A FAST SERIAL PARALLEL BINARY MULTIPLIER

Vijay Sharma

Shashank Varshney

Abstract - A Serial parallel binary multiplier design is made using a carry save adder. Latches are used as memory storage elements and allow for slack borrowing in case of fast operation of preceding block. The design is hierarchical and can be extended to n number of bits. For n bit multiplication, 2n clock cycles are required.

I. Introduction

Binary multiplication is required by almost all signal processing applications. Some of them require the multiplication to be done at faster rate. Since large numbers need to be multiplied, there must be a system that can perform multiplication using less hardware and having low power as well as having a design that can be extended without much design variations.

II. Proposed structure

The proposed structure uses a carry save adder. The difference from a conventional full adder is that it takes all three inputs at the same time and produces both sum and carry bit simultaneously[1][2]. Latches are used to store both the sum and carry generated so as to use those values in the next clock cycle. The carry out is used by the same block in the next cycle while the sum is used by the next block. An and gate is used for multiplication at bit level. The output of and gate is the third input to carry save adder.

Thus, we have made a standard cell consisting of a carry save adder, an and gate and two latches which can be replicated if the input size increases. One of the input is fixed while the other is given one bit at a time to the system to simulate the working of multiplication algorithm. At every clock cycle, the input bits are multiplied and then added to the result of previous multiplication. This way, at every clock pulse, we have one output bit coming out of the system starting from the least significant bit and in 2n cycles, we have the complete output of multiplication.

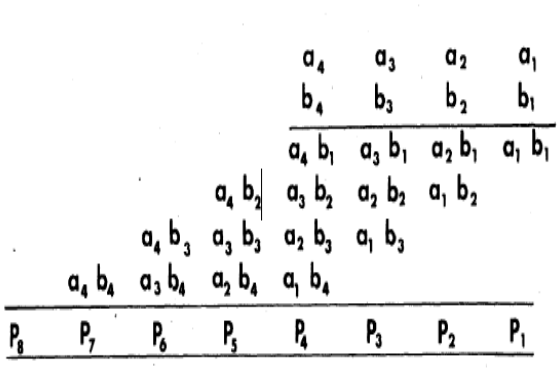

Figure 1 Product matrix

III. Implementation

The algorithm implemented is presented in [1] . the proposed algorithm is implemented using 130nm technology. Latches are designed using transmission gates. Mirror adder has been used for the carry save adder design. Delays of all the components used is tabulated in table 1. The timing calculations used have been formulated as follows

$$T_{clk} >= T_{p,adder} + T_{c\text{-}q,latch} + T_{setup,\ latch}$$

$$T_{+ve} > \max(T_{and},\ T_{c\text{-}q,\ latch}) + T_{setup,adder}$$

$$T_{+ve} < 2T_{c\text{-}q,cd} + T_{setup,adder} + T_{setup,latch} + T_{adder,\ cd}$$

$$T_{-ve} > T_{adder,max}$$

From the above equations, we arrived at a clock cycle of 300ps.

Component	Max(ps)	Min(ps)
AND	65.5	39.85
ADDER	230	207
LATCH	64.76	14.8

Table 1

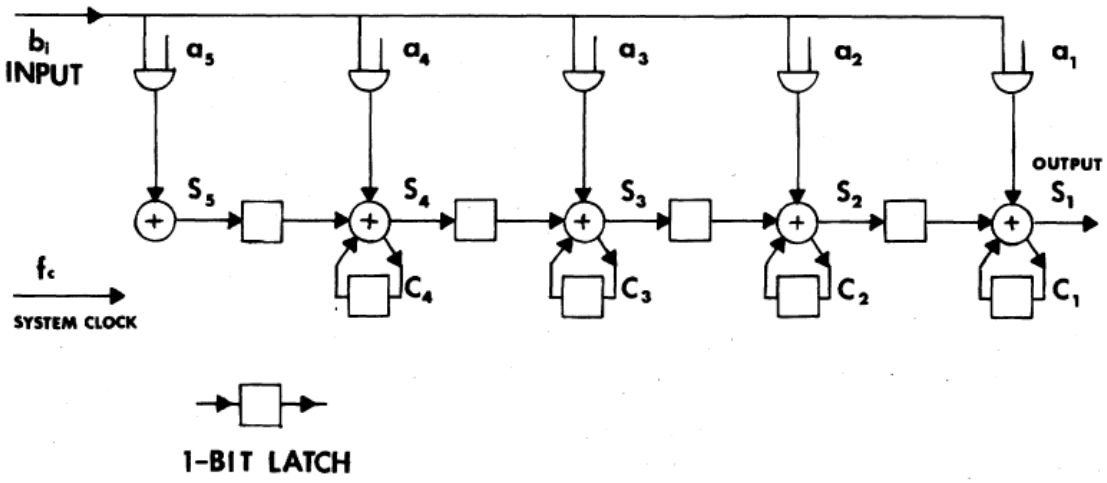

Figure 2 Unsigned multiplier

IV. Comparison

A. Timing comparison –

Comparing the implemented structure with an array multiplier, Implemented structure takes 2n clock cycles as mentioned previously with

$$T_{sum} < T_{clk} < 2\ T_{sum}$$

For an n bit array multiplier the typical delay is

$$T_{mul} = [2n-3]T_{carry} + [n - 1]\ T_{sum} + T_{and}$$

Assuming T_{carry} to be equal to T_{sum},

$$T_{mul} = [3n -4]\ T_{sum} + T_{and}$$

Timing comparison for some cases is tabulated in table 2.

B. Hardware comparison

The implemented structure uses n full adders for n x n bit multiplication.

However, number of adders in an array multiplier is exponential vs number of bits.

No. of bits	Implemented (ns)	Array (ns)
4	2.4	1.84
8	4.8	4.6
16	9.6	10.12
32	19.2	21.16

Table 2

V. Conclusion

The implemented structure is better in terms of both delay and hardware required if the no. of bits is large.

REFERENCES

[1] R. Gnanasekaran,"A Fast Serial – Parallel Binary Multiplier," *IEEE Trans. on Comput.*, vol. C-34, pp. 741-744, Aug. 1985

[2] A. Habibi and P. A. Wintz, "Fast multipliers," *IEEE Trans. Comput.*,vol. C-19, pp. 153-157, Feb. 1970