VLSI Implementation of Heterogeneous Adder for Performance Optimization

Raminder Preet Pal Singh
Dept. ECE, Arni University
Kathgarh, Himachal Pardesh, India

Ashish Chaturvedi
Dept. CSE, Arni University
Kathgarh, Himachal Pardesh, India

ABSTRACT

An Adder is one of the significant hardware blocks in most digital systems such as digital signal processors and microprocessors etc. Over the last few decades lot of research have been carried out in order to design an efficient adder circuits in terms of compactness, high speed and low power consumption. However, area and speed are two conflict parameters. So, improving speed results always in larger area occupied by circuit on chip and vice-versa. In order to design an optimized adder circuit which provides area/delay tradeoffs, we studied different available parallel, synchronous adders and proposed a new adder based on combination of them. In this paper, we proposed a new type of adder architecture known as heterogeneous adder that consists of concatenation of sub-adder (homogeneous adder) of different types. The heterogeneous adder architecture provides better design tradeoffs in terms of area and delay characteristics.

Keywords

Adder, Ripple carry adder, Lookahead carry adder, VHDL simulation.

1. INTRODUCTION

Addition is one of the most common and often used arithmetic operations among a set of real-time digital system processing benchmarks. Therefore, large numbers of research have been carried out to design an optimized adder circuit. In this paper, we study parallel adders where all the inputs are available before the start of the computation. The parallel adders we have chosen for our experiment vary widely in their delay and speed characteristics. Since asynchronous systems are not very common, therefore we chose synchronous adders for our experiment. The adders studied are linear time ripple carry adder, Ling adder, square-root time carry skip adder and logarithmic time lookahead carry adder. These single (homogeneous adder) has their own benefits and limitations w.r.t. performance parameters e.g. implementing Ripple carry adder utilizes less area but at the cost of large delay, whereas, Lookahead carry adder gives delay efficient design but at the cost of large chip area requirement. Therefore, for efficient design various hybrid architecture were proposed by adopting a different scheme for carry and sum generation. In this paper, we propose new architecture that combine different types of adder to form a single heterogeneous adder to satisfy design constraints.

The adders presented in this paper are all modeled by using VHDL for 16-bit unsigned data. XILINX ISE v 9.1i is used as synthesis tool and FPGA-Spartan III (XC3S250E) device is selected to get area report. Modelsim XE III 6.2g is used to get timing simulation.

2. PRIOR WORK

In 1990, modified Carry-Skip Adders was presented by reducing first block delay with carry-lookahead adders using multidimensional dynamic programming [13]. In 1996, transistor-level simulation of the adders using HSPICE is done for area, time and power trade-off between different fast adders [6]. 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 [7]. In 2007, a new 54×54-bit multiplier is designed using high-speed carry-look-ahead adder and has been fabricated by CMOS technology [4]. In 2008, low power multipliers based on new hybrid full adders is presented [5]. In 2008, Hasan Krad et al worked on the performance analysis for a 32-Bit Multiplier with a Carry-Look-Ahead Adder and a 32-bit Multiplier with a Ripple Adder using VHDL [3].

3. FAST PARALLEL ADDERS

3.1 Ling Adder Design

The Ling Adder is a type of Look-Ahead Adder with a slight modification that results in significant hardware saving. Ling's modification consists of propagating hi = ci + ci-1 instead of ci. This result in reduction of number of gates required for implementation. Therefore, the Boolean expression for calculating next carry and sum are:

$$Ci = hi\ (Gi\text{-}1 + Pi\text{-}1) \quad (1)$$

$$Si = Pi \text{ xor } hi\ (Gi\text{-}1 + Pi\text{-}1) \quad (2)$$

Where Pi is Carry propagation and Gi is Carry generation.

3.2 Hybrid Carry Skip Adder (CSKA)

In case of N-bit Ripple carry adder, carry has to propagate through all N stages, which results in large delay in performing binary addition. In contrast, it is possible to skip carry over group of n-bits in case of Carry Skip Adder. This result in less delay as compare to ripple carry adder. Figure.3 shows 16-bit Carry-Skip Adder divided into 4 blocks and each block are a 4-bit Lookahead Carry Adder.

