JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A LINEAR 24 BIT MULTIPLIER WITH VLSI APPLICATIONS USING SORTING BASED BINARY COUNTERS

[1]Pinniboyina Anand Venkat Seshu Babu, [2]K. Rajasekhar

[1]M.Tech Student, [2]Assistant Professor
[1]Electronics and Communication Engineering,
[1]University College of Engineering (A) JNTUK, Kakinada, India.

Abstract: The paper presents a new approach to rapid counters, including binary counters, approximation (4:2) compressors, and so on, all based on a sorting network. In order to speed things up, a high compressor counter is required. After the counter inputs are asymmetrically divided in half, they are used as inputs to arranging organizations to produce arrangements that can be addressed by basic one-hot code groupings. The (7:3) counter, which can be built and developed using this way, outperforms other designs in terms of latency, overall size, and power consumption in most situations. The result is a counter with a factor of 15, which has the advantages of lower power consumption and smaller footprint, but at the expense of increased latency. As an extra, we use approximation compressors that employ a sorting network (4:2). They made a 8 x 8, 16 x 16 digit multiplier to test the viability of their circuits. Viability of a 24Bit multiplier configuration is delivered and reproduced in Xilinx Vivado.

Index Terms – Binary counters, sorting network, approximate 4:2 compressor, 24 bit multiplier and One-hot code.

I. INTRODUCTION

The counters dense N columns into log2n lines. That was exhibited a symmetric stacking structure. While it is unsaturated, it is quite rapid in contrast to other patterns. They then utilize a MUX to create a (7:3) saturated counter on the key path, which affects the swiftness. It is also advised to combine three (5:3) counters to create a (15:4) counter and to reduce the planned (6:3) counter to a (5:3) counter. However, this tactic is useless. We start by using design as the primary benchmark for comparison. They devised a fast counter with this symmetric stacking structure, and built another counter, the 7:3 saturated counter, from it (6:3). The 7:3 counter design is the quickest of the seven, but it has poor postpone decrease since it adds a multiplexer to a course prior to streamlining it. To start with, rather than the standard symmetric stacking structure, we stack two sorting networks in an asymmetrical fashion. To expedite multiplication, approximate multipliers are frequently utilized. Stall encoding evaluations and incomplete item hole are utilized to infer an unpleasant multiplier.

Compressors with high-speed approximation (4:2) utilize a symmetric stacking structure. In this concept, saturating counters with better efficiency include (7:3) and (15:4). We provide an asymmetric classification scheme for networks in this set of architectures. Then, through using two extended bits towards logical simplicity, the new system is designed. Using the arranging organization, we can likewise construct accurate/surmised (4:2) blowers. As its name suggests, the arranging network is an elite exhibition equal equipment network utilized for information grouping. A sorting network can sort any number. Might sort a collection of information that use the well-known 0, 1 principle when its members are all 1-bit integers. In this article, 1-bit data sorting is the only method used.

Typical 3 and 4 way sorting networks. Every standing bar in a sorting network functions as a sorter with two data inputs and outputs, each of which are one-bit values. Always sort the bigger input first, then the smaller.

Binary Data Sorter: As previously mentioned, the sorter arranges two inputs according to their quantitative data. Two 1-bit data sets may well be sorted with ease using the logical circuit. While 3 and 4 way sorting networks employ 3 layers of 2-input basic logic gates, a sorter just needs 1 layer of these gates.

We are not required to compute with great precision or precision if little inaccuracies do not even significantly affect the outcomes. As a result, approximation computing is promoted as a cutting-edge technique with great accuracy.

In approximation computing, there is a system architecture which has high performance and is energy-efficient. In such complex calculations, addition errors are much more sensitive than multiplication errors, hence multipliers may be able to accept a greater approximation than adders. The adoption of approximation arithmetic circuits should improve these applications' performance and power efficiency in addition to the quality of deep learning and image processing. A rapid system that is less complex and power-intensive results from approximated arithmetic operations. The compromise accuracy, that wouldn't necessarily issue with machine learning and multimedia applications operating as normally. The inability of the human eye to detect minute differences in images and videos.

II. EXSITING WORK

By creating a series of one-hot codes, an effective (7:3) counter was created. They have developed three Boolean operations that simplify the Boolean equations with outputs.

Looking at Fig. 1, we notice a counter with both the ratio (7:3). The C2, C1, and S are not required again for H and I sequences. Therefore, a useful (7:3) saturation counter is created.

