Satish Paidi, Rohit Sreerama, K.Neelima / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 2,Mar-Apr 2012, pp.1103-1105

A Novel High Speed Leading Zero Counter For Floating Point Units

Satish Paidi*, Rohit Sreerama*, K.Neelima**
*(M.Tech VLSI, Sree Vidyanikethan Engineering College, Tirupathi,)
** (Assistant Professor, Sree Vidyanikethan Engineering College, Tirupathi,)

ABSTRACT: - This paper mainly focuses on leading zero counting. In this paper a novel leading zero counter is proposed. New Boolean expressions are derived for the new leading zero counter. We can explore various designs of leading zero counting unit by using the proposed approach. When compared with the known architectures the new circuits can be implemented effectively in both static and dynamic logic and also requires less energy per operation. The proposed leading zero counter can also be integrated with a leading zero anticipation logic for better and fast results. Finally, a simple and effective Leading Zero Counter was proposed and the occupancy rate and the delay characteristics are investigated.

Keywords - Floating Point Unit, Normalization, Leading Zero Counter (LZC), Priority Encoder.

I. INTRODUCTION

The speed and accuracy of micro processors developed rapidly during last decade. We have to consider optimizing both the delay and energy consumption of the modern microprocessors, in such a scenario the floating point units [3]-[5] play a dominant role. The main operation of the floating point datapaths is the normalization. The output of the normalization will follow the IEEE-754 standard format [2] i.e., 1.xxxxx.., x ϵ (0,1). The normalization process is done by involving the leading zero counting or detection unit. The problem of normalization [2] of the result involves counting the number of leading zeros in the result and then shifting the result in accordance to the outcome of the leading zero counter unit.

To update the exponent part correctly the derived leading zero counter is also required. Beside floating point datapaths the leading zero counter are also useful in many cases. Almost all instruction sets [5] of contemporary microprocessors include a count leading zeros (CLZ) instruction for fixed-point operands [3]-[5]. Also the leading zero counters are utilized in many hardware based function evaluation algorithms to speed up their computation. The leading zero counter can be used for some variable length code decoding architectures which are used in data compression.

This paper proposes a new leading zero counter which works effectively when compared with existing methods and also delay characteristics of the proposed method are discussed. In section 2 we discuss functionality of existing leading zero counting methods. In section 3 we discuss about the proposed leading zero counter, the derived Boolean relations that describes the proposed leading zero counter. In section 4 the implementation results and the comparison with the existing method are discussed and section 5 concludes the paper follows by the references.

II. FUNCTIONALITY OF EXISTING LEADING ZERO COUNTING METHOD

This section discusses the method which is used in leading zero counting, their implementation and functionality. A leading zero is any zero bit that leads a number string in positional notation. In binary representation consider the number of consecutive zeros that appear in a word before the first more significant bit that equal to one. The latter is called leading digit. Leading zeros occupy most significant digits, which could be left blank or omitted for the same numeric value. The process of encoding these leading zeros is called leading zero counting [1][6]-[8]. The n bits are assumed as input $A=A_{N-1},A_{N-2}\ldots A_0$ in the LZC, where A_{N-1} is the most significant bit and produces $\log_2 n$ bits of the leading-zero count Z and a flag V that denotes the all-zero case for the input A. The existing method is based on the two step encoding procedure. The position of the leading digit of the input is marked first and then the remaining bits are set to zeros.

For example, let us consider the input as 00011010 the position of the leading digit is determined as 00010000. An intermediate string S is produced to derive the one hot representation. The bits of S that follow the leading digit are set to one and the other more significant bits are set to zero. For the same input the value of S is equal to 00011111. The i^{th} bit of S is denoted as S_i , $0 \le i \le n-1$ is defined as follows

$$S_i=A_{n-1}+A_{n-2}+\ldots..+A_{i+1}+A_i.$$

Each bit in the equation reveals the existence of at least one bit equal to 1 between the MSB and the i^{th} bit position. The one-hot representation of the leading digit (L word) is determined by detecting the case (0,1) for two consecutive bits of S.

$$L_i=\underline{S}_{i+1}.A_i$$

The value of L_{s-1} is equal to the value of S_{n-1}. Here (.) represents the logical AND and $\underline{x}$ represents compliment operations respectively. The priority encoder [1] computes the value of L. the encoder translates the number of leading zeros to its weighted binary representation when the L word is given to it. For the case of an 8-bit input operand the number of leading zeros, Z bits, are given by

$$Z_2=L_3+L_2+L_1+L_0$$
$$Z_1=L_5+L_4+L_1+L_0$$
$$Z_0=L_6+L_4+L_2+L_0$$

The all-zero flag V is set to $\overline{S}_0$ which shows that no bit is equal to one in A. In dynamic CMOS floating-point-unit implementations this approach is mostly preferred. The leading-zero count is computed in few logic stages by employing wide dynamic OR gates for the computation of both the S and the Z bits.

