WWW.IJRPR.COM

**International Journal of Research Publication and Reviews**

Journal homepage: www.ijrpr.com ISSN 2582-7421

# Binary Counters Based on Symmetric Stacking

***B.Venkatesu[1], G.Navyasri[2] , L.Pranathi[2], P.Jyoshna[2] , R.Pranavi[2]***

*[1]Working as Assistant Professor in Electronics and Communication Engineering Dept,[2]Pursing BTech in ECE Dept*

*[2]Pursing BTech in ECE Dept ,Puttaparthi (M),(D)*

**ABSTRACT:**

This project is to design binary counter using solely full adders and after with new symmetric stacking method. Evaluating these two techniques and displaying how the symmetric stacking method is decreasing the EX-OR gate delays in the essential route of the binary counter. This kind of our proposed counter is very useful in the existing counter based totally Wallace tree multiplier design. With this new symmetry stacking counter we are lowering delay and Power Consumption.The designing and simulating our proposed quick binary counter incorporated in Xilinx ISE layout suite 14.5

Key words: Binary Counter , Full Adder , Symmetry Stacking , Wallace Tree , Xilinx 14.5

## INTRODUTION:

The present disclosure may be embodied as a counting method that uses bit stacking circuits followed by a method of combining two small stacks to form larger stacks. A 6:3 counter built using this method uses no XOR gates or multiplexers on its critical path. VLSI simulation results show that the presently-disclosed 6:3 counter is at least 30% faster than existing counter designs while also using less power. Simulations were also run on full multiplier circuits for various sizes. The same Counter Based Wallace (CBW) multiplier design was used for each simulation while the internal counter was varied. Use of the presently-disclosed counter improves multiplier efficiency for larger circuits, yielding 64- and 128-bit multipliers that are both faster and more

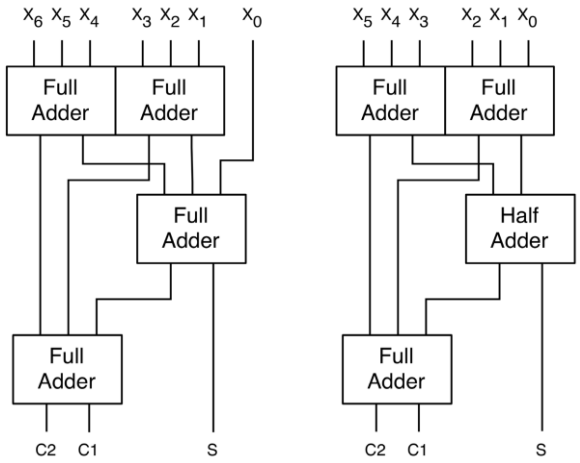


**Figure 1 A 7:3 counter and a 6:3 counter built from full and half adders**

efficient, at least by 25% and 40%, respectively, in terms of power-delay product (PDP). It outperforms the fastest in terms of latency and it consumes less power than the most efficient, meaning that the use of the presently-disclosed counter in a CBW multiplier yields a pure gain

## LITERATURE SURVEY

1.C. S. Wallace, "A suggestion for a fast multiplier," IEEE Trans. Electron. Comput., vol.

In this a design is developed for a multiplier which generates the product of two numbers using purely combinational logic,and it is found that the cost of the unit would be about 10 per cent of the cost of a modern large-scale computer.

2.S. Veeramachaneni, L.Avinash, M. Krishna, and M. B. Srinivas, "Novelarchitectures for efficient (m, n) parallel counters,".
In this paper, novel architectures and designs for high speed, low power (3, 2), (7, 3), (15, 4) and (31,5) counters capable of operating at ultra-low voltages are presented,based on these counters, a generalized architecture is derived for large (m, n) parallel counters,. The proposed counter designs have been compared with existing designs and are shown to achieve an improvement of about 45% in delay and a reduction of about 25% in power consumption.

