AN EFFICIENT BASE-4 LEADING ZERO DETECTOR DESIGN

Alvernon Walker[1] and Evelyn R. Sowells-Boone[2]

[1]Dept. of Engineering and Aviation Sciences, UMSE, Prince Anne, USA
[2]Dept. of Computer Systems Technology,
North Carolina A&T State University, GSO, USA.

ABSTRACT

A base-4 leading zero detector (LZD) design is proposed in this paper. The design is similar to the approach originally proposed by V.G. Oklobdzija with a different technique. The circuit modules used in the base-4 LZD approach are designed and several N-bit LZD circuits are implemented with a standard-cell realization in the Taiwan Semiconductor Manufacturing Company (TSMC) 0.65um CMOS process. The performance and layout area of the base-4 LZD realization is compared for implementations that contain only 4-to-1 and 2-to-1 multiplexers.

KEYWORDS

Leading Zero Detector, Multiplexer, Floating-point Unit

1. INTRODUCTION

A little-known fact is that floating point arithmetic is an essential component in computer systems for several reasons. Almost every computer language has floating point data type and accelerators. Compilers and operating systems are capable of processing information in the floating-point format. Even more importantly, is how essential the floating-point unit is to high performance computing (HPC), mobile applications and embedded systems. Computer system's performance is measured in Floating Point Operations per Second; more commonly known as FLOPs.

The overall performance of HPC system or any other computing system is greatly affected by the Floating-Point Unit (FPU) design; thus, the architecture can affect overall performance and power dissipation. Within the floating-point unit are several components: Adder/Subtractor, Multiplier and Divider. As their names suggest, they each have a specific computational role. However, the Adder is the single most commonly used component in this unit. According to data Pappalardo et al in Application-oriented analysis of power/precision tradeoff, signal processing algorithms require on average, 40% multiplication and 60% addition operations; once again, reinforcing the importance of FPU.

In this paper we will examine a one of the most power and time-consuming components of a FPU, the leading zero detector (LZD).A LZD circuit is a combinational logic block that determines the number of leading zeros in the primary input word. It is a critical block in floating-

DOI : 10.5121/eeij.2018.5101

point units (FPU) because it is required for the normalization step in the IEEE-754 floating-point standard. It computes the left shift distance and the exponent offset that is used to remove the significand leading zeros and that is required to adjust the exponent of the initial floating-point number in the normalization process. Figure 1 shows the forth stage of the floating-point addition. This stage contains the LZD which processes the 24-bit from the mantissa to send to the 27-bit left barrel shifter. This LZD block is complex and slow, in general, because its output is a function of all the input bits which can contain from 24-bits (the number bits in the single-precision floating-point number significand) to 113-bits (the number of bits in the quadruple-precision floating-point numbers significand) [1,2,3]. A novel modular base-4 LZD circuit is proposed in this paper. The propagation delay, area and complexity of a LZD block realized with this approach is less than an implementation designed with logic synthesis because the maximum fan-in in the proposed technique is not greater than 4 [1,2].

