

High Density and High Speed Multiplier Using Wallace Adders with Pipelining

Reza Hashemian, Senior Member IEEE
Northern Illinois University
Dept of Electrical Engineering
Reza@ceet.niu.edu

I - Introduction

Multipliers are among the most frequently used devices in digital applications particularly in the digital signal processing. The delays associated with high-density multipliers are relatively large and they are the main cause of slow processes in the data flow [1 – 5]. Our objective here is to design high-density multipliers that have high-speed throughput, and have delays not linear but logarithmically proportional to the data size. It is shown that, through a three stage pipelining a balanced delay between stages is better achieved. For instance, for a 32-bit multiplier that uses Wallace adders [3, 4], followed by a carry select two-operand adder [6] the total delay in each stage, and hence that of the multiplier's throughput, is reduced to equivalent to ten (two-input NAND) gate-delays.

II – Design Development

Given two 32-bit operands $A = a_{31} a_{30} \dots a_1 a_0$, and $B = b_{31} b_{30} \dots b_1 b_0$ the multiplication $C = A*B$, presented here, is performed by going through three stages, namely, AND-array generation, Wallace adders reducing the operands into two, and finally, a high speed two-operand adder generating the final product C [2]. An *AND-array*, consist of a two dimensional 32 by 32 array that is simply formed by using AND gates to generate $a_i b_j$ data entries, where i and j are integers 0, 1, ... and 31. The entire AND-array uses 1024 AND gates, and it takes one unit gate-delay to produce $a_i b_j$ entries. The next stage in the operation is the Wallace adders.

Wallace Adders

Wallace adders [3] take column-by-column entries from the AND-array and reduce each column ultimately to a one or two entry column, ready for two-operand addition. Figure 1(a), symbolically, shows the use of Wallace adders in a 4 by 4 multiplier. As illustrated, there are two levels of adders used in this configuration and with three adders in each level we basically need six full adders to construct this Wallace adders structure. Now, if we consider the delay associated to a full adder being almost equivalent to two gate-delays, then for a two level Wallace structure the total delay amounts to four gate-delays. Figure 1(b), shows the numerical listing entries in this 4 by 4 Wallace adders structure. In addition, the number of full adders used in each level is also indicated. Figure 2, similar to Fig. 1(b), shows the numerical listing for a 16 by 16 Wallace adders structure. As

shown, there are 6 levels of full adders used in this structure and the total delay amounts to 12 gate-delays. It is inherent that the number of entries from one level in the Wallace adders structure to the next is reduced by a factor of 3/2. As a result, the number of operands (terms in each column) is logarithmically reduced as we move from one level to the next. Hence, the time delay associated with a Wallace adders structure also takes a logarithmic form in relation to the word size. Table 1 shows the proportionality between the data size and the number of levels (delay) in the Wallace adders, which is very much in accordance to the logarithmic ratios.

Table 1

Data Size	$\text{Log}(n/2) / \text{Log}(3/2)$	# Levels in the Wallace adders
4	1.71	2
8	3.42	4
16	5.13	6
32	6.84	8
64	8.55	10

As indicated, for our particular design of a 32-bit multiplier, Table 1 shows 8 levels for the Wallace structure.

Two operands adder

The final stage in the proposed multiplier is a 64-bit carry select adder developed in [6]. As reported, due to the exponential nature of the carry propagation in this technique the time required for complete generation of carries here is also logarithmically proportional to the size of the operands. This delay for a 64-bit adder amounts to six (2-to-1 MUX) gate-delays, and with adding two more gate-delays for remainder processes, the overall time delay for a 64-bit adder amounts to eight gate-delays.

Pipelining

As indicated earlier, our objective here is to design a 32-bit multiplier using Wallace adders with pipelining. To do an effective pipelining we first need to know the overall time delay for a multiplier of this size, and then break the operation into sections for pipelining. From our previous discussion, we can calculate the entire time delay by adding up the delays corresponding to the stages in the multiplier. A simple calculation reveals that total of 25 gate-delays are needed to complete a multiplication of 32 – bit in size. Now, if we split the entire process into three sections and assign nine gate-delays for each section plus one gate-delay for the associated registers in the pipeline, then we need ten gate-delays to sequentially produce multiplication results through the pipeline. This is almost in the range of a high-speed adder of this size.

Figure 3 shows the numerical listing of the column entries in each level of a 32 by 32 Wallace adders plus the number of full adders used in each level of the structure. Notice

that the entire structure is split into two 4-level sections, separated by a register array, to do the pipelining. In Fig. 4 a more detailed data-structure in each level in the Wallace adders is shown. Again, the structure is partitioned into two 4-level stages for pipelining and a final two operands adder generates the product C. As illustrated, the data is first generated in the AND-array, and the resulted 32 operands flow into the first four levels of full adders. At the end of the fourth level, the data is reduced to a maximum of seven operands, and is stored in the register R1. The second section of the pipeline also consists of four levels of Wallace adders. The data processed in this section of the pipeline is stored in the register R2, which consists of two operands. In the final stage of the pipeline the data from R2 is shifted into a 64-bit carry-select adder and the multiplication product C is finally produced.

Table 2 shows the number of gates and delays associated with a 32-bit multiplier. Note that with three sections pipelining (Fig. 4) we get a balance of 8-gate delays between the sections, and an overall 10 gate delays for the product throughput.

