F6.

Multiplier Design Utilizing Improved Column Compression Tree and Optimized Final Adder in CMOS Technology

Vojin G. Oklobdzija
Electrical and Computer Engineering Department
University of California
Davis, CA 95616
vojin@ece.ucdavis.edu

David Villeger
Ecole Superieure d'Ingenieurs en Electrotechnique et Electronique
93162 Noisy le Grand CEDEX
FRANCE
villeger@apo.esiee.fr

ABSTRACT

In this paper we discuss improvements in bit reduction techniques and a final adder which is optimized for the uneven signal arrival profile in a CMOS multiplier. Different architectures of the column compression counters and carry propagate adders which take advantage of the speed of the carry signal are considered. The configuration of column compression counters is optimized in order to reduce the longest signal path. The final adder is designed for the uneven input arrival time of the signals originating from the multiplier tree. This results in more compact wiring and balanced delays yielding a faster multiplier.

1. INTRODUCTION

The critical signal path in a parallel multiplier can be divided into 3 domains: Booth encoder, Column Compression Tree (CCT) and the Final Adder (FA). The delay introduced by the Booth encoder is relatively small compared to the other two components especially for the large size multiplier. This component of a delay is also relatively independent of the size of the multiplier. The delay introduced by the CCT and the FA constitutes a dominant component of the delay in the multiplier of which the delay introduced in the column compression tree is about twice that of the final adder. Therefore major enhancement to the speed of a multiplier will result from improvement in those two parts.

In this paper we discuss the efficiency of various known ways of compressing the bit matrix. We will examine the efficiency of the several column compression techniques as applied in [2-5],[7] and discuss ways of optimizing the critical path in the multiplier.

We considered a 24-bit multiplier and we implemented several different configurations which were simulated and analyzed for possible delay optimization.

2. CRITICAL PATH OPTIMIZATION IN THE MULTIPLIER TREE

Signals from the multiplier tree do not arrive at the last stage (Final Adder) at the same time (Fig.2). This is due to the fact that the number of bits to be reduced is larger in the middle of the multiplier tree. Also, it is because we apply carry-save summation (or use counters) in the same way across the entire multiplier tree. This observation leads us to believe that we should reduce the paths in the middle by allowing them to cross into the portion of the multiplier tree with the smaller *depth* (i.e.. number of bits in the column). Also we do apply different types of counters (even full adders) in the middle of the multiplier tree than at the ends. Our objective is to reduce the signal arrival time for the paths in the middle at the expense of the paths at the ends of the tree. Finally, after *flattening* the signal arrival profile as much as possible, we will take advantage of it by tuning the final adder into the resulting profile for an additional gain in speed.

2.1 Use of 4:2 Counters

A major departure from the traditional use of the 7:3 and 3:2 counters in the implementation of the Wallace tree has been the 1981 idea of A. Weinberger of IBM [1] which was made visible by the work of Mark Santoro in [2] and implemented in [3,4]. The significance of a 4:2 counter has not only been in the higher compression ratio of 2, but in the reduced delay due to the ability to redesign the circuitry of the 4:2 counter [4]. The resulting

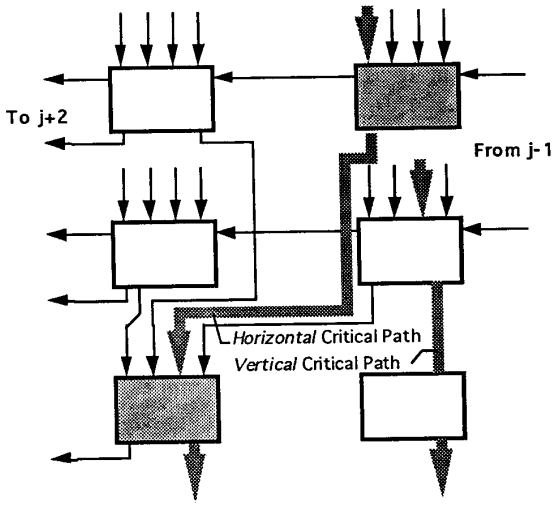


Fig.1. Critical Signal Path in the Multiplier Tree Consisting of 4:2 Counters

* This work has been supported by a mini-grant from the Office of Research, University of California Davis.

delay of 4:2 counter is reduced to 3 XOR equivalent CMOS gates instead of 4 XOR resulting in faster column compression. In our case this results in 11 XOR equivalent gate delays. However, significance of the 4:2 counter has been in the departure from the traditionally *vertical* direction of the signal path. In our design, which uses the counters of higher compression ratio, we allow for *horizontal* as well as *vertical* propagation of the signal. We designate the signal path as *vertical* if the signals travel from the decoders staying in the same bit power position until the final adder is reached, while by *horizontal* we understand the propagation is from lower to higher bit-power position. An example of horizontal propagation is the C_{out} signal from the 4:2 compressor. A critical path in the vertical direction as well as a path consisting of vertical and horizontal directions are shown in Fig.1.