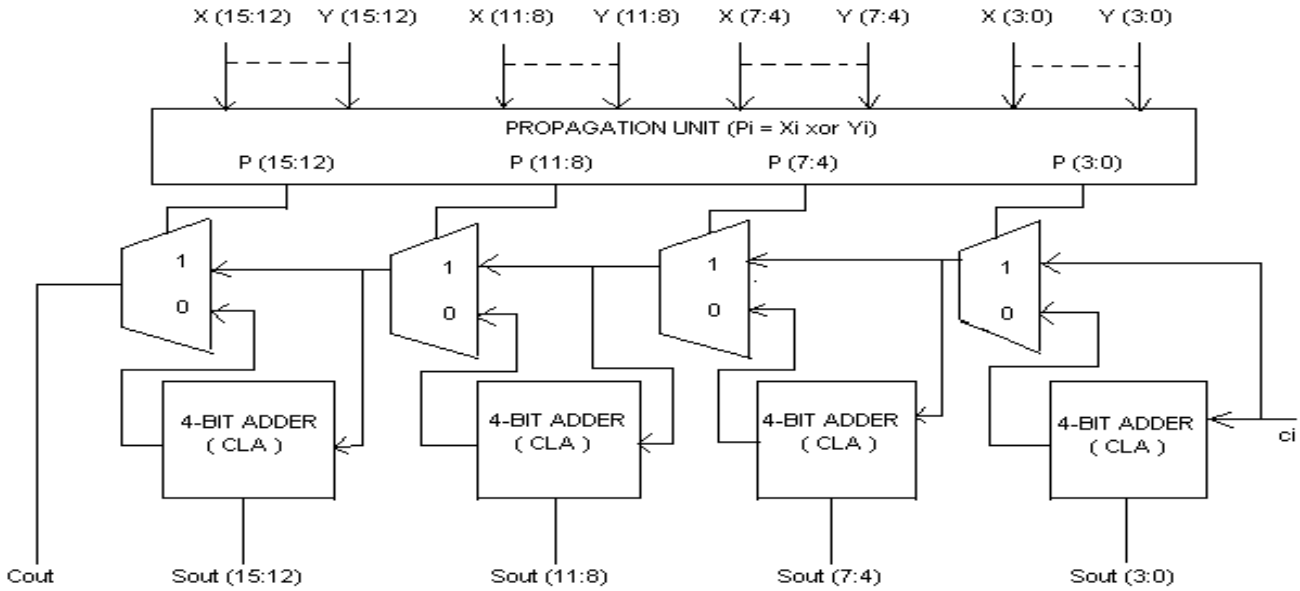

Figure 1: 16-bit Carry Skip adders.

VHDL implimentation and performance analysis of Ling and carry skip adder is given in the Table 1.

Table 1: **16-bit Ling and Carry skip adder design comparison for Area and Delay**