III. PROPOSED LZC

This section discusses about the proposed leading zero counter and the mathematical equations are derived for the proposed leading zero counter. The Boolean relations that describe the bits of the leading-zero count are simplified. The proposed method will be presented using an example of an 8-bit LZC unit. Then the value of the encoded bits Z_0,Z_1 and Z_2 are

$$Z_2=\overline{S}_4.S_3+\overline{S}_3.S_2+\overline{S}_2.S_1+\overline{S}_1.S_0$$
$$Z_1=\overline{S}_6.S_5+\overline{S}_5.S_4+\overline{S}_2.S_1+\overline{S}_1.S_0$$
$$Z_0=\overline{S}_7.S_6+\overline{S}_5.S_4+\overline{S}_3.S_2+\overline{S}_1.S_0$$

For the value i>j the pair (S_i,S_j) will never take the value of (1,0) as the string S is monotonically increasing. As the string S monotonically increases we have $S_i.S_j=S_i$ and Si+Sj=Sj for i>j. The tabular column below shows the reduction when i>j.

Table 1:- Reduction Table

S_i	S_j	$S_i+S_j=S_j$	$S_i.S_j=S_i$
0	0	0	0
0	1	1	0
1	1	1	1

By using these equations the value of the encoded bits can be reduced further as

$$Z_2=\overline{S}_4.S_0$$
$$Z_1=\overline{S}_6.S_4+\overline{S}_2.S_0$$
$$Z_0=\overline{S}_7.S_6+\overline{S}_5.S_4+\overline{S}_3.S_2+\overline{S}_1.S_0$$

When we consider the above equations, no normalization is required when the input is equal to zero, the Z bits are also set to zero. We can also further simplify the above equations to get the below mathematical equations

$$Z_2=\overline{S}_4$$
$$Z_1=\overline{S}_6.S_4+\overline{S}_2$$
$$Z_0=\overline{S}_7.S_6+\overline{S}_5.S_4+\overline{S}_3.S_2+\overline{S}_1$$

Instead of assuming the value of the string S we can directly compute the leading zero counting from the input operand A. So the expression of Si is represented by the value of A_i. The final equations are thus derived as follows

$$\underline{V}= A_7+A_6+A_5+A_4+A_3+A_2+A_1+A_0$$
$$Z_2=A_7+A_6+A_5+A_4$$
$$Z_1=(\underline{A_7+A_4})[(\underline{A_3+A_2})(A_7+A_6+A_5+A_4)]$$
$$Z_0=[\underline{A}_7.\underline{(\underline{A_7+A_6}).A_5)}]\underline{[(\underline{A_7+A_6})(\underline{A_5+A_4})][A_3+(\underline{A_3+A_2})A_1}]$$

The proposed leading zero counter is shown in the figure below.

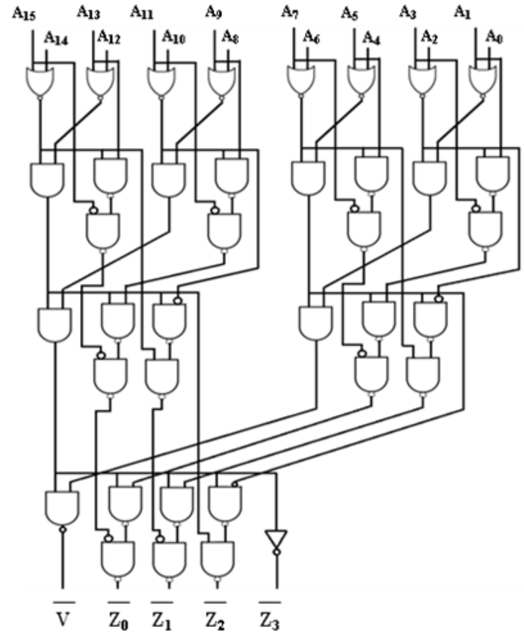

Figure 1:- 16 bit implementation of proposed leading zero counter

Thus by the above mathematical derivations we can say that the proposed leading zero counter works effectively and the total delay is reduced compared with the existing leading zero counters.