Table 2

Disc.	Pipeline sections	# Gates	# Gate-delays
AND-Array	1	1024 AND	1
Wallace – 1	1	4638	8
Register (R1)	-	252 FF	1
Wallace – 2	2	942	8
Register (R2)	-	117 FF	1
2-Operand Adder	3	642	8
Register (R3)	-	64 FF	1
32-bit Multiplier	-	7679	10

REFERENCES

- [1] N. R. Scott, *Computer Number Systems and Arithmetic*. Englewood Cliffs, NJ: Prentice-Hall, 1985, pp. 54-57.
- [2] B. Parhami, *Computer Arithmetic, Algorithm and Hardware Design*, Oxford University Press, New York, 2000, pp. 114-123.
- [3] C. S. Wallace, "A Suggestion for a Fast Multiplier," *IEEE Trans. Electronic Computers*, vol. 13, pp. 14 – 17, 1964.
- [4] L. Dadda, "On Parallel Digital Multipliers," *Alta Frequenza*, vol. 45, pp. 574 – 580, 1976.
- [5] E. E. Swartzlander, "Parallel Counters," *IEEE Trans. Computers*, vol. 22, no. 11, pp. 1021 – 1024, 1973.
- [6] R. Hashemian, "A New Design for High Speed and High-Density Carry Select Adders", *43rd Midwest Symposium on Circuits and Systems*, Lansing, Michigan, August 8-11, 2000.

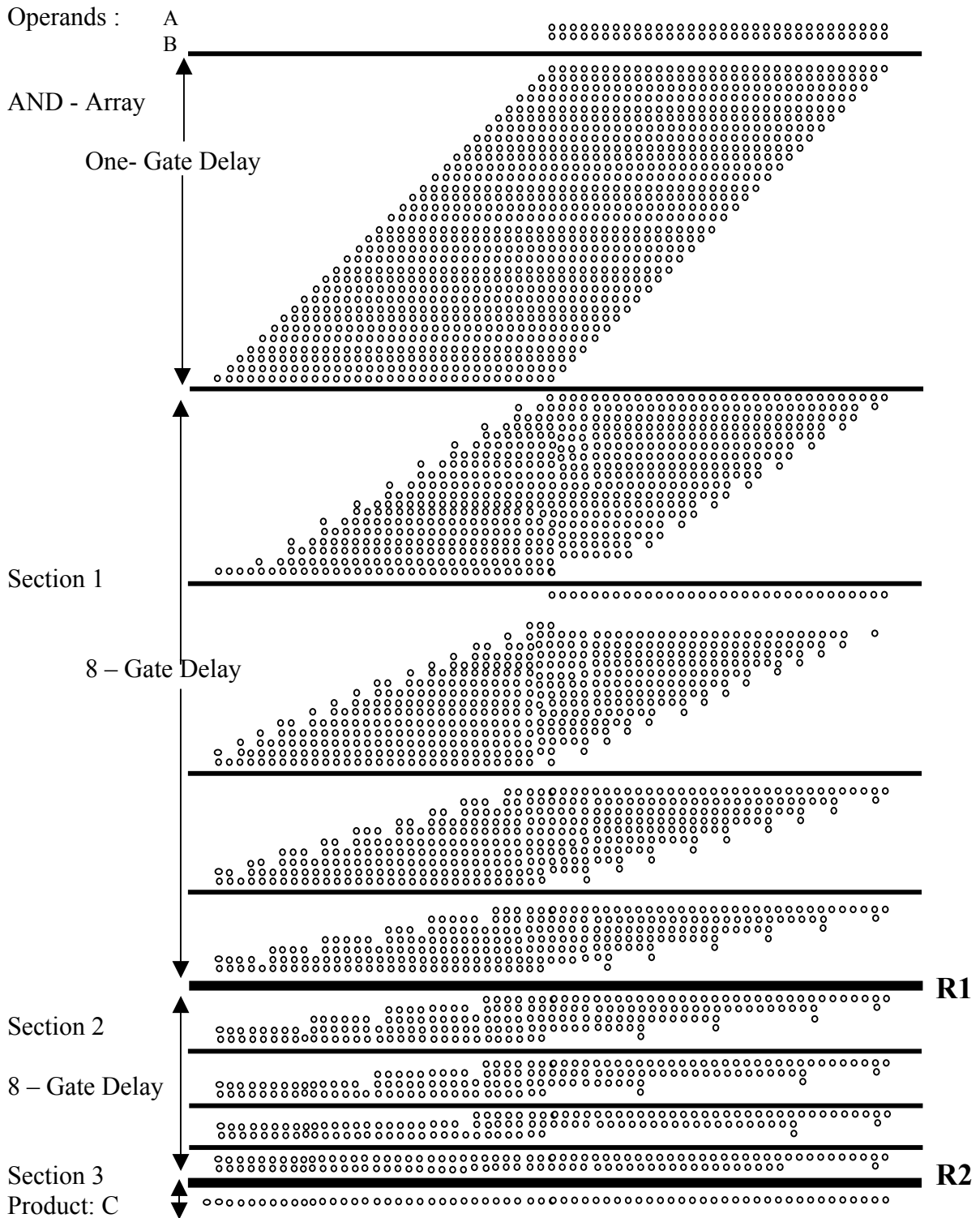


Fig. 4 – Three sections pipelined 32 – bit multiplier using Wallace adders structure.