

TRACING OF DELAY ESTIMATION IN HETEROGENEOUS ADDER WITH FPGA

Alisha¹, Tilak Raj²

¹PG Student, ²Assistant Professor, ECE Department, S(PG)ITM Rewari, Haryana, India

I. INTRODUCTION

HETROGENEOUS ADDER

The architecture of a heterogeneous adder includes different types of adder implementations. 32-bit heterogeneous adder proposed in this work consists of four sub adders (SA), 8-bit carry look-ahead adder, 8-bit carry skip adder, 8-bit carry select adder and 8-bit ripple carry adder. Bit size selection for each sub-adder can be done on the basis of requirements (i.e. Area, Speed and Power constraints) of particular application where the design is to be implemented. For example, ripple carry adder cover small area and less power consumption but at the cost of large operation delay whereas carry skip adder gives high speed of operation but at the cost of large area. Therefore in order to optimize adder design as per requirement, 32-bit heterogeneous adder is designed as shown in figure. In past, the major challenge for VLSI designer is to reduce area of chip by using efficient optimization techniques. Then the next phase is to increase the speed of operation to achieve fast calculations like, in today's microprocessors millions of instructions are performed per second. Speed of operation is one of the major constraints in designing DSP processors. Now, as most of today's commercial electronic products are portable like Mobile, Laptops etc. That requires more battery backup. Therefore, lot of research is going on to reduce power consumption. Therefore, there are three performance parameters on which a VLSI designer has to optimize their design i.e., Area, Speed and Power. It is very difficult to achieve all constraints for particular design, therefore depending on demand or application some compromise between constraints has to be made with some constraints. Adders are commonly found in the critical path of many building blocks of microprocessors and digital signal processing chips [1], [2].

Adders are essential not only for addition, but also for subtraction, multiplication, and division. Addition is one of the fundamental arithmetic operations. A fast and accurate operation of a digital system is greatly influenced by the performance of the resident adders. The most important for measuring the quality of adder designs in the past were propagation delay, and area. Many different adder architectures have been proposed for speeding up binary addition over the literature survey like RCA, CLA, and CSLA etc [3], [4],[5],[6]. But different adder architectures have some drawbacks in terms of area or delay. As adder forms very important unit in various circuits and systems there is a need to design adder architecture with minimum area without affecting the speed of operation. In this paper various heterogeneous adder architectures are

proposed by using different homogeneous adder architectures. Later the designed adders are compared with each other in terms of area (number of LUTs) and delay (ns).

II. RELATED WORK

Adder is the essential component in any digital system and many variations are introduced in the carry generation schemes for area, speed and power trade-offs. Hybrid adders were developed in the past to provide area and speed trade-off by utilizing different schemes for sum and carry logic separately. For example in, carry look-ahead adder was used for carry generation and carry select adder for sum generation. Several homogeneous adders were reconfigured with their bit widths to achieve variable performance and power trade-offs. Architectures reported in have adder variants where larger bit system is partitioned in to smaller bits and reconfigured using additional bits. For example partitioned the 64 bit Carry select adder to perform as one 64-, two 32-, four 16-, and eight 8-bit adders. Similarly in a carry skip adder has been illustrated. Such adders provide the selection of the bit widths of the adders and improve efficiency of the design. An effort has been put in, to add the extra flexibility into the system where different adder variants of smaller bit widths are incorporated in the larger adder system to address the delay optimization under power constraints or power optimization under delay constraints. Such architectures are called as Heterogeneous adders. In this paper, we propose the low power heterogeneous adder architecture to provide power optimization with variable performance. Limitations of the state of the art reconfigurable architectures,

a) In the regular reconfigurable architectures, static/dedicated adder architectures are utilized and multiplexer selects the required adder variant. This requires more area and consumes more power to achieve variable performance and reconfigurability between the adder variants.

b) In the state of the art heterogeneous architecture, the static sub-adder blocks consumes more power while still providing good performance. In this paper, we address the above limitations: • Low power adder architectures o Complex cells were utilized to build the adder architecture, as they eliminate the interconnect delays between the gates and helps in reducing power. Elimination of inverters in the critical path reduces the switching power.

• The proposed concept is suitable for any bit widths and at any level of abstractions where the application needs different operating corners by providing the flavors of different adder variants in the same design.

• Heterogeneous adder architecture was incorporated in the adder block of digital filter.