Comparing the signal arrival profiles from the tree implemented from 3:2 counters - RWT (designated as the Regular Wallace Tree) and the tree implemented using 4:2 counters - MWT (designated here as a Modified Wallace Tree) we can observe that MWT has its maximal delay moved slightly toward the right (higher order bits) with slightly reduced maximum. This comes from the fact that the critical path in the tree is not only vertical but horizontal as well.

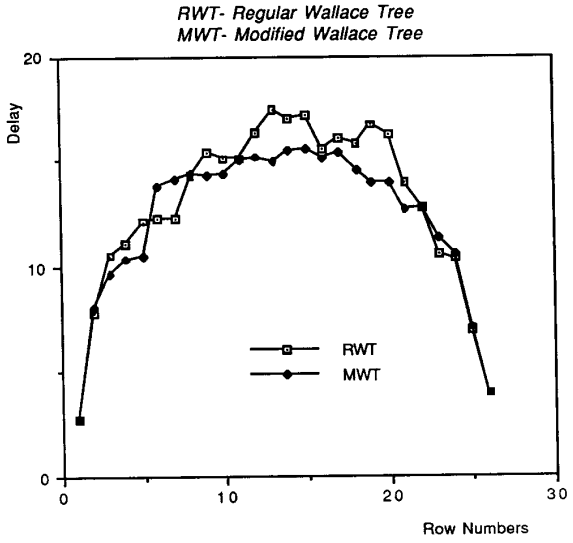


Fig. 2. Signal Arrival Profile from the regular (RWT) and modified (MWT) column compression tree (24-bit example)

We use the fact that we can achieve faster horizontal signal propagation by special design of the column compressors and interleave the CCT in such a way that we use the fastest combinations in what is normally the longest signal path in the bit compression tree. The result of applying mixed sizes of column compression counters is not only a more balanced CCT but also a more even signal arrival profile of the inputs that are fed to the FA as shown in the Fig.2. Optimization of the counter cell for the 4:2 counter results in almost equal propagation times in our example, which is due to the limitation of the CMOS library. However, it should be noticed that the fact that the carry signal can be made faster than the sum would lead to the reduction of the critical path involving *horizontal* propagation of the signals. This is the basis of our design. The schematic of the optimized 4:2 counter has been shown in Fig.3.

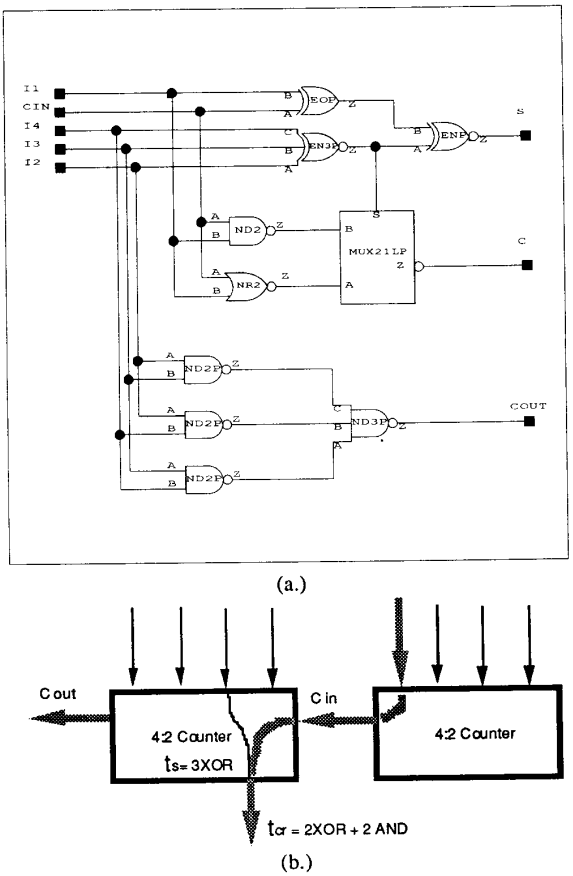


Fig.3. (a.) Optimized 4:2 Counter (b.) Signal Delay in 4:2 counter

The delay profile obtained from the CCT using 4:2 counters is shown in Fig.4. The delay of the middle columns has been reduced at the expense of the side columns and the worse case occurs in the bit 29 position.

