Design of Efficient 16-Bit Parallel Prefix Ladner-Fischer Adder

[1]S.Baba Fariddin
M.Tech., E.C.E Department
Gudlavalleru Engineering College
Gudlavalleru, A.P, India.

[2]E.Vargil Vijay
Assistant Professor, E.C.E Department
Gudlavalleru Engineering College
Gudlavalleru, A.P, India.

ABSTRACT
A parallel-prefix adder gives the best performance in VLSI design. However, performance of Ladner-Fischer adder through black cell takes huge memory. So, gray cell can be replaced instead of black cell which gives the Efficiency in Ladner-Fischer Adder. The proposed system consists of three stages of operations they are pre-processing stage, carry generation stage, post-processing stage. The pre-processing stage focuses on propagate and generate, carry generation stage focuses on carry generation and post-processing stage focuses on final result. In ripple carry adder each bit of addition operation is waited for the previous bit addition operation. In efficient Ladner - Fischer adder, addition operation does not wait for previous bit addition operation and modification is done at gate level to improve the speed and to decreases the memory used.

General Terms
Ripple carry adder, Efficient Ladner–Fischer adder, Black cell, Gray cell

Keywords
Efficient Ladner-Fischer adder-(ELF).

1. INTRODUCTION
Addition operation is the main operation in digital signal processing and control systems. The fast and accuracy of a processor or system depends on the adder performance. In general purpose processors and DSP processors the addition operation addresses are taken from simple ripple carry adder [1].

Ripple carry adder is used for the addition operation i.e., if N-bits addition operation is performed by the N-bit full adder. In ripple carry adder each bit full adder operation consists of sum and carry, that carry will be given to next bit full adder operation, that process is continuous till the N^{th} bit operation. The N-1^{th} bit full adder operation carry will be given to the N^{th} bit full adder operation present in the ripple carry adder.

For 16-bit ripple carry adder, the first bit carry is given to second bit full adder, second bit carry is given to the third bit full adder, similarly the operation is continue till fifteenth bit carry is given to sixteenth bit full adder. The addition operation is performed from least significant bit to most significant bit in ripple carry adder. Configuration logic and routing resources in Field Programmable Gate Array.

2. LADNER-FISCHER ADDER
The Ladner-Fischer is the parallel prefix adder used to perform the addition operation. It is looking like tree structure to perform the arithmetic operation [4]. Ladner-Fischer adder is used for high performance addition operation. The Ladner-Fischer adder consists of black cells and gray cells [3]. Each black cell consists of two AND gates and one OR gate [2]. Multiplexer is combinational circuit which consists of multiple inputs and a single output. Each gray cell consists of only one AND gate. P_i denotes propagate and it consists of only one AND gate [5] given in equation 1. G_i denotes generate and it consists of one AND gate and OR gate [6] given in equation 2.

$$P_i = B_i \text{ AND } B_{i-1} - - - - - (1)$$

$$G_i = A_i \text{ OR } [B_i \text{ AND } A_{i-1}] - - (2)$$

G_i denotes generate and it consists of one AND gate and OR gate given in equation 3[8].

$$G_{i-2} = A_{i-2} \text{ OR } [B_{i-2} \text{ AND } A_{i-1}] \ - - (3)$$

3. PROPOSED LADNER-FISCHER ADDER
The proposed Ladner-Fischer adder is flexible to speed up the binary addition and the structure looks like tree structure for the high performance of arithmetic operations.

In ripple carry adders each bit wait for the last bit operation. In parallel prefix adders instead of waiting for the carry propagation of the first addition, the idea here is to overlap the carry propagation of the first addition with the computation in the second addition, and so forth, since repetitive additions will be performed by a multioperand adder.

Research on binary operation elements and motivation gives development of devices. Field programmable gate arrays [FPGA's] are most popular in recent years because they improve the speed of microprocessor based applications like mobile DSP and telecommunication. The construction of efficient Ladner-Fischer adder consists of three stages. They are pre-processing stage, carry generation stage, post-processing stage.