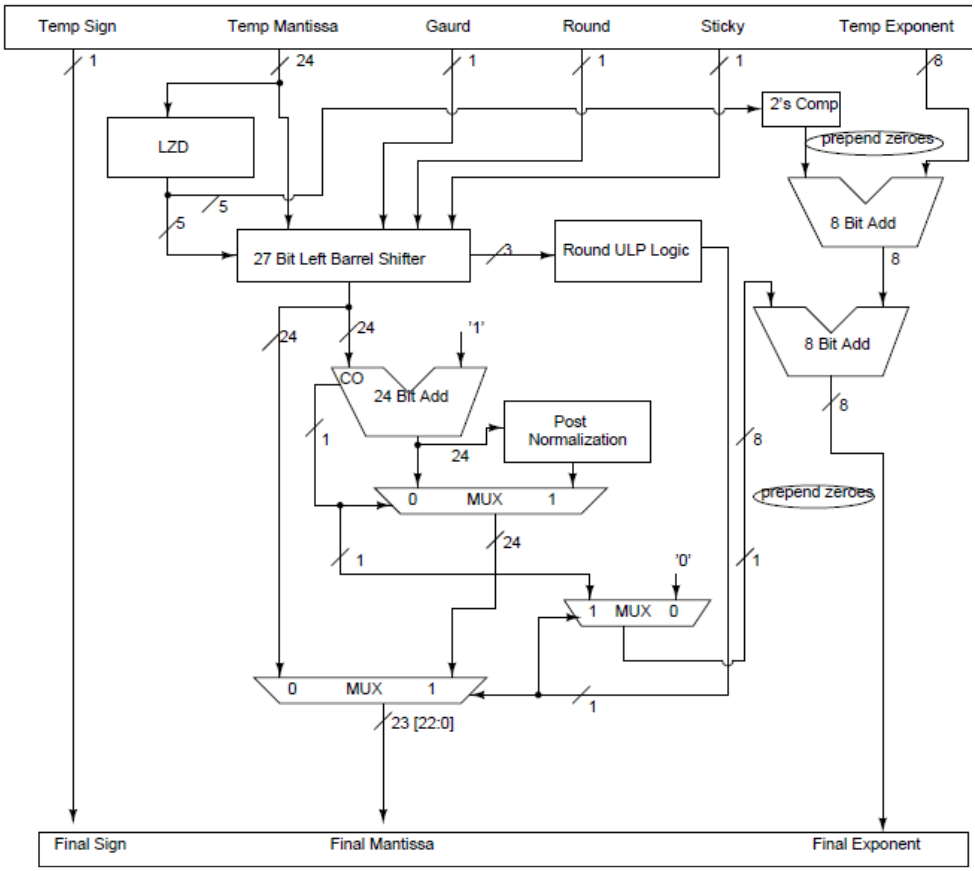

Figure 1: Stage 4 of 32-Bit Floating-Point Adder with LZD

A base-2 approach was proposed by Oklobdzjia in [1,2] with logarithmic complexity for LZD circuit design. The approach produces a binary tree structure with $\lceil log_2(N) \rceil$ levels of 2-to-1 multiplexers and 2 input OR gates, where N is the number of input bits [1,2]. The propagation delay of the structure is a function of the number of logic levels therefore the propagation of a 23-bit LZD realizations is the same as a 32-bit implementation. Some designers have used a leading ones predictor (LOP) instead which predicts the leading number of zeros in the addition result.

However, this design requires larger area because of the pre-encoding stage. In terms of performance area analysis, the LZD offers better results.

The gate-level realization and circuit simulation results are presented for 16 and 64-bit word lengths in this paper. The implementation of the proposed LZD approach is presented in section 2 of this paper. The circuits in this paper are implemented in the TSMC 0.65μm process. Section 3 contains the circuit simulation results. It also includes the analysis of the proposed technique. The conclusion is section 4.

2. IMPLEMENTATION

Our goal here is to design a module that can be used to design for an arbitrary number of input bits. We will use a novel 4-bit LZD and 2-t0-1 Multiplexers to realize the N-nit LZD circuit. The focus here will be on the design of the 4-bit LZD module. The proposed module will be a combinational logic block that processes 4-bits of the input X and produce a 2-bit output Y. The truth table of the 4-bit module used to implement the proposed LZD circuit is shown in Table 1. The module has the following two outputs: (1) a two-bit number Y that equals the number of leading zeros in the binary number at the primary input and (2) a zero flag.