3.S. Asif and Y"Design of an algorithmic Wallace multiplier using high speed counters,".

Wallace tree multipliers provide a power-efficient strategy for highspeed multiplication. The use of high speed 7:3 counters in the Wallace tree reduction can further improve the multiplier speed. This paper presents an algorithmic approach to construct the counter based Wallace tree multipliers, The detailed comparison of traditional and counter based Wallace multipliers is performed which shows that the counter based Wallace multiplier is up to 22% faster as compared to the traditional Wallace multiplier.

## SYMMETRIC BIT STACKING:

The present disclosure proposes a new paradigm for computer arithmetic designs called bit stacking. Exemplary circuits are presented that stack 3 bits, then 6 bits, and a discussion is provide to show how the principles used in those circuits can be used to build circuits that stack vectors of many more sizes. In the present disclosure, the focus will be on the use of these bit stacking circuits to design binary counters. Ultimately, these counters may used in, for example, large multiplier circuits to produce substantial savings in terms of latency and power consumption over the prior art counter designs. However, although these counter circuits based on bit stacking will achieve high performance and low power consumption, a large part of the delay and complexity comes from converting the bit stacks to binary counts. Thus, the present disclosure provides applications that do not require binary counts but can make use of the bit stacks directly. Such applications can achieve very large performance improvements using bit stacking.Bit stacking is an intuitive concept that can be visualized as pushing all of the '1' bits together. The length of the bit stack is exactly the number of '1' bits in X, which the present disclosure will define later as the Population Count of X. Determining the count from a stacked vector is much simpler than adding up the '1' bits to determine the count.

### *THREE BIT STACKING :*

In the exemplary 6-bit stacker, the primitive stacking circuit we will use is a 3-bit stacker. Given inputs $X_0$, $X_1$, and $X_2$, a 3-bit stacker circuit will have three outputs $Y_0$, $Y_1$, and $Y_2$ such that the number of '1' bits in the outputs is the same as the number of '1' bits in the inputs, but the '1' bits are grouped together to the left followed by the '0' bits.

Given inputs X0, X1, and X2, a 3-bit stacker circuit will have three outputs Y0, Y1, and Y2 such that the number of "1" bits in the outputs is the same as the number of "1" bits in the inputs, but the "1" bits are grouped together to the left followed by the "0" bits. It is clear that the outputs are then formed by Y0 = X0 + X1 + X2

(1) Y1 = X0X1 + X0X2 + X1X2

(2) Y2 = X0X1X2.

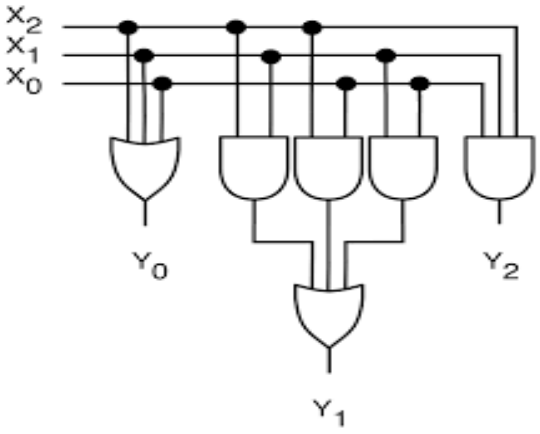


**Figure 2Three bit stacking circuit**

### MERGING STACKS :

The two smaller stacks may be merged into one large one.We wish to form a 6-bit stacking circuit using the 3-bit stacking circuits discussed. An example for this process is shown below