ADDERS	GATES COUNT	DELAY (ns)	
		SUM	CARRY
CSKA	348	15.47	12.16
LING	240	23.3	23.12

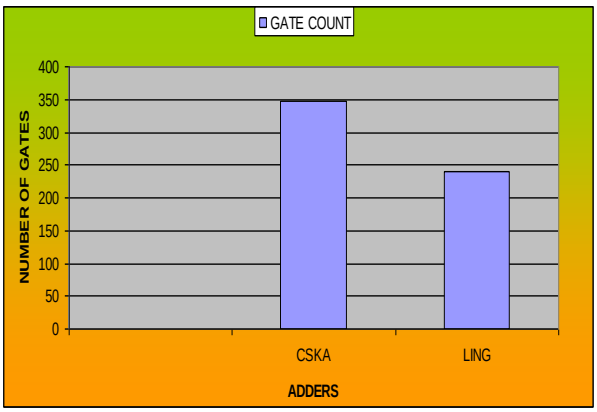

Figure 2: **Hardware utilization comparison of CSKA and Ling adder design.**

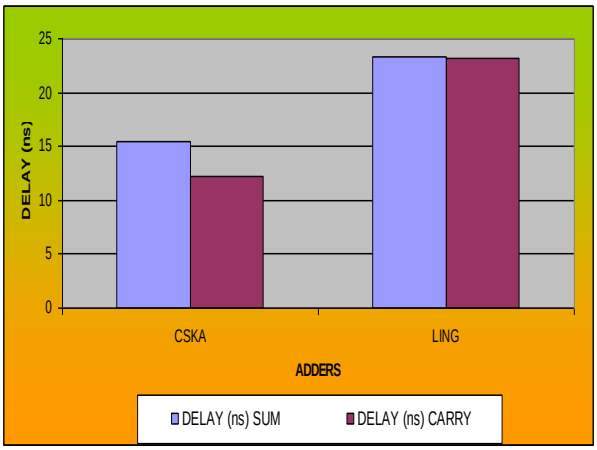

Figure 3: **Speed of operation comparison between CSKA and Ling adder design**

4. PROPOSED HETEROGENEOUS ADDER DESIGN

16-bit Heterogeneous adder proposed in this paper consists of two sub adders SA1 and SA2. Sub adder (SA1) consists of 12-bit Ling architecture and Sub adder SA2 consists of 4-bit Carry skip adder architecture. Both sub adders concatenates to form a heterogeneous adder.

The order of sub adder has an impact on the performance of a heterogeneous adder. From fig. 3, it is observed that Ling adder requires less area but at the cost of large delay and vice-versa in case of hybrid carry skip adder.

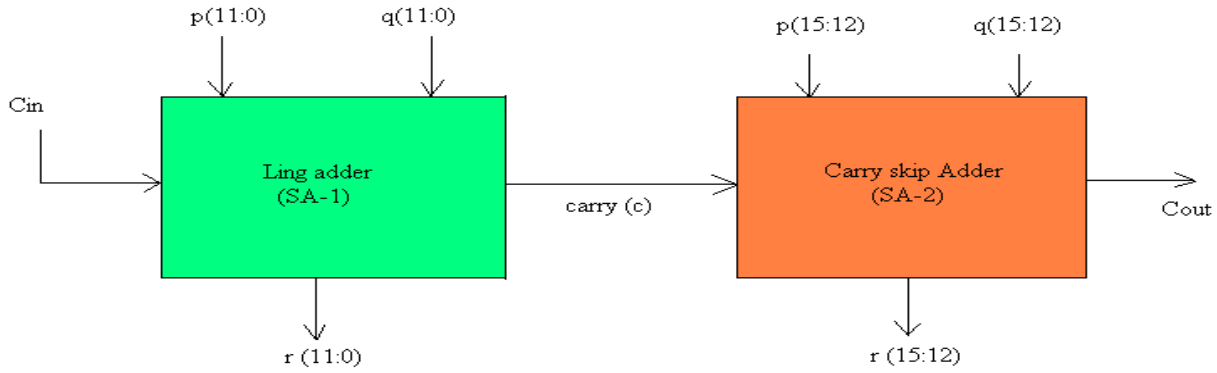

Figure 4: A Heterogeneous Adder

Therefore, in order to get optimized result in terms of area utilization and speed of operation we combine two adders to form a single adder with different bit size. Now, the main point of is that how to choose number of bits for each sub-adder? As shown in fig.4, we took 12-bits out of 16 for Ling adder and 4-bit for carry skip adder in order to reduce hardware utilization but not at the cost of speed of operation.

4. SIMULATION RESULTS

Table 2: Area and Delay report for adders

ADDERS	GATE COUNT	DELAY (ns)	
		SUM	CARRY
CSKA	348	15.47	12.16
LING	240	23.3	23.12
HETRO	231	22.48	20.64

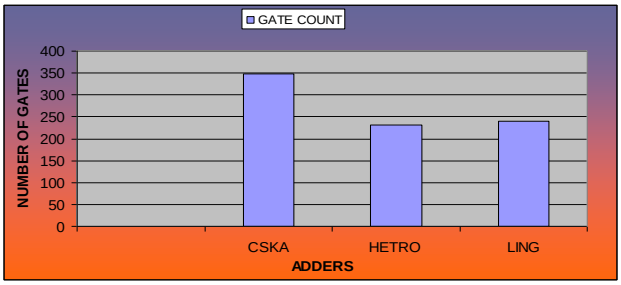

Figure 5: Hardware utilization comparison w.r.t. heterogeneous adder.