Table 1: Truth Table of the LZD 4-Bit Module

Module 4-Bit Input $(X_0, X_1), (X_2, X_3)$	Leading Zero Position	Module 2-Bit Output **Y**	Module Zero Flag Output **Z**
1011	0	00	0
0100	1	01	0
0011	2	10	0
0001	3	11	0
0000	X	00	1

Bit **Z** that equals 1 when all input bits are zero otherwise it is 0. The output ***Y*** is invalid when ***Z*** is 1. The input pairs $(X_0, X_1)$and(X_2, X_3) represent the two base-4 LZD module inputs. The minimized Boolean output equations for the 4-Bit LZD module after mapping to a staticCMOS realization are shown in equation (1). The CMOS gate-level schematic of the module is shown in Fig. 2. The simplified Boolean logic used to realize the LZD module output is shown in the equation (1) below.

$$\begin{aligned} Y_0 &= \overline{\overline{X_0}\cdot X_1 + \overline{X_0}\cdot\overline{X_2}} = \overline{\overline{X_0}\cdot\left(X_1 + \overline{X_2}\right)} = \overline{\left(X_0 + \overline{X_1}\cdot X_2\right)} \\ Y_1 &= \overline{\overline{X_0}\cdot\overline{X_1}} = \overline{\left(X_0 + X_1\right)} \\ Z &= \overline{\overline{X_0}\cdot\overline{X_1}\cdot\overline{X_2}\cdot\overline{X_3}} = \overline{X_0 + X_1 + X_2 + X_3} \end{aligned} \quad (1)$$

Logical effort was used to size the transistors in each of these gates. The CMOS gate-level realization of the 4-bit module is shown in figure 2. Input $\boldsymbol{X_0}$ is the MSB of the circuit. The four input NOR gate in equation (1) was replaced with the two-input NOR and two-input NAND gate tree shown in Fig. 1 to reduce module area and increase its performance. The module has both a *Z* and *Z* bar (i.e. *ZN*) output because they're both required to implement an *N*-bit leading zero detector. The area of the module physical design is 87λx $72\lambda$where λ is the minimum feature size of a CMOS process.

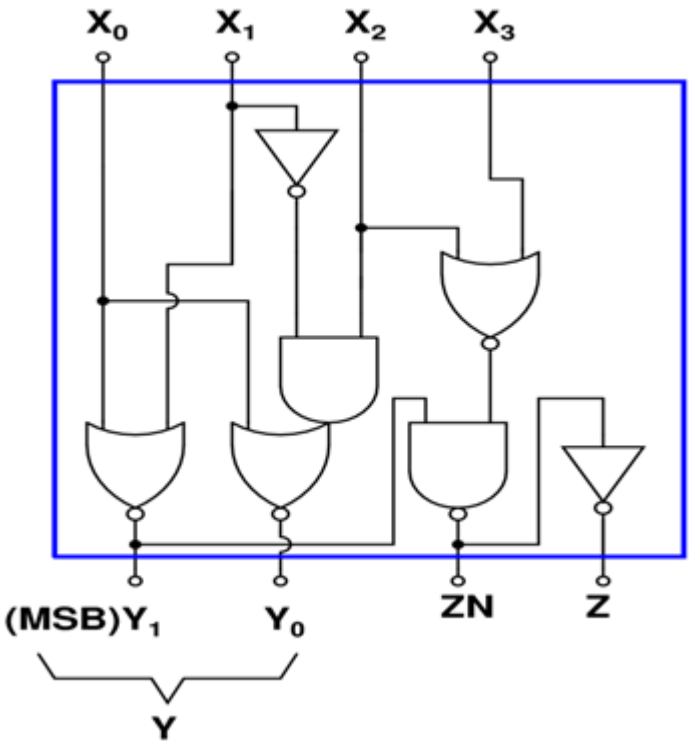

Figure 2: 4-Bit LZD Module

The figure below shows the dataflow of a 16-bit LZD component comprised of five 4-bit LZD modules. The 16-bit LZD circuit shown in Fig. 3 is based on the proposed approach and the 4-bit LZD module developed here. The multiplexers used in this design are based on the circuit in [4]. The critical path of the design is shown in red. The critical path includes the multiplexer select inputs, of both multiplexer realizations, because they form the largest capacitive load in the circuit. This is due to the increasing input word length of the multiplexers at each level of the multiplexer tree in the LZD circuit.