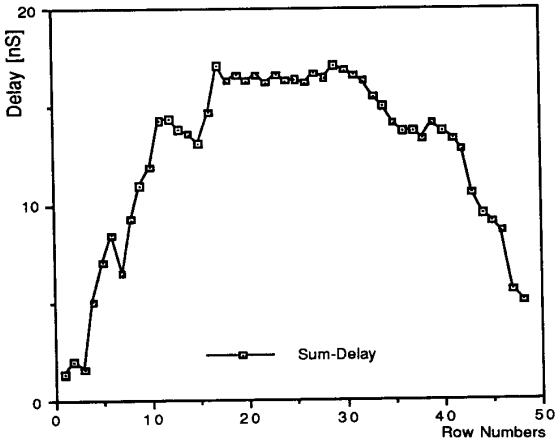


Fig.4. Delay profile of the CCT using 4:2 counters with reduced delay.

2.2. 9:2 Counter

The idea of using higher order counters extends itself into the use of 9:2 and higher such as 27:5 [7]. The advantage of using higher order counters is mainly in more compact and regular wiring, while a disadvantage is the fact that due to their size the efficiency of their use decreases with the size of the counter. The structure of the 9:2 counter is shown in Fig.5.

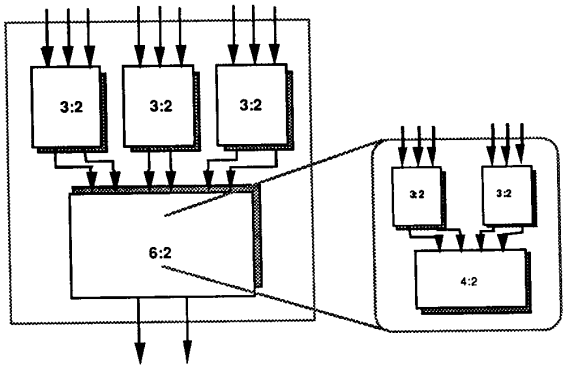


Fig.5. 9:2 Counter

In our case (of a 24X24 bit multiplier) the use of a 9:2 counter would result in two stages. In the first stage we have three 9:2 counters reducing the number of bits to 6 which is compressed into two with one stage of 6:2 counters. Delay of such a tree measured in terms of XOR gate delays is equivalent to 12 XOR gate delays. Fig.6 shows signal delay for the signals from the multiplier tree consisting of 9:2 counters.

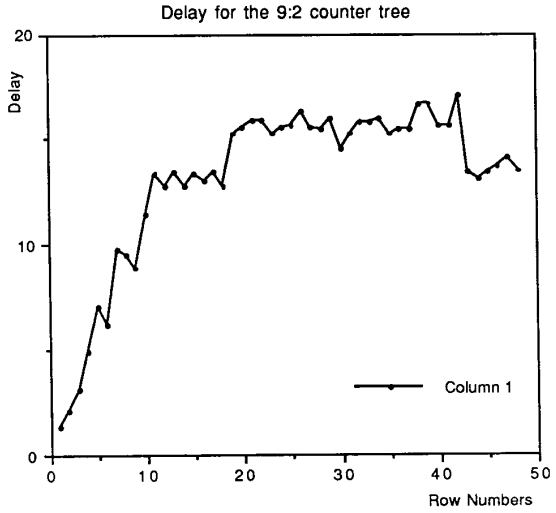


Fig.6. Delay from the multiplier tree implemented using 9:2 counters

2.3 Using 4-bit Adder in the CCT

By introducing *horizontal* and *vertical* signal propagation in the CCT we realized that we can trade some of the speed by adjusting the critical path to travel in the *horizontal* direction for some portion of the signal. We are taking advantage of the fact that we can design a counter (adder) in such a way that carry signal can be made faster than the sum. Therefore, our assumption is that we can accomplish a *horizontal* propagation in the same time as *vertical* , and we know that this is the best we can achieve.