3.1 Pre-Processing Stage
In the pre-processing stage, generate and propagate are from each pair of inputs. The propagate gives "XOR" operation of input bits and generates gives "AND" operation of input bits [7]. The propagate (P_i) and generate (G_i) are shown in below equations 4 & 5.

$$P_i = A_i \text{ XOR } B_i - - - -(4)$$

$$G_i = A_i \text{ AND } B_i ----(5)$$

3.2 Carry Generation Stage

In this stage, carry is generated for each bit and this is called as carry generate (C_g). The carry propagate and carry generate is generated for the further operation but final cell present in the each bit operation gives carry. The last bit carry will help to produce sum of the next bit simultaneously till the last bit. The carry generate and carry propagate are given in below equations 6 & 7.

$$C_p = P_1 \text{ AND } P_0 ------(6)$$

$$C_g = G_1 \text{ OR } (P_1 \text{ AND } G_0) --(7)$$

The above carry propagate C_p and carry generation C_g in equations 6&7 is black cell and the below shown carry generation in equation 8 is gray cell. The carry propagate is generated for the further operation but final cell present in the each bit operation gives carry. The last bit carry will help to produce sum of the next bit simultaneously till the last bit. This carry is used for the next bit sum operation, the carry generate is given in below equations 8.

$$C_g = G_1 \text{OR } (P_1 \text{ AND } G_0) --(8)$$

3.3 Post-processing stage

It is the final stage of an efficient Ladner-Fischer adder, the carry of a first bit is XORed with the next bit of propagates then the output is given as sum and it is shown in equation 9.

$$S_i = P_i \text{ AND } C_{i-1} ----- (9)$$

It is used for two sixteen bit addition operations and each bit carry is undergoes post-processing stage with propagate, gives the final sum.

The first input bits goes under pre-processing stage and it will produce propagate and generate. These propagates and generates undergoes carry generation stage produces carry generates and carry propagates, these undergoes post-processing stage and gives final sum. The step by step process of efficient Ladner-Fischer adder is shown in Fig 1.

The Efficient Ladner-Fischer adder structure is looking like tree structure for the high performance of arithmetic operations and it is the fastest adder which focuses on gate level logic. It designs with less number of gates. So, it decreases the delay and memory used in this architecture.

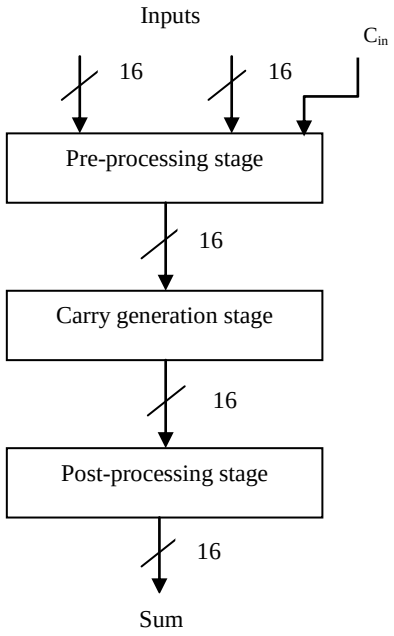

Fig 1: Block Diagram

In Efficient Ladner-Fischer adder, black cell operates three gates and gray cell operates two gates. The gray cell reduces the delay and memory because it operates only two gates. The proposed adder is design with the both black and gray cells. By using gray cell operations at the last stage of proposed adder gives a enormous dropping delay and memory used.

The proposed adder is shown in fig 2 which improves the speed and decreases the memory for the operation of 8-bit addition. The input bits A_i and B_i concentrates on generate and propagate by XOR and AND operations. These propagates and generates undergoes the operations of black cell and gray cell and gives the carry C_i. That carry is XORed with the propagate of next bit, that gives sum.

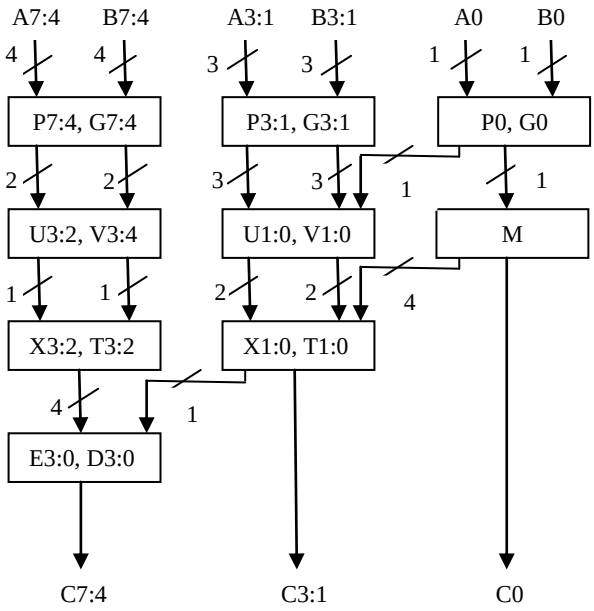

Fig 2: 8-Bit Efficient Ladner-Fischer Adder

The architecture of Efficient Ladner-Fischer adder gives the less delay and less memory for the operation of 16-bit addition. The properties of the operations are evaluated in parallel and accept the trees to overlap which leads to parallelization.