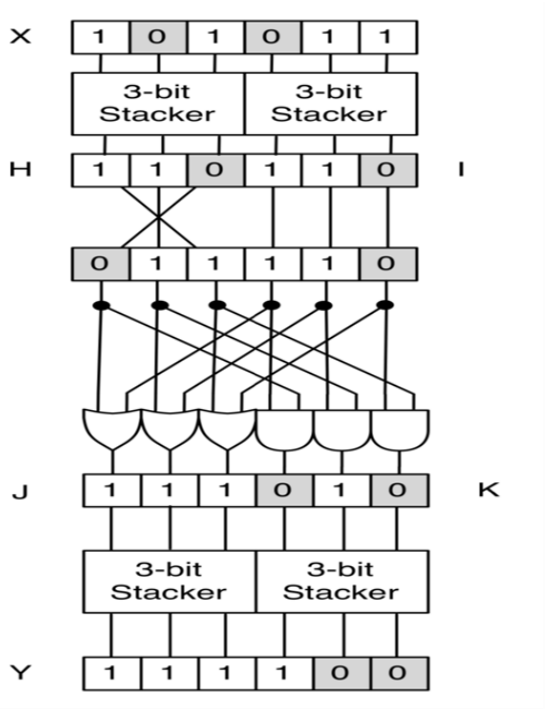


**Figure 3Six bit stacking example**

The two smaller stacks may be merged into one large one. We wish to form a 6-bit stacking circuit using the 3-bit stacking circuits discussed. Given six inputs $X_0, \ldots, X_5$, we first divide them into two groups of three bits which are stacked using 3-bit stacking circuits. Let $X_0$, $X_1$, and $X_2$ be stacked into signals named $H_0$, $H_1$, and $H_2$ and $X_3$, $X_4$, and $X_5$ be stacked into $I_0$, $I_1$, and $I_2$. First, we reverse the outputs of the first stacker and consider the six bits $H_2H_1H_0I_0I_1I_2$. See the top of FIG. 3 for an example of this process. We notice that within these six bits, there is a train of '1' bits surrounded by '0' bits. To form a proper stack, this train of '1' bits should start from the leftmost bit.

In order to form the 6-bit stack output, two more 3-bit vectors of bits are formed called, $J_0J_1,J_2$ and $K_0$, $K_1$, $K_2$. The objective is to fill the J vector with ones before entering ones in the K vector. So we let:

$J_0=H_2+I_0$ (5)

$J_1=H_1+I_1$ (6)

$J_2=H_0+I_2$ (7)

In this way, the first three '1' bits of the train fill into the J bits although they may not be properly stacked. Now to ensure no bits are counted twice, the K bits are formed using the same inputs but with AND gates instead:

$K_0=H_2I_0$ (8)

$K_1=H_1I_1$ (9)

$K_2=H_0I_2$ (10)

In this way, the J bits will fill before the K bits. Focusing on the J vector, if one, two, or three of the input bits are set, we can see from Equations, 5, 6, and 7 that the same number of J bits will be set as none of the expressions have any overlap but together they cover all six bits from H and I. Now we consider the K bits. Note that Equations 8, 9, and 10 give the three possibilities when four input bits are set: either the left 3-bit stack was full and we had one '1' bit from the right 3-bit stack, or there were two '1's in each of the 3-bit stacks, or we had one '1' but from the left 3-bit stack and a full stack on the right. Because the inputs to the AND gates are spaced by three bits, and because $H_0$, $H_1$, $H_2I_0$,$I_1I_2$ contains a continuous train of '1' bits if more than three inputs are '1's, any extra bits will spill into the K vector. The entire symmetric bit stacking process for b=6 bits is illustrated

## CONVERTING BIT STACK TO BINARYNUMBER :

We will now use the bit stacking method defined above for b=6 specifically and for general b above to create a 6:3 counter circuit. The 6:3 counter will have six binary inputs $X_0, \ldots, X_5$ and produce three outputs $C_2$, $C_1$, and S such that the binary number $C_2C_1S$ is equal to Pc(X), the number of '1's in the input vector X. Clearly, the length of the '1' bits in the bit stack called Y shown above is the count of the number of '1' bits in X. Also, it is clear that we can identify the length of a bit stack by finding the point at which a '1' bit is adjacent to a '0' bit. Given a proper bit stack called S, we can

identify a stack of length l by checking

$S_{l-1}S_l$ (31)

Furthermore, we can check that the length of the stack of '1' bits in the properly-stacked vector S is bounded by any lengths x and y such $x \leq l < y$ (32)

by checking $S_{x-1}S_{y-1}$ (33)

In order to convert $S_{5:0}$ to a binary count, we need to identify which stack lengths result in each output bit being set. Table 1 shows the minterms which correspond to each case in which the output bits $C_2C_1S$ should be set.