Along the growth of multimedia application, the demand for high-performance and low-power digital signal processing (DSP) is getting higher and higher. The FIR digital filter is one of the most widely used essential devices. Some applications need the FIR filter to operate at high frequencies such as video processing, whereas some other applications request high throughput with a low-power circuit such as multiple-input-multiple-output systems used in cellular wireless communication. Furthermore, when narrow transition band characteristics are required, the much higher order in the FIR filter is unavoidable. Parallel processing in the digital FIR plays key role. The hardware implementation cost of the parallel processing increases due to increase in block size L , hence parallel processing technique loses its advantage to be employed in practice. Adders are most commonly used in various electronic applications e.g. Digital signal processing in which adders are used to perform various algorithms like FIR, IIR etc. The major challenge for VLSI designer is to reduce area, power and increase the speed of operation using efficient optimization techniques Speed of operation is one of the major constraints in designing DSP processors. Many parallel adders are studied where all the inputs are available before the start of the computation. The parallel adders which are selected for the project vary widely in their delay and speed characteristics. Since asynchronous systems are not very common, we concentrate on building synchronous adders for the paper. The adders studied are linear time ripple carry adder, Ling adder, square-root time carry skip adder and logarithmic time carry look-ahead adder. These adders have their own benefits and limitations with respect to performance parameters e.g. implementing Ripple carry adder utilizes less area but at the cost of large delay, whereas, carry look-ahead 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 project, new architecture is proposed that combine different types of adder to form a single heterogeneous adder to satisfy design constraints. The adders presented here are all modeled by using VHDL for 16-bit unsigned data.

III. CARRY SELECT ADDER

Carry Select Adders (CSAs) have been considered as a compromise solution between RCAs and CLAs time because they offer a good trade-off between the compact area of RCAs and the short delay of CLAs. The carry select adder comes in the category of conditional sum adder. Conditional sum adder works on some condition. Sum and carry are calculated by assuming input carry as 1 and 0 prior the input carry comes. When actual carry input arrives, the actual calculated values of sum and carry are selected using a multiplexer. The CSA is used in many systems to overcome the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum. But the CSA is not area efficient because it uses multiple pairs of Ripple Carry Adders (RCA) to generate partial sum and carry by considering carry input $c_{in} = 0$ and $c_{in} = 1$, then the multiplexers are used to get final sum and carry are used.

Heterogeneous Computing and the OpenCL Computing Language

In the high-performance computing field, heterogeneous computing systems are emerging to solve a wide range of scientific computing challenges. A standard CPU with an attached accelerator device, such as a graphic processor unit (GPU) or FPGA, can accelerate a wide range of functions including data search, image processing, financial, or seismic simulations. With these heterogeneous systems, programming standards have emerged to allow easier adaptation of algorithms from standard systems to accelerated heterogeneous systems.

OpenCL is a framework for writing programs that execute across heterogeneous platforms consisting of CPUs, GPUs, digital signal processors (DSPs), FPGAs, and other multi-core processors. The OpenCL framework includes a language based on standard ANSI C99 for programming these devices and application programming interfaces (APIs) to control the platform and execute programs on the compute devices. The Khronos Group (2), a non-profit organization, manages the standard. An advantage of OpenCL is portability of programs from one vendor's accelerator device to another. Several vendors, including Altera, provide compilers for OpenCL. To claim conformance to the OpenCL standard, the vendor's compiler must accurately compile and execute a suite of over 8,500 OpenCL programs. Altera, Intel, AMD, and Nvidia provide OpenCL conformant compilers.

Programming with FPGAs

Traditionally, hardware developers have designed and verified digital circuits on FPGAs at the register-transfer level (RTL) using hardware description languages (HDLs) such as Verilog HDL and VHDL. While these traditional methods are effective to ensure efficient use of the devices, they are impractical for implementing complex algorithms such as gene sequencing. In early 2012, Altera introduced the Altera SDK for OpenCL, a software development kit that allows use of the OpenCL programming language to program Altera's FPGA as computing accelerator devices. In late 2014 Xilinx Corporation, another leading FPGA vendor, announced they were also developing a compiler for OpenCL. Altera's SDK for OpenCL has been utilized for a wide array of algorithms in a variety of computing fields.