In the case of adders being used instead of a counter, the size will be determined by the time it takes to propagate carry signal. The optimal size will be the size at which the time taken by the carry signal equals that of the sum. We have chosen a 4-bit adder for our analysis. The compression ratio of this scheme is slightly larger than 2, which is due to the fact that we have to collect carry signals after each 4 bits. However, *vertical* delay of the adder is 2 XOR equivalent gates which is better than that of the 4:2. A partial CCT consisting of 4-bit adders is shown in Fig.7.

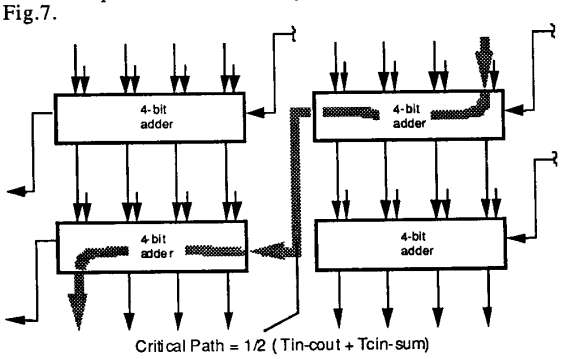


Fig.7 Column Compression Tree consisting of 4-bit Adders. (critical path involves carry propagation and one XOR delay)

Critical path for a 24-bit multiplier tree using 4-bit adders is estimated between 10 and 11 XOR delays depending on how fast carry signal can be propagated. This is faster than what we can achieve with 4:2 and 9:2 counters. (the use of 3:2 counters results in a delay of 14 equivalent XOR delays)

The signal arrival profile from the multiplier tree using 4-bit adders is shown in Fig.8.

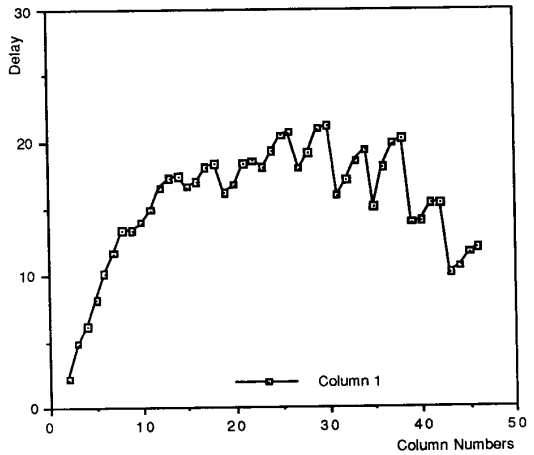


Fig.8 Signal arrival profile for the multiplier tree using 4-bit adders.

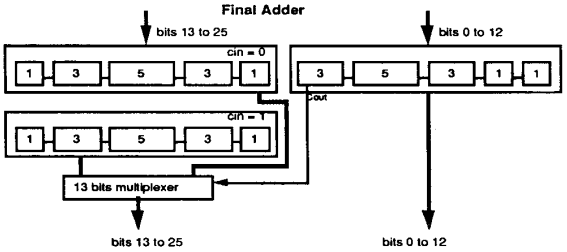
3. SPEED IMPROVEMENT IN THE FINAL ADDER

Finally multiplier speed is improved via optimization of the FA to the non-uniform signal arrival profile of its inputs. It is well known that the signals applied to the inputs of the FA arrive first at the ends of the FA and the last ones are the signals fed to the bits in the middle of the FA. The shape of the signal arrival profile from the multiplier tree is shown in Figures 2,4,6,8.

In choosing the FA scheme we considered the Conditional Sum Adder (CSA) and the Carry Select Adder (CSLA). CSLA is more economical to implement - but can be treated as a subset of CSA. We first optimized them using dynamic programming optimization techniques to find an optimal configuration of the adder resulting in the shortest signal delay for both CSA and CSLA. The optimal CSLA 32-bit configuration is: 1-2-3-4-5-7-10 (for uniform input arrival). When applying a non-uniform signal arrival profile the optimal CSLA configuration is: 1-2-3-1-3-4-2-3-4-8-1 with a corresponding speed improvement.

An important conclusion is that the difference in speed between CSA and CSLA diminishes as the input arrival profile is changed from uniform to non-uniform. Though the CSLA adder is still slower than the CSA adder the difference in speed is offset by the relative simplicity of its implementation with respect to CSA. This is a fortunate finding given that most designers prefer CSLA over CSA. Smaller size and simpler layout of CSLA would further affect the relative speed difference reducing the advantage of CSA.