**TABLE 1**

**6:3 COUNTERSIMULATION RESULTS :**

| Design | Gate Delays ($T_{NAND}$) | Latency (ns) | Avg. Power (μW) | Transistors |
|---|---|---|---|---|
| CMOS full adder based | 15 | 3.1 | 270 | 162 |
| Parallel Counter [4,5] | 10 | 2.3 | 181 | 158 |
| Mux-Based [6] | 9 | 1.8 | 268 | 112 |
| Present Design | 7 | 1.4 | 146 | 120 |

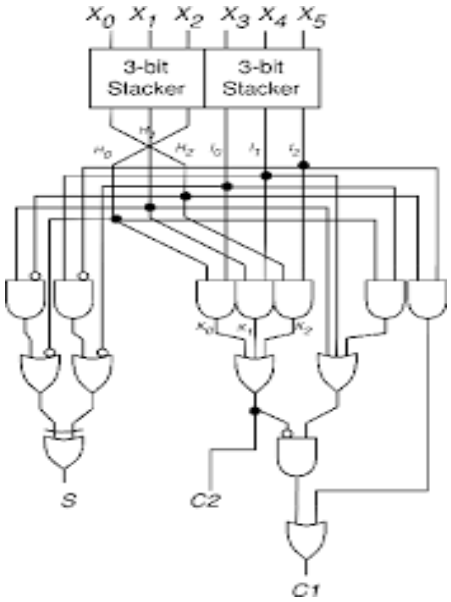


**Figure 4 6:3 counter based on symmetric stacking**

Note that for the mux-based counter design we can take the propagation delay for a mux as $2T_{NAND}$ which follows from the standard implementation of a mux and is also verified by simulation.

Because the presently-disclosed 6:3 counter based on bit stacking has no XOR gates on its critical path, it operates 30% faster and consumes at least 20% less power than all other counter designs. Additionally, the presently-disclosed counter has less wiring complexity in that it uses fewer internal subnets than all other simulated designs. Thus, this method of counting via bit stacking allows construction of a counter for a substantial performance increase at a reduction in power consumption.

**TABLE 2**
**7:3 COUNTERSIMULATION RESULTS :**

| Design | Latency (ns) | Avg. Power (μW) |
|---|---|---|
| CMOS full adders | 3.1 | 350 |
| Parallel Counter [4,5] | 2.2 | 266 |
| Mux-Based[6] | 2.1 | 402 |
| Present Design | 1.8 | 282 |