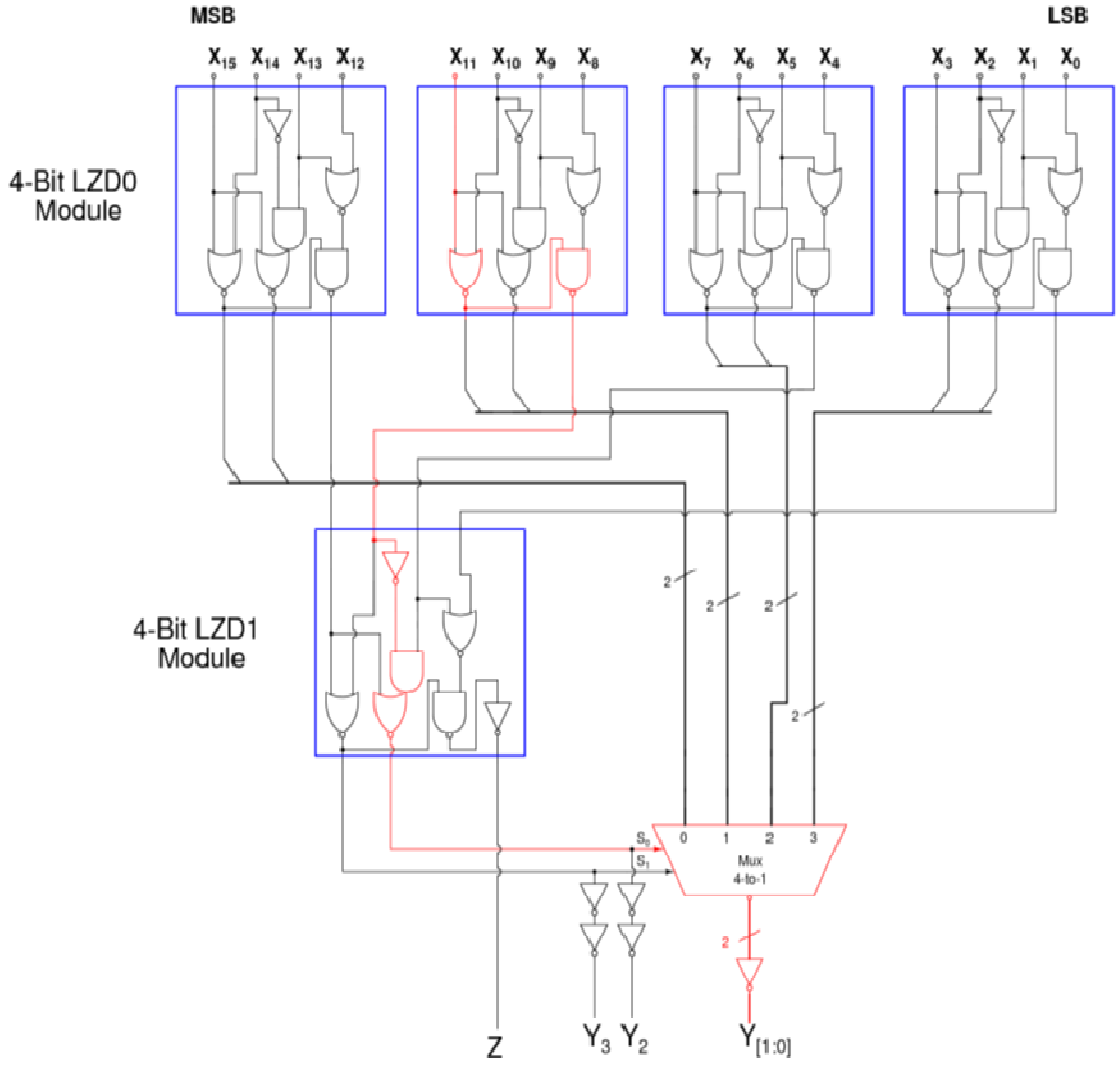

Figure 3: The Proposed Leading Zero Detection Circuit

3. RESULTS

The proposed base-4 LZD design approach is implemented with two types of 4-bit LZD modules to reduce critical path propagation delay and layout area. The first module type does not include output ***Z*** or the inverter associated with that output. This module henceforth will be called the 4-bit LZD0 module. The second 4-bit module only has the two-bit output ***Y*** and the single-bit output ***Z***. It is a called a 4-bit LZD1 module. The 4-to-1 multiplexers in the proposed base-4 LZD design approach are implemented with a single multiplexer that fits this specification. Designs based on both of these multiplexer realization approaches are examined in this paper. A 16-bit and 64-bit standard-cell LZD realization is implemented both with only 4-to-1or 2-to-1 multiplexers. A circuit simulation is done with SPICE for each circuit extracted from the layouts of the LZD circuits. The following input vectors are applied in the simulation to determine the propagation delay along the critical path of the 16-bit LZD circuit: 8000 and 0800. The estimated propagation delay of each circuit is shown in table 2.

Table 2: Proposed LZD Approach Circuit Features

LZD Word Length	Multiplexer Type	Layout Area	Critical Path Propagation Delay
16-Bit	2-to-1	$379\lambda \times 223\lambda$	17.046 *ns*
16-bit	4-to-1	$329\lambda \times 315\lambda$	0.5275 *ns*
64-bit	2-to-1	$1468\lambda \times 374\ \lambda$	
64-bit	4-to-1	$1268\lambda \times 436\lambda$	

4. CONCLUSION

The number of levels of 4-bit LZD modules in a *N*-bit LZD circuit realized with the proposed approach is $log_2 N$. The LZD design is hierarchical. The circuit structure can be realized with 2-to-1 or 4-to-1 multiplexers. LZD implementations that are realized with 4-to-1 multiplexers are faster than designs implemented with 2-to-1 multiplexers. However, the layout area of implementations realized with 2-to-1 multiplexers is less than LZD circuits designed with 4-to-1 multiplexers. The fan-out associated with the multi-bit multiplexer tree in the LZD hierarchy limit the performance of the proposed LZD implementation approach.