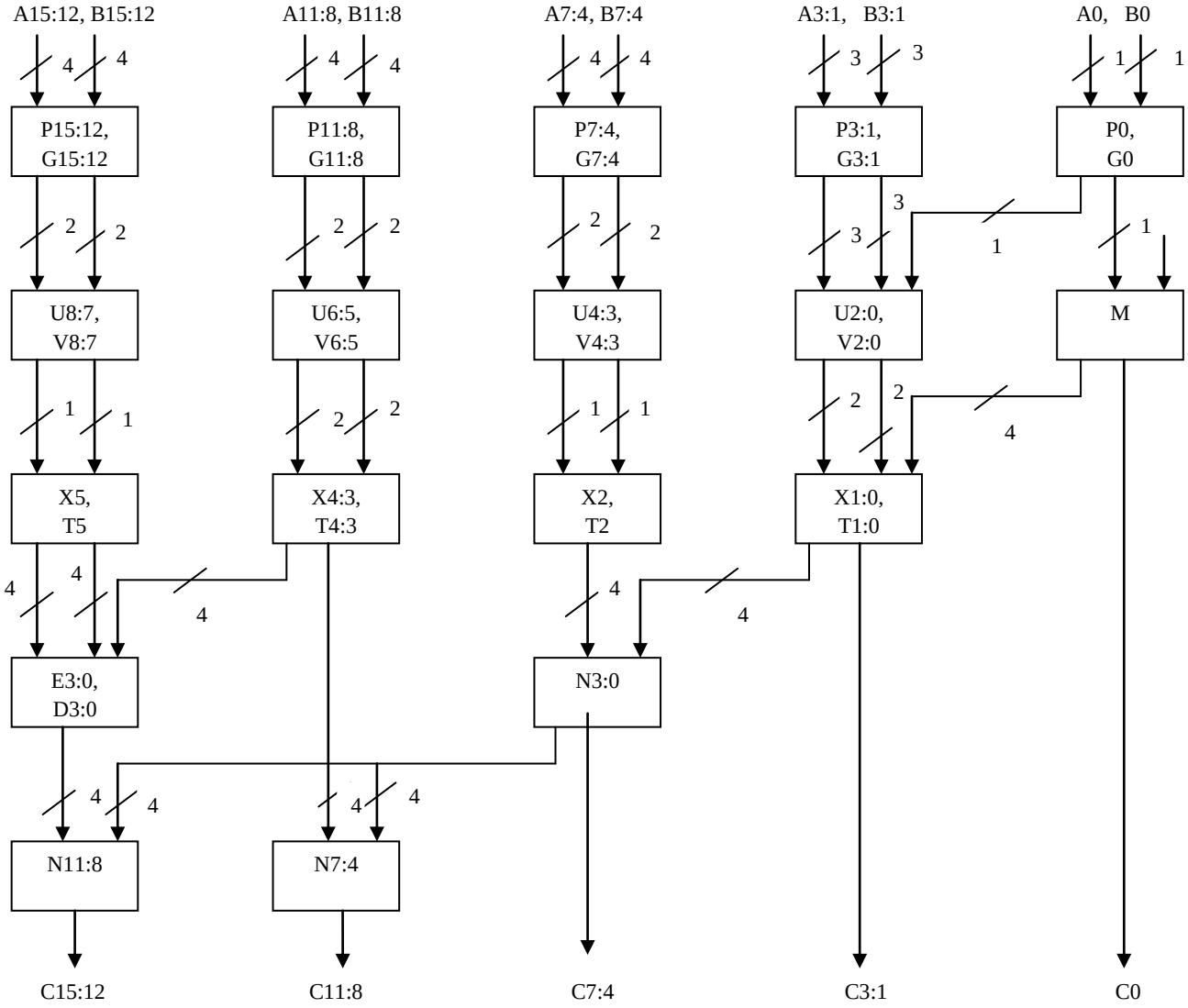

Fig 3: 16-Bit Efficient Ladner-Fischer Adder

The architecture of 16-bit Efficient Ladner-Fischer adder is shown in Fig 3. The logical circuit is using multiple adders to find the sum of N-bit numbers. Each addition operation has a carry input (C_{in}) which is the previous bit carry output (C_{out}).