1) 7 & 8 Way Sorting Networks: To achieve its goal, this sorting network employs a complex combination of six tiers of simple logic gates. By removing one bit, an 8-way sorting network may be contracted to a 7-way arranging network comprised of 6 layers of essential rationale entryways. In particular, the successions H (which contains H1-H8 and is reached out to H0-H9) and I (which incorporates H0-H9) are the 8-way and 7-way results of arranging organizations, separately (incorporates I1-I7 and is stretched out to I0-I8). Each of the unique code sequences P (P0-P8) and Q is generated using A&B logic (Q0-Q7).These patterns resemble those found in the counter (7: 3).

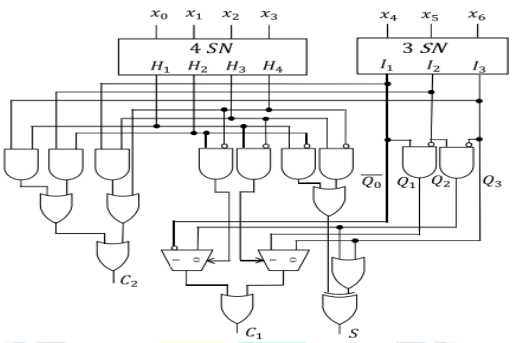

Fig. 1: (7:3) Counter

2) For anyone interested, the counter's (15: 4) four-bit output looks like this: C3C2C1 S. Counterattack (15:4) Like we did with the counter, start by introducing subscripts into logic equations to establish a relationship between C3C2C1 S and the groupings P and Q. (7:3). Additionally, the first Boolean assertions are exorbitantly extensive, so we utilize the Verilog linguistic structure to portray them all things considered.

3) Overall Structure: Hey Transport is a module that for the most part utilizes Stomach muscle rationale doors to process the vital signs, hence its overall structure is rather simple. R1 = H1&I7... R7 = H7&I1, and R8 = H need seven AND operations to be done between the sequences H and I, and the resulting sequences are labelled R1 through R8.

4) The principle behind a (4:2) compressor is similar. These give classification networks that may be used to make a fast (4:2) compressor. In order to address the divergence brought on by insufficient sorting, the output expressions have been modified.

III. PROPOSED WORK

Arithmetic Logic Units perform arithmetic operations such as adding, subtracting, cubing, squaring, multiplying, and dividing (ALUs). It's no secret that multiplication is the ALU's go-to operation and most fundamental function. It allows you to multiple one integer by another. A 24 bit multiplier is used to multiply two different 24 bit values.

Given that 1 is greater than 0, all ones appear at the start of the series and all zeros appear at the end, as shown in Fig. 3. The definition of a series. If both ones and zeros exist, the rearranged sequence must have a location where the two 1s and 0s cross. In order to ensure that the 0, 1-junction exits forever, alter the sequence if it only includes ones and zeros by putting one at the top and a zero at the bottom.

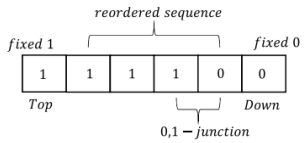

Fig. 2: Definition of a sequence.

The number of 1s and 0s in the rearranged sequence is the same as in the original. It is also not counted as a 1 since it is clear that the extra ones will change the overall number of 1s in the new sequence.

Since this is the case, the time required to sort data using a 3 way or 4 way network is comparable. On this premise, we split the seven contributions of a (7:3) counter in two. There are four pieces in a single segment and three pieces in another.

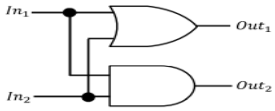

Fig.3: Two-input binary sorter.

Locate the encodings 0, 1-Junction, and One-Hot. A rearranged sequence is represented by the junction under the extended fixed ones and zeros. Therefore, we'll continue to use the 4 SN as an example. Ultimately, it led to the formation of the Figure 1 structure. The design relies on a mathematical equation to produce a new sequence. Given that the rearranged sequence has just one junction, the sole possible value for the first digit of the sequence is 1.

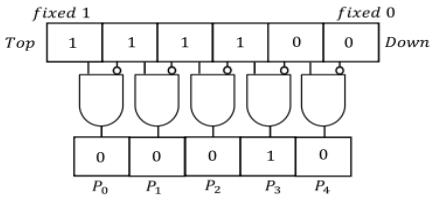

Fig 4: One-hot code generation circuit.

