International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 03 | Mar 2019 www.irjet.net p-ISSN: 2395-0072

FPGA implementation of low power configurable adder for approximate computing

Mrs.B.Sathyabhama, T.Anitha, G.Bhavani , K.R Praveena, A.Charumathi

[1]*Professor, Dept. of ECE, Panimalar Engineering College, Poonamalle, TamilNadu, India.*
[2, 3, 4,5]*UG students, Dept. of ECE, Panimalar Engineering College, Poonamalle, TamilNadu, India.*

Abstract *- Approximate computing is an efficient approach for error-tolerant applications because it can trade off accuracy for power. Addition is a key fundamental function for these applications. In this paper, we proposed a low-power yet high speed accuracy-configurable adder that also maintains a small design area. The proposed adder is based on the conventional carry look-ahead adder, and its configurability of accuracy is realized by masking the carry propagation at runtime. Compared with the conventional carry look-ahead adder, it have area overhead, the proposed 16-bit adder reduced power consumption, and critical path delay most according to the accuracy configuration settings, respectively. Furthermore, compared with other previously studied adders, the experimental results demonstrate that the proposed adder achieved the original purpose of optimizing both power and speed simultaneously without reducing the accuracy.*

1.INTRODUCTION

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. Commonly used multimedia applications have Digital Signal Processing (DSP) blocks as their backbone. Most of these DSP blocks implement image and video processing algorithms, where the ultimate output is either an image or a video for human consumption. The limited perception of human vision allows the outputs of these algorithms to be numerically approximate rather than accurate. This relaxation on numerical exactness provides some freedom to carry out imprecise or approximate computation. The freedom can be taken advantage of to come up with low-power designs at different levels of design abstraction, viz. logic, architecture, and algorithm.

EXISTING SYSTEM

An accuracy gracefully-degrading adder (GDA) which allows the accurate and approximate sums of its sub adders to be selected at any time. Our adder proposed in this paper does not consider a pipeline structure either. To generate outputs with different levels of computation accuracy and to obtain the configurability of accuracy, some multiplexers and additional logic blocks are required. However, the additional logic blocks require more area. Furthermore, these blocks will cause power wastage when their outputs are not used to generate a sum.

This problem was addressed based on a low-power configurable adder that generates an approximate sum by using OR gates. The proposed adder also uses OR gates to generate an approximate sum, and focuses on only power consumption and its delay is large. Thus, it may fail to meet the speed requirement of an application

BLOCK DIAGRAM OF CARRY MASKABLE ADDER

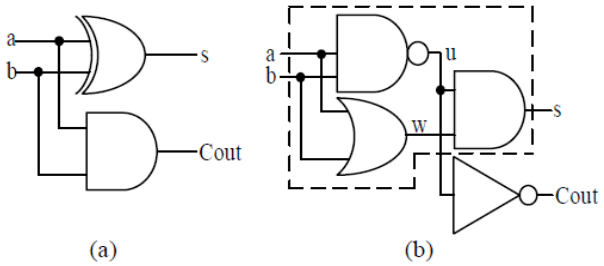


Figure 2: (a) Conventional half adder, and (b) equivalent circuit of a half adder.

Table 1: Truth table for the equivalent circuit of a half adder.

Inputs		Internal signals		Outputs	
a	b	u	w	s	Cout
0	0	1	0	0	0
0	1	1	1	1	0
1	0	1	1	1	0
1	1	0	1	0	1

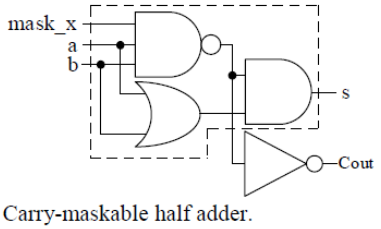


Figure 3: Carry-maskable half adder.

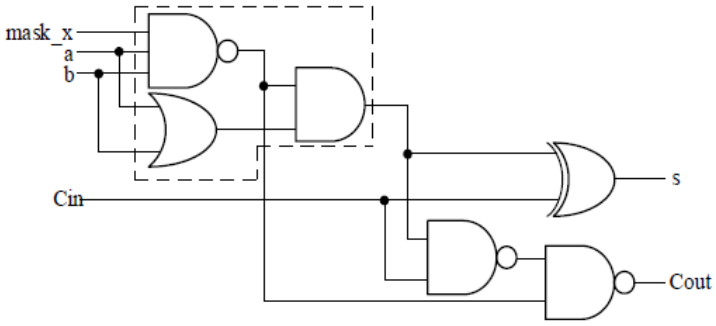


Figure 4: Carry-maskable full adder.