In our 24-bit implementation we used a combination of CSLA and VBA (variable block adder) for the reason of minimizing the gate count. The FA design is shown in Fig.9.a. and the resulting signal arrival profile from the multiplier tree and FA is shown in Fig.9.b.



(a) Structure of the Final Adder

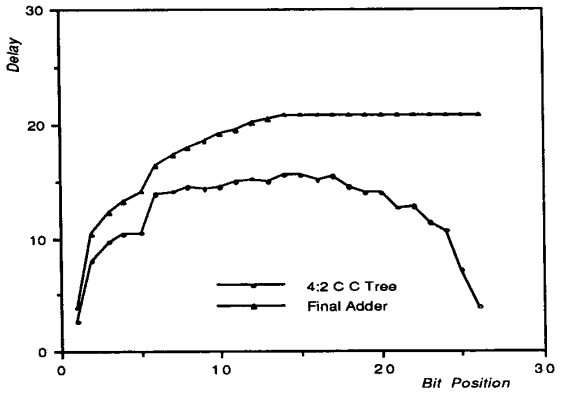


Fig. 9. (b) Signal Arrival Profile from the column compression tree and the final adder

4. CONCLUSION

In this paper we have discussed ways of compressing the bits of the multiplier tree. We examined ways of designing and using different counters and concluded that the use of full adders is warranted because the propagation delay per stage is close to 2 equivalent XOR gate delays (with proper design of the carry path) while the compression ration is close to 2. XOR equivalent delays for various counters in a 24-bit multiplier are shown in Table 1.

Table 1. Delay for various counters applied

Counter Applied	No. Stages	Delay of 24-bit multiplier (equiv. XOR)	No. Stages	Delay of 53-bit multiplier (equiv. XOR)
3:2	7	14	9	18
4:2	3 + 3:2	11	5	15
4-bit adder	5	10+tc	6	12+tc

As it can be observed from Table 1., the use of 4-bit adders starts to show an advantage as the size of the multiplier increases (such as 53-bit). Application of carry-propagate adders in the multiplier tree creates an extra row of carry-out signals from the adders which has to be compressed. Those bits can be absorbed at the ends of the tree due to the fact that there are empty inputs available to the column counters at the ends. The problem arises in the middle of the tree where extra carry bits could not be absorbed. However, this extra row can be eliminated by using a larger carry-propagate adder in the middle and also by careful positioning of the adders (tiling) in a way which enables the carry to be absorbed. This reduces the tree delay to 10 and 12 XOR delays in 24-bit and 53-bit multiplier respectively.

Our actual times did not reflect the above results due to our inability to modify LSI 10K cells which were the only ones available to us at the time. However, our circuit simulation has shown that it is indeed possible to create counter that would satisfy desired features.

ACKNOWLEDGMENT

We gratefully acknowledge assistance from Simon Liu for simulating the 9:2 counter. LSI Logic Corporation has generously supported us with their libraries and ASIC tools.

REFERENCES

[1] A. Weinberger, "4-2 Carry-Save Adder Module", IBM Technical Disclosure Bulletin.,Vol.23, January 1981.

[2] Mark R. Santoro, "*A Pipelined 64X64b Iterative Array Multiplier*", Digest of Technical Papers Int'l Solid-State Circuits Conference, February 1988.

[3] J. Mori et al, "*A 10nS 54X54-b Parallel Structured Full Array Multiplier with 0.5-u CMOS Technology*", IEEE Journal of Solid State Circuits, Vol. 26, No. 4, April 1991.

[4] Masato Nagamatsu et al, "*A 15nS 32X32-bit CMOS Multiplier with an Improved Parallel Structure*", in 1989 Digest of Technical papers, IEEE Custom Integrated Circuits Conference.

[5] T. Soulas, D.Villeger, V.G.Oklobdzija, "*An ASIC Multiplier for Complex Numbers*", Proceedings of EURO-ASIC-93, The European Event in ASIC Design, Paris, France, February 22-25, 1993.

[6] A.K.W.Yeung, R.K.Yu, "*A Self-Timed Multiplier with Optimized Fnal Adder*", Final Report for CS 292I (Prof. Oklobdzija), University of California Berkeley, Fall 1989.

[7] P.Song, G. De Michelli,"Circuit and Architecture Trade-offs for High Speed Multiplication, IEEE Journal of Solid State Circuits, Vol.26, No.9, September 1991.