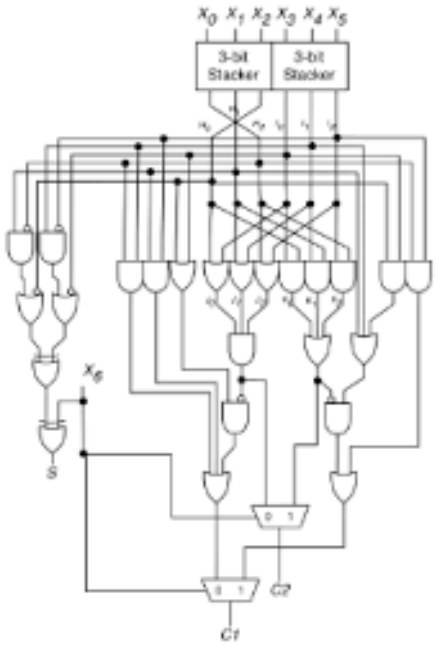


**Figure 5 7:3 counter using symmetric bit stacking**

The present symmetric stacking method can be used to create a 7:3 Counter as well. 7:3 Counters are desirable as they provide a higher compression ratio—the highest compression ratio for counters that output three bits. The design of the 7:3 Counter involves computing outputs for $C_1$ and $C_2$ assuming both the cases where $X_6 = 0$ (which matches the 6:3 Counter) and assuming $X_6 = 1$. We compute the S output by adding one additional XOR gate as this will again not be on the critical path.

[0068] To compute $C_1$ if $X_6 = 0$, we use the same process as before in the 6:3 counter. If

$X_6 = 1$, then everything will shift up as shown in Table 4. The output will be '1' if we have at least one but less than three of $X_0$, ..., $X_5$ as a '1' or if ve of these inputs are '1'. This is because $X_6$ will add one more to the count causing the same case as before.

## FULL MULTIPLIER SIMULATIONS :

To demonstrate a use case of the proposed 6:3 counter, multiplier circuits of different sizes were constructed using different internal counters. No new multiplier design is proposed; rather, existing architectures are simulated with different internal counters. For reference, a standard Wallace tree was implemented for each size. Then, the counter-based Wallace tree was used from which achieves the fewest reduction phases. The internal 7:3 and 6:3 counters used for this CBW multiplier were varied. The 5:3 and 4:3 counters were kept the same for each multiplier, using the counter designs from Standard CMOS implementations were used for the full and half adders. Because of the efficiency of the 6-bit version of the proposed counter, for simulations using the stacker-based counter, we use the 6-bit version with no 7:3 counters, even though this results in one additional reduction phase for each size. An example of a CBW multiplier reduction tree that uses up to 6:3 counters for 16-bit inputs

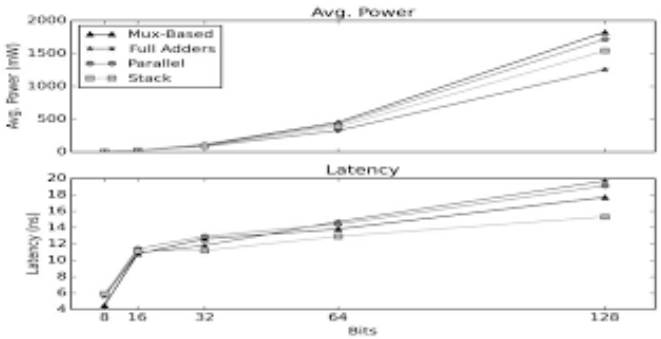


**Figure 6. Power and latency for CBW multipliers with different counters**

## CONCLUSION :

In this brief, a new binary counter based on a novel symmetric bit stacking approach is proposed. We showed that this counting method can be used to implement 6:3 and 7:3 counters, which can be used in any binary multiplier circuit to add the partial products. We demonstrated that 6:3 counters implemented with this bit stacking technique achievehigher speed than other higher order counter designs while reducing power consumption. This is due to the lack of XOR gates and multiplexers on the critical path. The 64-bit and 128-bit counterbased Wallace tree multipliers built using the proposed 6:3 counters outperform both the standard Wallace tree implementation as well as multipliers built using existing 7:3 counters.

## REFERENCES :

[1] C. S. Wallace, "A suggestion for a fast multiplier," IEEE Trans. Electron. Comput., vol. EC-13, no. 1, pp. 14–17, Feb. 1964.

[2] L. Dadda, "Some schemes for parallel multipliers," Alta Freq., vol. 34, pp. 349–356, May 1965.

[3] Z. Wang, G. A. Jullien, and W. C. Miller, "A new design technique for column compression multipliers," IEEE Trans. Comput., vol. 44, no. 8, pp. 962–970, Aug. 1995.