PROPOSED SYSTEM

In this paper, we propose a configurable approximate adder, which consumes lesser power with a comparable delay and area. In addition, the delay observed with the proposed adder is smaller with comparable power vaccuracy configurability the proposed adder achieved the optimization of power and delay simultaneously and with no bias toward either. We implemented the proposed adder, the conventional carry look-ahead adder (CLA), and the ripple carry adder (RCA) in Verilog HDL. Then we evaluated the power consumption, critical path delay, and design area for each of these implementations. Compared with the conventional CLA, mean relative error distance (MRED), the proposed adder reduced power consumption and critical path delay. We provided a crosswise comparison to demonstrate the superiority of the proposed adder. Moreover, we implemented two previously studied configurable adders to evaluate power consumption, critical path delay, design area, and accuracy

MODULE EXPLANATION:

ACCURACY-CONFIGURABLE ADDER:

Typically, a CLA consists of three parts: (1) half adders for carry generation (G) and propagation (P) signals preparation, (2) carry look-ahead units for carry generation, and (3) XOR gates for sum generation. We focus on the half adders for G and P signals preparation in part 1. Consider an n-bit CLA; each part of it can be obtained as follows:

$$P_i = A_i \oplus B_i,\ G_i = A_i \cdot B_i,$$
$$C_i = G_i + P_i \cdot C_{i-1},$$
$$S_i = P_i \oplus C_{i-1}.$$

Where, *i*is denoted the bit position from the least significant bit. Note that owing to reuse of the circuit of A*i* *XOR* B*i* for S*i* generation, here P*i* is defined as A*i* *XOR* B*i* instead of A*i* *OR* B*i*. Because C0 is equal to G0, if G0 is 0, C0 will be 0. We find that C1 is equal to G1 when C0 is 0. In other words, if G0 and G1 are equal to 0, C0 and C1 will be 0. By expanding the above to *i,* C*i* will be 0 when G0, G1, …, G*i* are all 0. This means that the carry propagation from C0 to C*i* is masked. We can obtain that S*i* is equal to P*i* when C*i*-1 is 0.

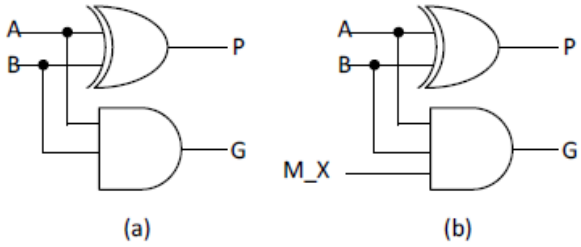
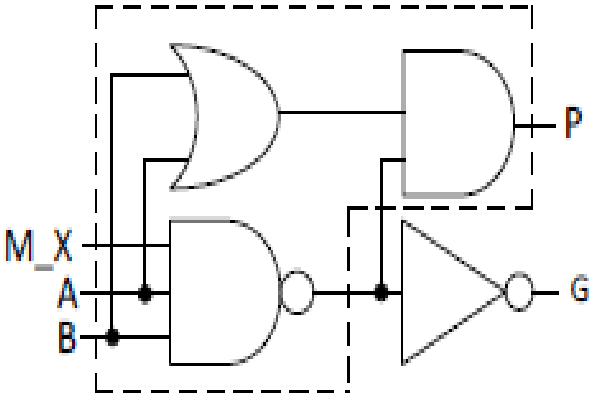


Fig. (a) An accurate half adder, and (b) a half adder with a select signal.

From the perspective of approximate computing, if G is controllable and can be controlled to be 0, the carry propagation will be masked and S (=P) can be considered as an approximate sum. In other words, we can obtain the selectivity of S between the accurate and approximate sum if we can control G to be A *AND* B or 0. Evidently, we can achieve selectivity by adding a select signal. Figure (a) is a conventional half adder and Fig. (b) is a half adder to which the select signal has been added. Compared with the conventional half adder, we add a signal named "M_X" as the select signal and use a 3-input AND gate to replace the 2-input one. When M_X = 1, the function of G is the same as that of a conventional half adder; when M_X = 0, G is equal to 0.

Fig. A carry-maskable half adder.