Consequently, it states that the grouping P0-P4 is just a one-hot encoding that fulfills the circumstances.

$$P0|P1|P2|P3|P4 = 1. \quad (1)$$

As shown in the equation, the sequence is randomly split into two sections (2)

$$P0|P2|P4 = P3|P4. \quad (2)$$

To generate one hot encoding sequence Q0-Q3 from the sequence coming from 3SN, an identical technique is performed.

1) We were provided the sequences P and Q to generate an output. P0 = 1 means that the input sequence for such 4N sorting network has no ones, P1 = 1 indicating that there is one 1, Pi = 1 that there I 1s in the input sequence, and Q that there are I 1s in the input sequence. Three values, C2, C1, and S, are generated based on a 7:3 counter, with C2 carrying the highest weight.

$$S= 22C2 + 21C1 + 20 \quad (3)$$

When C2 is set to 1, the (7:3) counter requires a minimum of four 1s in its input sequence. While Q0 = 1 signifies the shortfall of 1s in succession I, P4 = 1 shows the presence of four 1s in grouping H. P4&Q0 = 1 demonstrates that there are 4 + 0 = 4 1s in the 7-piece input. Whenever the amount of P and Q's addendums is somewhere around 4, C2 is equivalent to 1. So, it's possible that you'll write C2 as

$$C2= (P4\& (Q0|Q1|Q2|Q3))| (P3\& (Q1|Q2|Q3))| (P2\& (Q2|Q3))|(P1\&Q3). \quad (4)$$

In the same way as in (2), we may write an equation for the sequence Q if it holds:

$$Q0|Q1|Q2|Q3 = 1 \quad (5)$$

$$Q1|Q2|Q3 = Q0. \quad (6)$$

Put (4) and (5) into (3), we get

When the subscripts of the sequences P and Q are added, C1 = 1, 2, 3, 6, and 7. Consequently, we have

$$C1= (Q0\& (P2|P3))| (Q1\& (P1|P2))|(Q2\&(P2|P3))|$$

$$(Q3\& (P1|P2))| \quad (7)$$

$$S = (P1|P3) \oplus (Q1|Q3). \quad (8)$$

Where ⊕ denotes XOR.

2) Additional optimization: Sequences H1-H4 and I1-I3 are at our command. Here, we add H0 and H5 to the sequence H1–H4. For sequence I, repeat. I0 and I4 both have constant values of 1. The equations derived are:

If the sequences' subsequences are covered and their subscripts are sequential, then the sequences I and H may be used as an alternative representation for the combined Q and P sequences.

In earlier investigations, a counter and compressors was utilized to reduce the multiplier count. The proposed design may be stretched to larger bit multipliers in a manner that is substantially efficient to past work without modifying the architecture of previously recommended multipliers. Without altering the 8*8 multiplier from of the previous work is needed to construct a 24*24 multiplier that use the previous design. An effective saturation counter will be a (15:4) and (31: 5) in certain circumstances.

IV. EXPERIMENTAL RESULTS

The Verilog code may be synthesised into an RTL schematic, where the individual circuit components, such as modules and submodules, and the gate-level design, can be seen.
The RTL schematic is shown in figs. 5,6,7,8,9,10

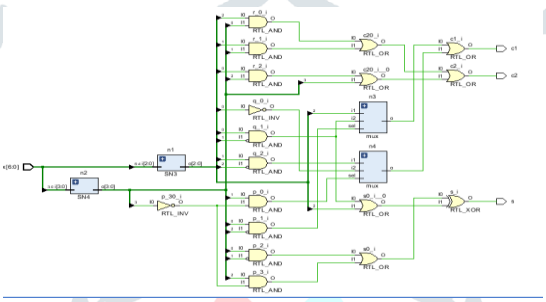

Fig. 4: (7:3) counter RTL Schematic

Synthesis of the code may produce a technology schematic, which is similar to an RTL schematic. The top block is all that can be seen, since there are many more blocks below it that would need too much space to display.