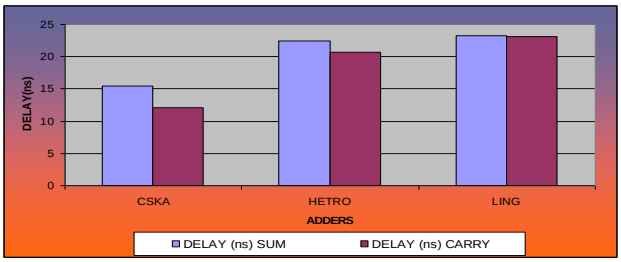

Figure 6: Speed comparison w.r.t. heterogeneous adder

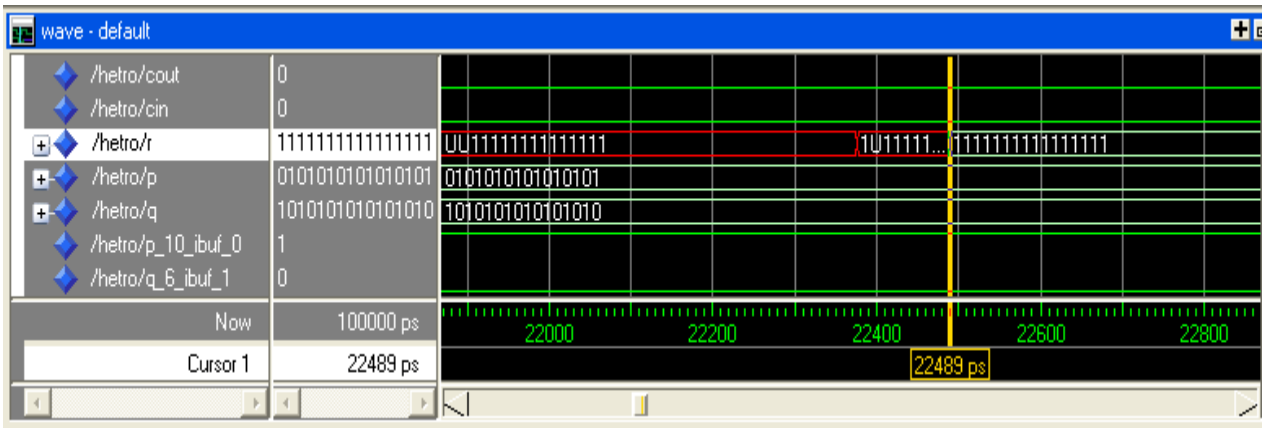

Figure 7: Simulation Waveform for Summation output of 16-bit Heterogeneous Adde

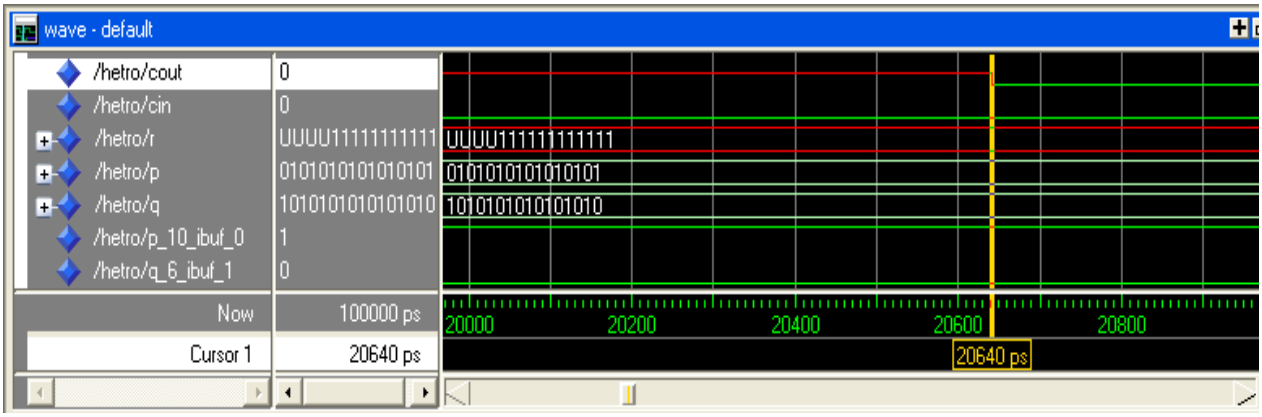

Figure 8: **Simulation Waveform for Carry output of 16-bit Heterogeneous Adder**

5. CONCLUSIONS AND FUTURE WORK

In this paper, we try to optimize adder design in term of hardware (Area) utilization and speed of operation. Based on the Synthesis and Simulation results for two 16-bit fast adders shown in table-1, it is observed that Ling adder design give better performance in terms of area utilization as compare to hybrid carry skip adder but at the cost of speed of operation. From table-1, we mark upper bound points for area utilization and delay time (gate count = 348, sum delay = 23.3ns and carry delay = 23.12ns). Therefore, to optimize adder design circuit, we concatenate two adders to form a single adder. The proposed adder design shown in fig.4 gives best performance in terms of hardware utilization (gate count = 231) as well as gives delay of operation less than the upper bound (sum delay = 22.48ns and carry delay = 20.64ns).

Similarly, this work can be extended for optimizing adder circuitfor another important constraint i.e. power consumption.

6. REFERENCES

[1] B. Parhami, 2000, Computer Arithmetic, Algorithm and Hardware Design, Oxford University Press, New York, pp. 91-119.

[2] Pong P. Chu "RTL Hardware Design Using VHDL: coding for Efficiency, Portability and Scalability" Wiley-IEEE Press, New Jercy, 2006

[3] Hasan Krad and Aws Yousif Al-Taie, "Performance Analysis of a 32-Bit Multiplier with a Carry-Look-Ahead Adder and a 32-bit Multiplier with a Ripple Adder using VHDL", Journal of Computer Science 4 (2008): 305-308.

[4] Asadi, P. and K. Navi "A novel high-speed 54-54-bit multiplier", Am. J. Applied Sci., 4 (9) (2007): 666-672.