XILINX ISE v 14.5i is used as synthesis tool and FPGA-Spartan VI (XC3S250E) device is selected to get area report. Model Sim XE III 6.2g is used to get timing simulation.

The remaining parts of this paper are arranged as follows. Proposed adder architecture and advantage is briefed in chapter 1. Heterogeneous adder architecture and its application are discussed in chapter 1. Results are discussed in chapter 4. Chapter 5 gives the conclusion and references for the paper are provided in the last.

Adder Variants

CLA (Carry Look Ahead Adder) CLAs reduce the computation time, by adding two binary numbers faster. They work by creating Carry Propagator 'P' and Carry

Generator 'G' signals. Carry propagator propagates to the next level whereas the carry generator generates the output carry regardless of input carry. The Architecture of CLA is shown in Fig.

The corresponding Boolean expressions for 'P' and 'G' are given are used to construct a CLA.

$$P_i = A_i \oplus B_i$$

$$G_i = A_i \cdot B_i$$

The output of sum and carry can be expressed as

$$Sum_i = P_i \oplus C_i \quad (3)$$

$$C_{i+1} = G_i + (P_i \cdot C_i); \text{ where } i = 0, 1 \dots n-1$$

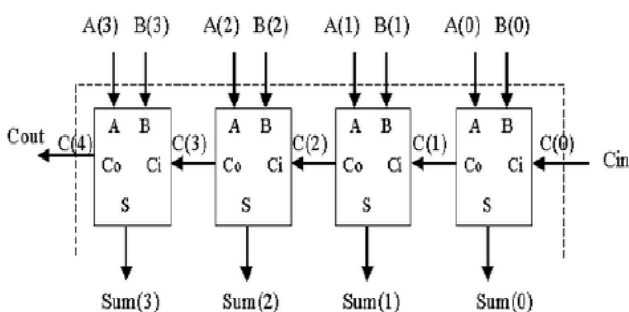
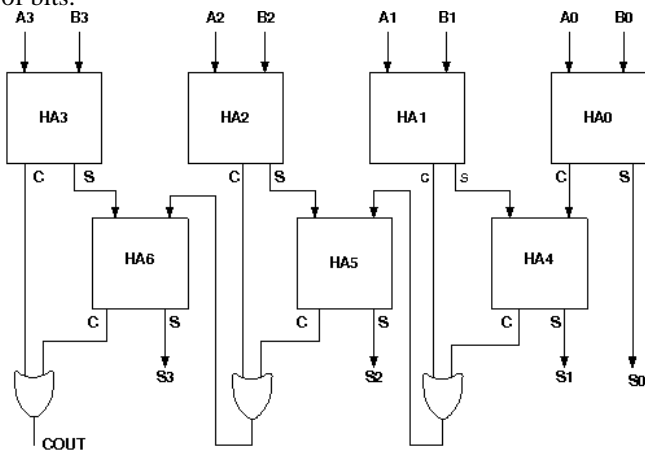
RCA (Ripple Carry Adder)

In RCA, Full Adders are cascaded in series, where the carry out from previous stage is connected to carry input of the next stage. Full Adder forms the basic building block of RCA, and it has three inputs say 'A', 'B', 'Cin' and two outputs say "Sum" and "Cout". The critical path of the RCA passes from carry-in to the carry-out along the majority gates. Carry-out expression is given by equation.

$$C_{out} = A \cdot B + (A + B) \cdot C_{in} \quad (5)$$

$$\text{Equation can be simplified to } C_{out} = A \cdot B + (A \oplus B) \cdot C_{in} = G + P \cdot C_{in} \quad (6)$$

Where P and G are the propagate and generate carry logics, Fig. shows the 4 bit ripple carry adder, and from equation, the critical path of RCA consists of chain of AND-OR gates. The delay of adder increases linearly with increase in number of bits.



Proposed Adder Architecture & Advantage The proposed adder architectures are built using complex cells and possible inverters are eliminated in the critical path of the architecture. Complex cells like AND-AND-OR and AND-OR are used. These complex cells have higher transistor stacks than the

regular cells. As per the known fact that the higher transistor stack will have more stack resistance and helps in reducing the leakage power of the devices. It also reduces interconnects and the delay associated between the gates, which in turn reduces any associated glitches. Inverters in the critical path are eliminated by utilizing inverter free equivalent gates. In the regular adder architectures inverters are used in XOR gate implementation.