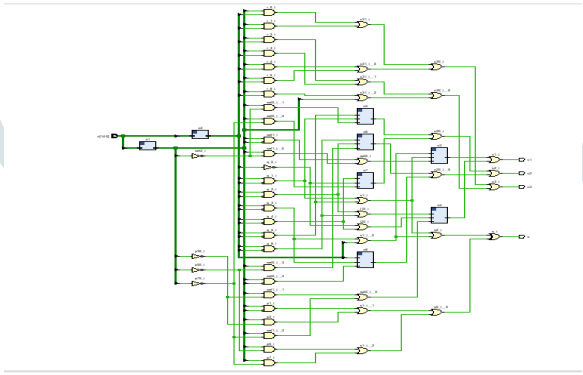

Fig. 5: (15:4) counter RTL Schematic

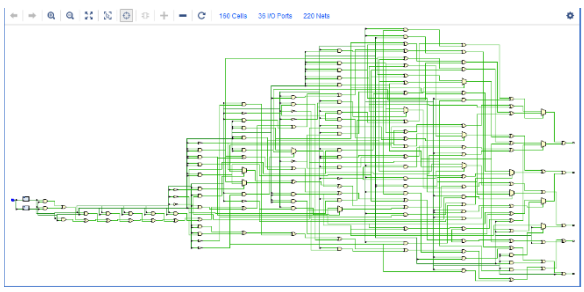

Fig.6: (31:5) counter RTL schematic

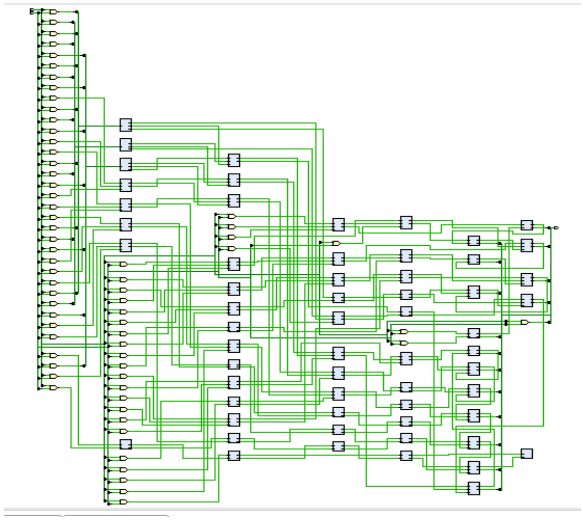

Fig .6: MUX 8 RTL schematic

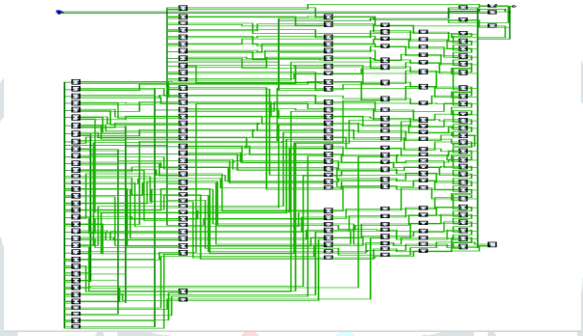

Fig.7: MUX 16 RTL schematic

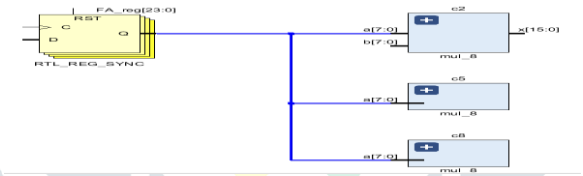

Fig.8: MUX 24 RTL schematic

When developing the Verilog code, simulation is required to verify the output's accuracy and to observe the output waveform. The resulting waveforms appears in Fig. 11

