So, speed of operation is the most important constraint to be considered while designing multipliers. Due to device portability miniaturization of device should be high and power consumption should be low. Devices like Mobile, Laptops etc. require more battery backup. So, a VLSI designer has to optimize these three parameters in a design. These constraints are very difficult to achieve so depending on demand or application some compromise between constraints has to be made. Ripple carry adders exhibits the

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

most compact design but the slowest in speed. Whereas carry look ahead is the fastest one but consumes more area. Carry select adders act as a compromise between the two adders. In 2002, a new concept of hybrid adders is presented to speed up addition process by Wang et al. that gives hybrid carry look-ahead/carry select adders design. In 2008, low power multipliers based on new hybrid full adders is presented. There are four factors that influence the power dissipation of CMOS circuits. They are technology, circuit design style, architecture, and algorithm. The challenge of meeting the contradicting goals of high performance and low power system operation has motivated the development of low power process technologies and the scaling of device feature sizes.

3. AN OVERVIEW OF PROPOSED SYSTEM

A ripple-carry adder works in the same way as pencil-and-paper methods of addition. Starting at the rightmost (least significant) digit position, the two corresponding digits are added and a result obtained. It is also possible that there may be a carry out of this digit position (for example, in pencil-and-paper methods, "9 + 5 = 4, carry 1"). Accordingly, all digit positions other than the rightmost one need to take into account the possibility of having to add an extra 1 from a carry that has come in from the next position to the right.

This means that no digit position can have an absolutely final value until it has been established whether or not a carry is coming in from the right. Moreover, if the sum without a carry is 9 (in pencil-and-paper methods) or 1 (in binary arithmetic), it is not even possible to tell whether or not a given digit position is going to pass on a carry to the position on its left. At worst, when a whole sequence of sums comes to …99999999… (in decimal) or …11111111… (in binary), nothing can be deduced at all until the value of the carry coming in from the right is known, and that carry is then propagated to the left, one step at a time, as each digit position evaluated "9 + 1 = 0, carry 1" or "1 + 1 = 0, carry 1". It is the "rippling" of the carry from right to left that gives a ripple-carry adder its name, and its slowness. When adding 32-bit integers, for instance, allowance has to be made for the possibility that a carry could have to ripple through every one of the 32 one-bit adders. Carry-look ahead depends on two things:

1. Calculating for each digit position whether that position is going to propagate a carry if one comes in from the right.
2. Combining these calculated values to be able to deduce quickly whether, for each group of digits, that group is going to propagate a carry that comes in from the right.

Supposing that groups of four digits are chosen the sequence of events goes something like this:

1. All 1-bit adders calculate their results. Simultaneously, the look ahead units perform their calculations.
2. Assuming that a carry arises in a particular group that carry will emerge at the left-hand end of the group within at most five gate

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

delays and start propagating through the group to its left.

3. If that carry is going to propagate all the way through the next group, the lookahead unit will already have deduced this. Accordingly, *before the carry emerges from the next group*, the lookahead unit is immediately (within one gate delay) able to tell the *next* group to the left that it is going to receive a carry – and, at the same time, to tell the next lookahead unit to the left that a carry is on its way.

The net effect is that the carries start by propagating slowly through each 4-bit group, just as in a ripple-carry system, but then move four times as fast, leaping from one lookahead-carry unit to the next. Finally, within each group that receives a carry, the carry propagates slowly within the digits in that group.

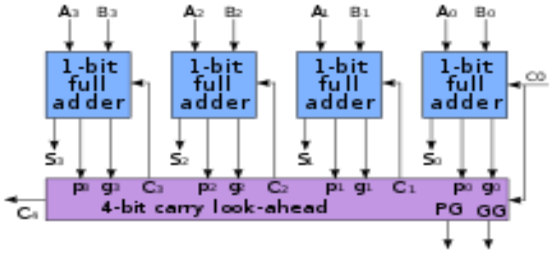

Fig.3.1. 4-bit adder with carry lookahead.

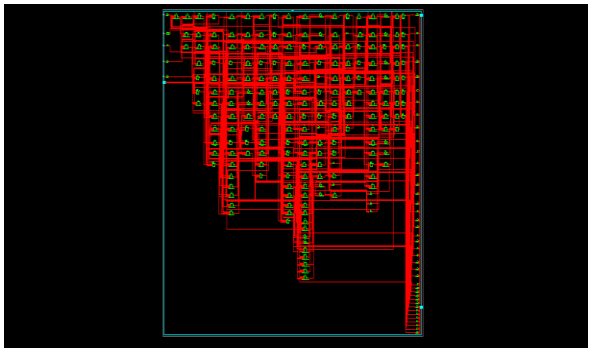

Fig.3.2. Schematic diagram.