The Efficient Ladner-Fischer Adder design takes less number of gates. Generally each black cell consists of two AND gates, one OR gate and gray cell consists of one AND gate, one OR gate. The last stage design with the gate level logic with the gray cell reduces delay and memory.

Research on binary addition motivates gives development of devices. Many parallel prefix networks describe the literature of addition operation.

The parallel prefix adders are Brent-kung, Kogge-stone, ladner-Fischer, Sklansky, etc. The fast and accurate performance of an adder is used in the very large scale integrated circuits design and digital signal processors.

4. SIMULATION RESULTS

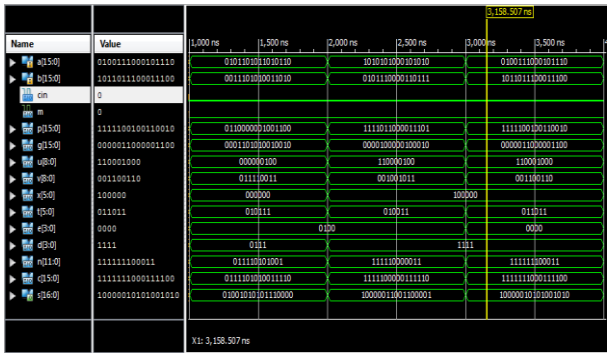

Fig 4: 16-Bit Efficient Ladner-Fischer Adder Simulation Waveform

The Efficient Ladner-Fischer adder is designed on VHDL (very high speed integration hardware description language). Xilinx project navigator 12.1 is used for synthesis. Simulation results are shown in Fig 4.

The design of adders is done on VHDL. The memory and delay performance Efficient Ladner-Fischer adder (ELF) is shown in Table 1.

Table 1. Delay and memory used in ELF

Adder	Delay(ns)	Memory (kb)
8-bit Efficient Ladner-Fischer adder	11.2	186228
16-bit Efficient Ladner-Fischer adder	12.2	188788

5. CONCLUSION

In this paper, a new approach to design an efficient Ladner-Fischer adder concentrates on gate levels to improve the speed and decreases the memory. It is like tree structure and cells in the carry generation stage are decreased to speed up the binary addition. The proposed adder addition operation offers great advantage in reducing delay.

6. ACKNOWLEDGMENT

I S.BABA FARIDDIN would like to thank Mr. E.VARGIL VIJAY, assistant professor E.C.E Dept, Gudlavalleru engineering college for giving me the technical ideas and guiding me throughout the project.

7. REFERENCES

[1] Pakkiraiah. Chakali, madhu Kumar. Patnala "Design of high speed Ladner - Fischer based carry select adder" IJSCE march 2013

[2] Haridimos T.Vergos, Member, IEEE and Giorgos Dimitrakopoulos, Member, IEEE," On modulo 2^n+1 adder design" IEEE Trans on computers, vol.61, no.2, Feb 2012

[3] David h, k hoe, Chris Martinez and Sri Jyothsna vundavalli "Design and characterization of parallel prefix adders using FPGAs", Pages.168-172, march2011 IEEE.

[4] K. Vitoroulis and A.J. Al-Khalili, "performance of parallel prefix adders implemented with FPGA technology," IEEE Northeast Workshop on circuits and systems, pp.498-501, Aug 2007.

[5] Giorgos Dimitrakopoulos *and* Dimitris Nikolos, *"*High Speed Parallel Prefix ...*Ling adders,"* IEEE Transactions *on* Computers, vol.54, no.2, February 2005

[6] S. Knowles," A family of adders,"proc.15thsymp. Comp. Arith, pp. 277-281, June 2001.

[7] R. Brent and H. Kung, "A regular layout for parallel adders," IEEE Trans. Computers, vol.C-31, no. 3, pp. 260-264, March 1982.

[8] R.E. Ladner and M.J. Fischer, "Parallel Prefix Computation," J. ACM, vol. 27, no. 4, pages 831-838, Oct. 1980.