[4] 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, Jun. 1991, pp. 43–50.

[5] S. Asif and Y. Kong, "Design of an algorithmic wallace multiplier using high speed counters," in Proc. IEEE Comput. Eng. Syst. (ICCES), Dec. 2015, pp. 133–138. This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination. IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS 5

[6] S. Veeramachaneni, L. Avinash, M. Krishna, and M. B. Srinivas, "Novel architectures for efficient (m, n) parallel counters," in Proc. 17th ACM Great Lakes Symp. VLSI, 2007, pp. 188–191.

[7] S. Veeramachaneni, K. M. Krishna, L. Avinash, S. R. Puppala, and M. B. Srinivas, "Novel architectures for high-speed and low-power 3-2, 4-2 and 5-2 compressors," in Proc. 20th Int. Conf. VLSI Design Held Jointly 6th Int. Conf. Embedded Syst. (VLSID), Jan. 2007, pp. 324–329.

[8] V. G. Oklobdzija, D. Villeger, and S. S. Liu, "A method for speed optimized partial product reduction and generation of fast parallel multipliers using an algorithmic approach," IEEE Trans. Comput., vol. 45, no. 3, pp. 294–306, Mar. 1996.

[9] S. Asif and Y. Kong, "Analysis of different architectures of counter based Wallace multipliers," in Proc. 10th Int. Conf. Comput. Eng. Syst. (ICCES), Dec. 2015, pp. 139–144.

[10] J. Gu and C.-H. Chang, "Low voltage, low power (5:2) compressor cell for fast arithmetic circuits," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), vol. 2. Apr. 2003, pp. 661–664.

[11] K. Prasad and K. K. Parhi, "Low-power 4-2 and 5-2 compressors," in Proc. Conf. Rec. 35th Asilomar Conf. Signals, Syst. Comput., vol. 1. Nov. 2001, pp. 129–133.

[12] I. Koren, Computer Arithmetic Algorithms, 2nd ed. Natick, MA, USA: A. K. Peters, 2002.

[13] M. Rouholamini, O. Kavehie, A.-P. Mirbaha, S. J. Jasbi, and K. Navi, "A new design for 7:2 compressors," in Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl., May 2007, pp. 474–478.

[14] A. Dandapat, S. Ghosal, P. Sarkar, and D. Mukhopadhyay, "A 1.2-ns 16 × 16-bit binary multiplier using high speed compressors," Int. J. Elect. Electron. Eng., vol. 4, no. 3, pp. 234–239, 2010.

[15] D. Radhakrishnan, "Low-voltage low-power CMOS full adder," IEE Proc.-Circuits, Devices Syst., vol. 148, no. 1, pp. 19–24, Feb. 2001.

[16] S.-F. Hsiao, M.-R. Jiang, and J.-S. Yeh, "Design of high-speed lowpower 3-2 counter and 4-2 compressor for fast multipliers," Electron. Lett., vol. 34, no. 4, pp. 341–343, Feb. 1998

## AUTHORS PROFILE :

**G.NAVYA SRI :**Pursuing B.Tech in the dept of Electronics and Communication Engineering Dept from Sanskrithi School of Engineering ,Puttaparthi (M),(D)

**L.PRANATHI :**Pursuing B.Tech in the dept ofElectronics and Communication Engineering Dept from Sanskrithi School of Engineering ,Puttaparthi (M),(D)

**P.JYOSHNA :**Pursuing B.Tech in the dept of Electronics and Communication Engineering Dept from Sanskrithi School of Engineering ,Puttaparthi (M),(D)

**R.PRANAVI :**Pursuing B.Tech in the dept of Electronics and Communication Engineering Dept from Sanskrithi School of Engineering ,Puttaparthi (M),(D)

**B.VENKATESU (**M.Tech) **:**working as Assistant Professor in Electronics and Communication Engineering Dept from Sanskrithi School of Engineering ,Puttaparthi (M),(D)