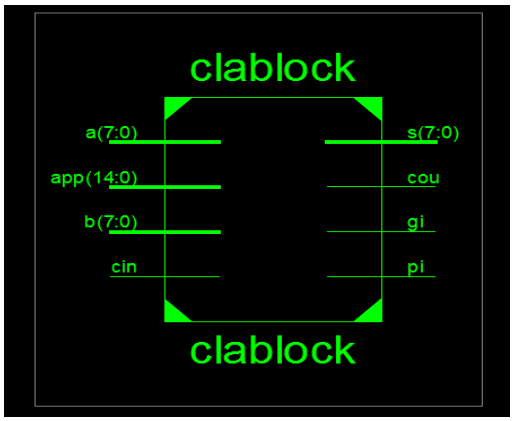

Fig.3.3. RTL BLOCK DIAGRAM

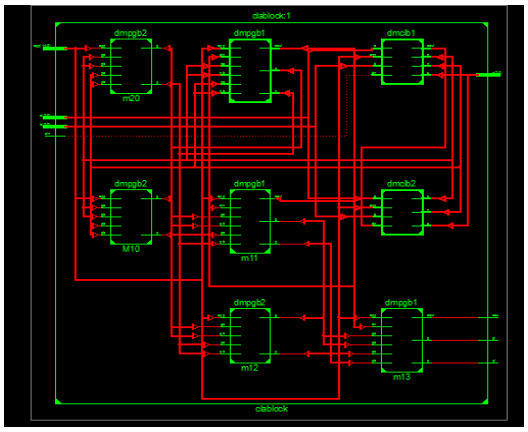

Fig.3.4 . INTERNAL BLOCK DIAGRAM

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of 4 input LUTs	43	17,344	1%	
Number of occupied Slices	24	8,672	1%	
Number of Slices containing only related logic	24	24	100%	
Number of Slices containing unrelated logic	0	24	0%	
Total Number of 4 input LUTs	43	17,344	1%	
Number of bonded IOBs	43	190	22%	
Average Fanout of Non-Clock Nets	2.31			

Fig.3.5. Area

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

```
----------------------------------------
Total                     11.337ns (7.947ns logic, 3.390ns route)
                                   (70.1% logic, 29.9% route)

=========================================================================


Total REAL time to Xst completion: 8.00 secs
Total CPU time to Xst completion: 7.88 secs
```

Fig.3.6. Delay

Fig.3.7. Power

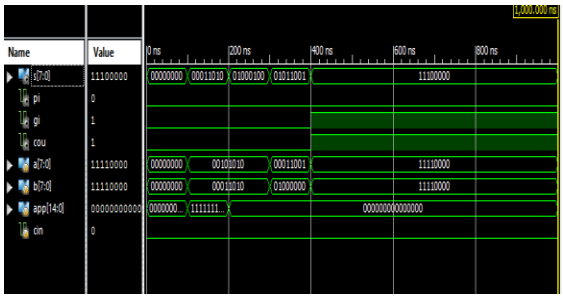

Fig.3.8 OUTPUT RESULTS.

4. CONCLUSION

An approximate Adder. There is small degradation in image quality which is tolerable by human eye. The overall area and Delay and Frequency analysis are presented and compared. From the results we can depict that approximately up to 25 to 35% of reduction at all levels are achieved. So due to this we use approximation, which will minimize delay. This design is particularly useful in computation-intensive applications which are robust to small errors in computation. The potential applications of this approximate Adder fall mainly in areas where there is no strict requirement on accuracy or where super-low power consumption and high speed performance are more important than the accuracy. One example of such applications is in the DSP application for portable devices such as cell phones and laptops.

REFERENCES

[1] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in Proc. IEEE Eur. Test Symp., May 2013, pp. 1–6.

[2] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," IEEE Trans. Comput.- Aided Design Integr. Circuits Syst., vol. 32, no. 1, pp. 124–137, Jan. 2013.

[3] V. Chippa, A. Raghunathan, K. Roy, and S. Chakradhar, "Dynamic effort scaling: Managing the quality-efficiency tradeoff," in Proc. Design Autom. Conf. (DAC), Jun. 2011, pp. 603–608.

[4] S.-L. Lu, "Speeding up processing with approximation circuits," Computer, vol. 37, no. 3, pp. 67–73, Mar. 2004.

[5] S. Venkataramani, K. Roy, and A. Raghunathan, "Substitute-andsimplify: A unified design paradigm for approximate and quality configurable circuits," in Proc. Conf. Design, Autom. Test Eur. (DATE), 2013, pp. 1367–1372.

[6] K. Du, P. Varman, and K. Mohanram, "High performance reliable variable latency carry select addition," in Proc. Conf. Design, Autom. Test Eur. (DATE), 2012, pp. 1257–1262.

[7] A. K. Verma, P. Brisk, and P. Ienne, "Variable latency speculative addition: A new paradigm for arithmetic circuit design," in Proc. Conf. Design, Autom. Test Eur. (DATE), 2008, pp. 1250–1255.
[8] N. Zhu, W. L. Goh, and K. S. Yeo, "An enhanced low-power high-speed adder for error-tolerant application," in Proc. Int. Symp. Integr. Circuits, 2009, pp. 69–72.
[9] N. Zhu, W. L. Goh, W. Zhang, K. S. Yeo, and Z. H. Kong, "Design of low-power high-speed truncation-error-tolerant adder and its application in digital signal processing," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 8, pp. 1225–1229, Aug. 2010.
[10] N. Zhu, W. L. Goh, and K. S. Yeo, "Ultra low-power high-speed flexible probabilistic adder for error-tolerant applications," in Proc. Int. SoC Design Conf. (ISOCC), 2011, pp. 393–396.
[11] G. Liu, Y. Tao, M. Tan, and Z. Zhang, "CASA: Correlation-aware speculative adders," in Proc. Int. Symp. Low Power Electron. Design, 2014, pp. 189–194