Advantages:

- Reduced Leakage power due to the higher transistor stacked complex cells
- Reduced dynamic power due to minimal interconnects
- Lesser area due to merging of smaller gates into complex cells

Table 16-bit Ling, Carry and heterogeneous adder design comparison for Area and Delay

RESULT

SIMULATION RESULTS

Table: 16-bit Ling, Carry and heterogeneous adder design comparison for Area and Delay

ADDER	GATE COUNT	DELAY(ns)	
		SUM	CARRY
CSKA	345	14.9	11.8
LING	235	22.7	23.12
HETRO	200	2.177	2.133

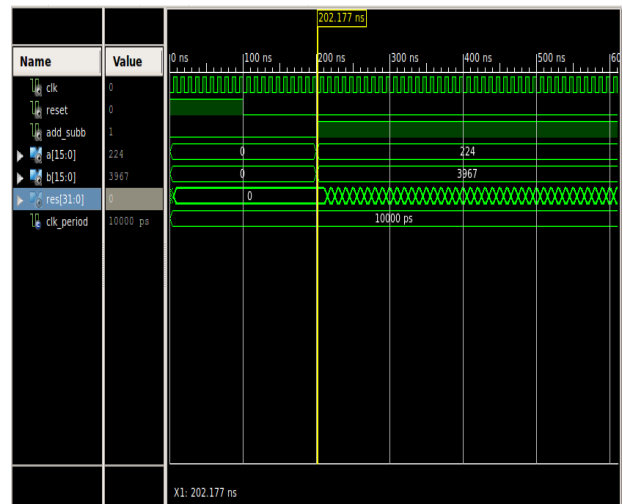


Fig. Waveform for 16 bit heterogeneous adder

IV. CONCLUSIONS AND FUTURE WORK

In this thesis, 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, 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 = 345, sum delay = 14.93ns and carry delay = 11.18ns). Therefore, to optimize

adder design circuit, we concatenate two adders to form a single adder. The proposed adder design shown in fig. gives best performance in terms of hardware utilization (gate count = 200) as well as gives delay of operation less than the upper bound (sum delay = 2.177ns and carry delay = 2.133ns).

REFERENCES

- [1] Arvind Kumar, A.K Goyal, "Study of Various Full Adders using Tanner EDA Tool," *International Journal of Computer Science and Technology*, Volume 3, Issue1,
- [2] Sutherland, B. Sproull, D. Harris, *Logical effort designing fast cmos circuit*, Morgan Kaufmann publisher, 1st Edition,1999.
- [3] Neha Agarwal, Satyajit Anand, "Study and performance comparison of VLSI adders using logical effort delay model," *IJATER*, Volume 2, Issue 6, Nov. 2012.
- [4] S.C.Tiwari, Kumar Singh, Maneesha gupta, "Design and Development of Logical Effort Based Automated transistor Width Optimization Methodology," *World Applied Science Journal IDOSI Publication*, Volume 16, pp. 29-36, 2012.
- [5] Hoang Q. Dao and V.G. Oklobdzija, "Performance Comparison of VLSI Adders Using Logical Effort," *PATMOS*, pp. 25-34, 2002.
- [6] S. Anand and P. K. Ghosh, "Optimization and comparison of 4-stage inverter, 2-i/p Nand, 2-i/p Nor Gate by using Logical Effort," *AIP Conference Proceedings*, Volume 1324, pp. 356-359, Nov.2010.
- [7] N. Weste, D. Harris, *CMOS VLSI design: a circuits and systems perspective*, 3rd Edition, Pearson education, 2005.
- [8] Jan M. Rabaey, *Digital integrated circuits: a design perspective*, Pearson Education, 2nd Edition, 2005.
- [9] X. Yu, V. Oklobdzija, "Application of Logical Effort on Design of Arithmetic Blocks in VLSI CMOS Technology," *35th Asilomar Conference on Signals, Systems and Computers*, Nov. 2001.
- [10] H. Dao and V. G. Oklobdzija, "Application of Logical Effort on Delay Analysis of 64 - bit Static Carry Look- ahead Adder, " *35th Asilomar Conference on Signals, Systems and Computers*, 2001.
- [11] R. P Singh and A. Chaturvedi, "VLSI Implementation of Heterogeneous Adder for performance Optimization," *International Journal of Computer Applications (0975- 8887)*, Volume 51– No.7, Aug. 2012.