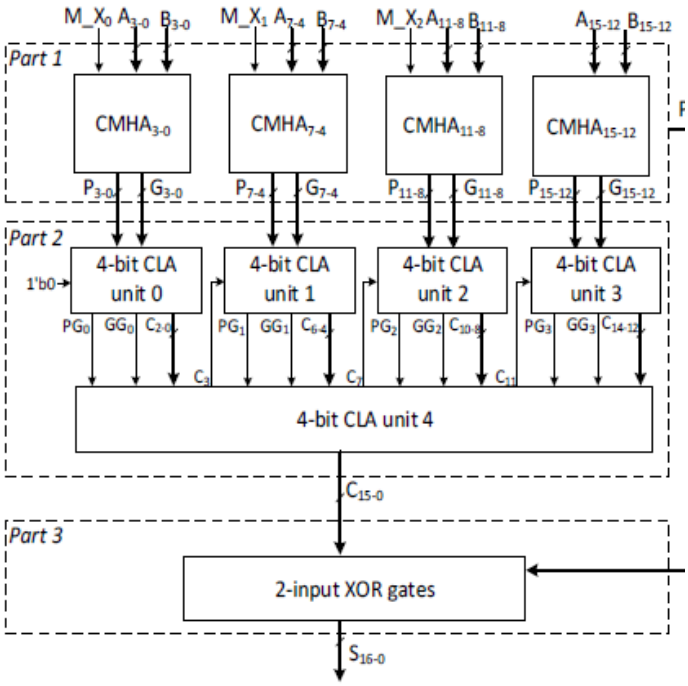


Fig: Structure of the proposed 16-bit adder.

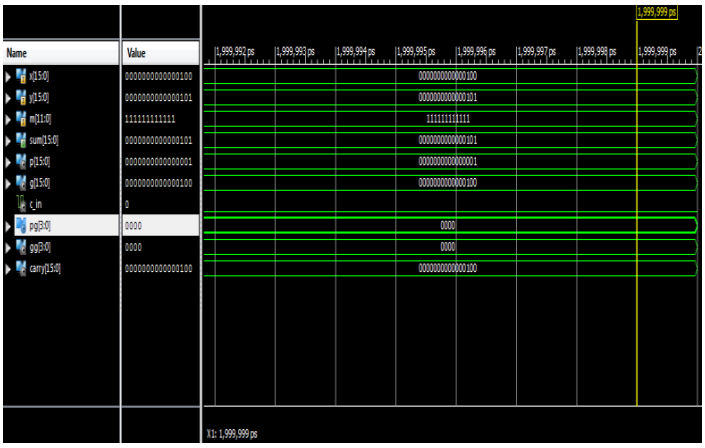
The structure of the proposed 16-bit adder is shown in Fig. as an example. Four groups (CMHA3-0, CMHA7-4, CMHA11- 8, and CMHA15-12) are used to prepare the P and G signals. Each group comprises four CMHAs There is no mask signal for CMHA15-12 in this example; therefore, accurate P15-12 (= A15-12 *XOR* B15-12) and G15-12 (= A15-12 *AND* B15-12) are always obtained. P15-0 and G15-0 are the outputs from Part 1 and are connected to Part 2. Note that P15-0 is also connected to Part 3 for sum generation. In Part 2, four 4-bit carry look-ahead units (unit 0, 1, 2, 3) generate four PGs (PG0, PG1, PG2, and PG3), four GGs (GG0, GG1, GG2, and GG3), and 12 carries (C2-0, C6-4, C10-8, and C14-12) first, and then the carry look-ahead unit 4 generates the remaining four carries (C3, C7, C11, and C15) by using the PGs and GGs. C15-0 is the output of Part 2 and is connected to Part 3. The fifteen 2-input XOR gates in Part 3 generate the sum.

OUTPUT:

EXISTING TECHNIQUE:

IF M-X = 1 ,CARRY will generate,

If m_x = 0, CARRY will not be generated.



.

TIMIMG ANALYSIS

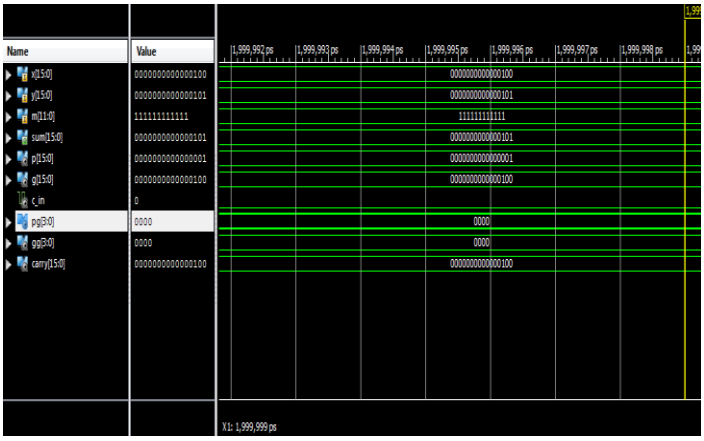
```
Timing Summary:
---------------
Speed Grade: -3

   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: 18.275ns

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

=========================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 796 / 32
-------------------------------------------------------------------------
Delay:               18.275ns (Levels of Logic = 15)
  Source:            a<0> (PAD)
  Destination:       carry<15> (PAD)
```