IV. IMPLEMENTATION RESULTS

The proposed Leading Zero Counter is implemented on Spartan 3E family device XC3S250E package FT256 with speed -4 and the delay characteristics and the occupancy rates are compared with the existing method and tabulated below. From the results below the number of slices occupied by the proposed method are 10 which is less than 1% occupancy rate. The delay is the total combinational delay and is obtained as 11.189ns. The proposed Leading Zero Counter is implemented using Verilog HDL on Xilinx ISE 10.1 tool.

Table 2:- Comparison of Proposed method with Existing method

Sl.no.	Parameters	Existing Method	Proposed Method
1	No. of Slices	12/2448	10/2448
2	No. Of 4 Input LUTs	21/4896	17/4896
3	No. of bonded IOBs	21/172	21/172
4	Combinational Path Delay	14.293ns	11.189ns

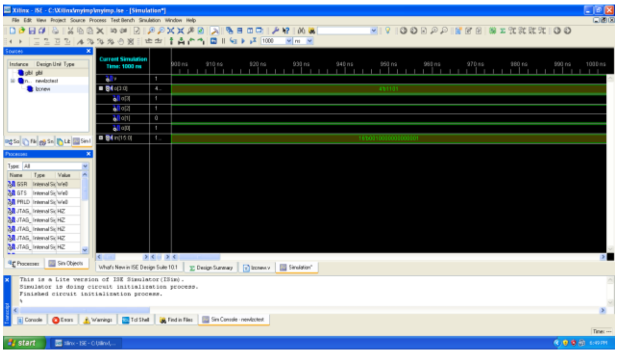

Figure 2:- screen showing the active low output of the proposed LZC for the given input 16 bit binary 0010000000000001.

V. CONCLUSION

In this paper a novel Leading Zero Counter is proposed and the delay characteristics of the proposed LZC are significantly reduced when compared with the existing methods. The design is implemented using Verilog HDL and verified using extensive directed-random vectors. The proposed leading zero counter can be utilised in many application areas like RISC, CISC, Microprocessors and DSP. This proposed LZC can be effectively utilised in high speed floating point units.

REFERENCES

[1] Giorgos Dimitrakopoulos, *Member, IEEE*, Kostas Galanopoulos, Christos Mavrokefalidis, *Student Member, IEEE*,and Dimitris Nikolos, *Member, IEEE* "Low-Power Leading-Zero Counting and Anticipation Logic for High-Speed Floating Point Units" *IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 16, NO. 7, JULY 2009*

[2] *IEEE Standard for Binary Floating-Point Arithmetic*, Std 754-1985, American National Standards Institute and Institute of Electrical and Electronic Engineers, 1985.

[3] A. Beumont-Smith, N. Burgess, S. Lefrere, and C. C. Lim, "Reduced latency IEEE floating-point standard adder architecture," in *Proc. 15th IEEE Symp. Comput. Arithmetic*, Jul. 2001, pp. 35–42.

[4] P. M. Seidel and G. Even, "Delay-optimized implementation of IEEE Floating - point addition," *IEEE Trans. Comput.*, vol. 53, no. 2, pp. 97–113, Feb. 2004.

[5] S. D. Trong, M. Schmookler, E. M. Schwarz, and M. Kroener, "P6 binary floating-point unit," in *Proc. 18th IEEE Symp. Comput. Arithmetic*, Jun. 2007, pp. 77–86.

[6] J. D. Bruguera and T. Lang, "Leading-one prediction with concurrent position correction," *IEEE Trans. Comput.*, vol. 48, no. 10, pp. 1083–1097, Oct. 1999.

[7] M. S. Schmookler and K. J. Nowka, "Leading zero anticipation and detection: A comparison of methods," in *Proc. 15th IEEE Symp. Comput. Arithmetic*, Jul. 2001, pp. 7–12.

[8] V. Oklobdzija, "An algorithmic and novel design of a leading zero detector circuit: Comparison with logic synthesis," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 2, no. 1, pp. 124–128, Mar. 1994.

[9] S. P. Boyd, S. J. Kim, D. Patil, and M. A. Horowitz, "Digital circuit optimization via geometric programming," *Operations Res.*, vol. 53, no. 6, pp. 899–932, Nov./Dec. 2005.