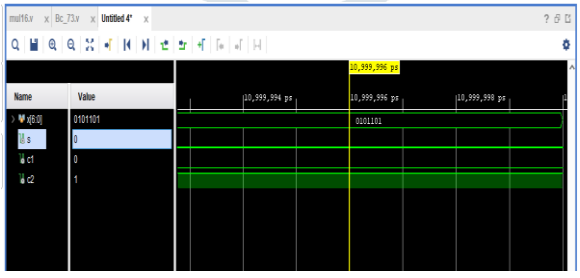

Fig.9 :(7:3) counter

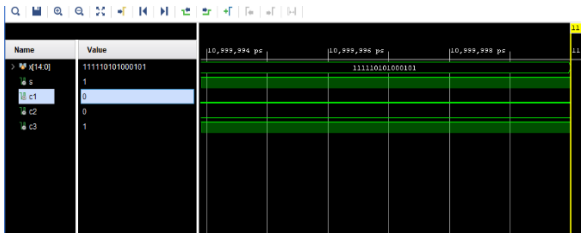

Fig.10:(15:4) counter

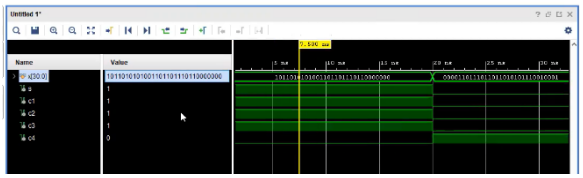

Fig.10: (31:5) counter

Fig.11: MUX 8

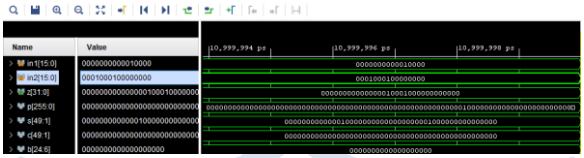

Fig.12: MUX 16

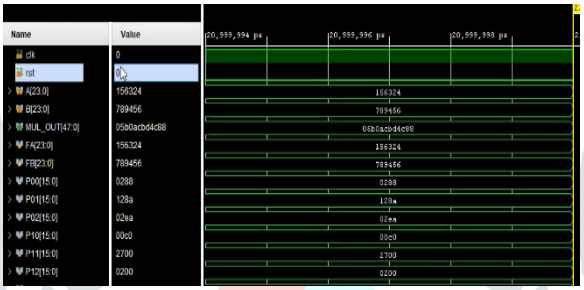

Fig.11:24 bit mux Simulation Results

Evaluation table for Area, Delay:

		AREA(LUT's)	DELAY(ns)	POWER(Watts)
BC_7,3	Proposed	6	2.643	0.079
	Existing	6	5.804	2.022
BC_15,4	Proposed	44	6.285	0.353
	Existing	44	8.775	3.205
BC_31,5	Proposed	140	9.277	1.036
	Existing	139	11.776	5.632
MUL8	Proposed	127	8.827	1.369
	Existing	122	14.041	14.188
MUL16	Proposed	748	14.968	10.433
	Existing	748	17.002	10.579
MUL24	Extension	1128	10.546	14.906

V. CONCLUSION

In increasing the proposed multipliers, which are 8 bit multipliers, into 24 bit multipliers while modifying their design, we may achieve optimum parameter values for higher bit multipliers. It may therefore be applied to higher bit multiplications.

This would result inside an area-efficient circuit because the area for higher order multipliers would've been the same as that of an 8 bit multiplier.

VI. REFERENCES

[1] C. S. Wallace, A suggestion for a fast multiplier, IEEE Trans.Electron. Compute. vol. EC-13, no. 1, pp. 14–17, Feb. 1964, doi:10.1109/PGEC.1964.263830.

[2] R. S. Waters and E. E. Swartz lander, A reduced complexity Wallace multiplier reduction, IEEE Trans. Comput., vol. 59, no. 8,pp. 1134–1137, Aug. 2010, doi: 10.1109/TC.2010.103.

[3] P. L. Montgomery, Five, six, and seven-term karatsuba-like formulae, IEEE Trans. Comput., vol. 54, no. 3, pp. 362–369, Mar. 2005, doi:10.1109/TC.2005.49.

[4] S. Asif and Y. Kong, Analysis of different architectures of counter based Wallace multipliers, in Proc. 10th Int. Conf. Comput. Eng. Syst. (ICCES), Cairo, Egypt, Dec. 2015, pp. 139–144, doi: 10.1109/ICCES.2015.7393034.

[5] S. Asif and Y. Kong, Design of an algorithmic Wallace multiplier using high speed counters, in Proc. 10th Int. Conf. Compute. Eng. Syst. (ICCES), Cairo, Egypt, Dec. 2015, pp. 133–138, doi:10.1109/ICCES.2015.7393033.

[6] C. Fritz and A. T. Fam, Fast binary counters based on symmetric stacking, IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25, no. 10, pp. 2971–2975, Oct. 2017, doi: 10.1109/TVLSI.2017.2723475.10.1109/EDSSC.2017.8126527.

[7] M. Mehta, V. Parmar, and E. Swartzlander, High-speed multiplier design using multi-input counter and compressor circuits, in Proc.10th IEEE Symp. Comput. Arithmetic, Grenoble, France, Jun. 1991, pp. 43–50, doi: 10.1109/ARITH.1991.145532.

[8] A. Fathi, B. Mashoufi, and S. Azizian, Very fast, high-performance 5-2 and 7-2 compressors in CMOS process for rapid parallel accumulations, IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 28, no. 6, pp. 1403–1412, Jun. 2020, doi: 10.1109/TVLSI.2020.2983458.

[9] A. G. M. Strollo, E. Napoli, D. De Caro, N. Petra, and G. D. Meo, Comparison and extension of approximate 4-2 compressors for low-power approximate multipliers, IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 67, no. 9, pp. 3021–3034, Sep. 2020, doi: 10.1109/TCSI.2020.2988353.