DESIGN SUMMARY:

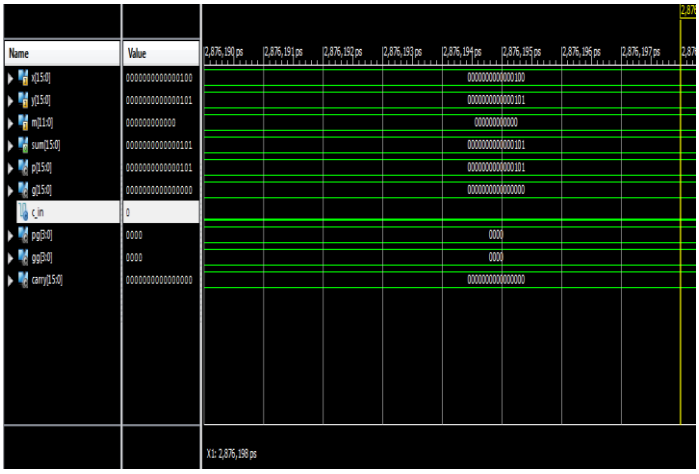
Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	39	5720	
Number of fully used LUT-FF pairs	0	39	
Number of bonded IOBs	76	102	

PROPOSED CODE:

IF M-X = 1 ,CARRY will generate,



If M_X = 0, CARRY will not be generated.



DESIGN SUMMARY (AREA ANALYSIS):

Device Utilization Summary				
Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	0	11,440	0%	
Number of Slice LUTs	24	5,720	1%	
Number used as logic	24	5,720	1%	
Number using O6 output only	18			
Number using O5 output only	0			

TIMING ANALYS

```
Timing Summary:
---------------
Speed Grade: -3

   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: 13.118ns

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

=========================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 637 / 16
-------------------------------------------------------------------------
Delay:              13.118ns (Levels of Logic = 11)
  Source:           m<1> (PAD)
  Destination:      sum<15> (PAD)
```

CONCLUSIONS

without suffering the cost of the increase in power or in delay for configurability was proposed. The proposed adder is based on the conventional CLA, and its configurability of accuracy is realized by masking the carry propagation at runtime. The experimental results demonstrate that the proposed adder delivers significant power savings and speedup with a small area overhead than those of the conventional CLA. Furthermore, compared with previously studied configurable adders, the experimental results demonstrate that

the proposed adder achieves the original purpose of delivering an unbiased optimized result between power and delay without sacrificing accuracy. It was also found that the quality requirements of the evaluated application were not compromised.

ACKNOWLEDGEMENT:

We would like to express our special thanks of gratitude to all my teachers as well as our principal who gave us the golden opportunity to do this wonderful project on this topic.

DESCRIPTION	AREA ANALYSIS	TIMING ANALYSIS
EXISTING TECHNIQUE	39 LUT's	18.275 ns
PROPOSED TECHNIQUE	24 LUT's	13.118 ns

REFERENCES

- "Energy-Aware Probabilistic Multiplier: Design and Analysis**",** Mark S. K. Lau, Keck-Voon Ling, Yun-Chung Chu.
- "Bio-Inspired Imprecise Computational Blocks for Efficient VLSI Implementation of Soft-Computing Applications", H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas.
- "A Low-Power Configurable Adder for Approximate Applications", Tongxin Yang TomoakiUkezonoToshinori Sato.
- "Low-Power Digital Signal Processing Using Approximate Adders", Vaibhav Gupta, Debabrata Mohapatra, Anand Raghunathan and Kaushik Roy.
- "On Reconfiguration-Oriented Approximate Adder Design and Its Application", Rong Ye, Ting Wang, Feng Yuan, Rakesh Kumar and Qiang Xu.
- "Addition related arithmetic operations via controlled transport of charge", Sorin Cotofana , Casper Lageweg , Stamatis Vasiallidis.