[5] Z. Abid, H. El-Razouk and D.A. El-Dib, "Low power multipliers based on new hybrid full adders", Microelectronics Journal, Volume 39, Issue 12, 2008, 1509-1515.

[6] Nagendra, C.; Irwin, M.J.; Owens, R.M.,"Area-time-power tradeoffs in parallel adders", Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on Volume 43, Issue 10, (1996): 689 – 702.

[7] Wang, Y.; Pai, C.; Song, X., "The design of hybrid carry look-ahead/carry-select adders, Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on Volume 49, Issue 1, (Jan 2002) : 16-24.

[8] May Phyo Thwal, Khin Htay Kyi, and Kyaw Swar Soe, "Implementation of Adder-Subtracter design with VerilogHDL", International Journal of Electronics, Circuits and Systems Volume 2 number 3, 2008.

[9] Min Cha and Earl E. Swartzlander, Jr, "Modified Carry Skip Adder for reducing first block delay", Proc. 43rd IEEE Midwest Symp. on Circuits and Systems, Lansing MI, 2000: 346-348.

[10] Yu Shen Lin, "Delay Efficient 32-bit Carry-Skip Adder", Electronics, Circuits and Systems, 2006, ICECS '06, pp. 506-509.

[11] Behnam Amelifard, Farzan Fallah, Massoud Pedram, "Closing the gap between Carry Select Adder and Ripple Carry Adder: A new class of Low-power and High-performance Adders", Proceedings of the Sixth International Symposium on Quality Electronic Design (ISQED'05) , 2005.

[12] Jin-Fu Li, Jiunn-Der Yu, Yu-Jen Huang, "A Design Methodology for Hybrid Carry-Lookahead/Carry-Select Adders with Reconfigurability", 0-7803-8834-8/05/$20 ©2005 IEEE, 2005.

[13] Pak K. Chan, et al, Delay Optimization of Carry-Skip Adders and Block Carry-Lookahead Adders Using Multidimensional Dynamic Programming, IEEE Transactions on Computers, vol. 41, No. 8 (Aug. 1992): 920-93.

[14] Sarabdeep Singh and dilip Kumar, "Design of Area and Power Efficient Modified Carry Select Adder", International Journal of Computer Application (November 2011), vol. 33 – No.3