REFERENCES

[1] V.G.Oklobdzija, “An Implementation Algorithm And Design Of A Novel Leading Zero Detector Circuit,” Signals, Systems And Computers, 1992. 1992 Conference Recordof The Twenty-Sixth Asilomar Conference On, Vol., No., Pp.391-395 Vol.1, 26-28 Oct.1992. Doi: 10.1109/ACSSC.1992.269243

[2] V.G. Oklobdzija, "An Algorithmic And Novel Design Of A Leading Zero Detector Circuit: Comparison With Logic Synthesis," Very Large Scale Integration (VLSI) Systems, Ieeetransactions On, Vol.2, No.1, Pp.124-128, Mar 1994.Doi: 10.1109/92.273153

[3] "IEEE Standard For Floating-Point Arithmetic," IEEE Std 754-200, Vol., No., Pp.1-58, Aug. 29 2008 DOI: 10.1109/IEEESTD.2008.4610935

[4] I, Sutherland, B. Sproull, And. D. Harris, “Logical Effort: Designing Fast CMOS Circuits,” (Morgankaufmann Publishers, San Francisco, CA, 1999).

AUTHORS

Dr. Alvernon Walker is Chairman of the Engineering and Aviation Science Department at University of Maryland Eastern Shore. His research includes Asynchronous and Selftimed Digital System Design, Low Power High Performance Digital Systems Design.

Dr. Evelyn R. Sowells-Boone is an Assistant Professor in the Computer Systems Technology department at North Carolina A&T State University's College of Science and Technology. Prior to joining the School of Technology faculty, she held positions at U.S. Department of Energy, N.C. A&T's Division of Research and College of Engineering. Dr. Sowells earned a Ph.D. in Electrical Engineering from North Carolina A&T State University's College of Engineering. She also holds a M.S. and B.S in Computer Science with a concentration in software engineering. Her primary research interests are in the areas of low-power digital systems design, self-timed digital system design and STEM education.