Akash Kumar, Deepika Sharma / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 3, Issue 4, Jul-Aug 2013, pp.1460-1462

Performance Analysis of Different Types of Adder for High Speed 32 Bit Multiply And Accumulate Unit

Akash Kumar[1], Deepika Sharma[2]
[1, 2](Department of ECE, Manav Rachna International University, Faridabad)

ABSTRACT

The addition and multiplication of two binary numbers is the fundamental and most often used arithmetic operation in microprocessors and digital signal processor. At the heart of data-path and addressing units in turn are arithmetic units such as comparators, adders and multipliers. Since multiplication dominates the execution time of most DSP algorithms, so there is a need of high speed multiply and accumulate (MAC) unit. Paper describes the analysis of speed and delay of different types of adder like carry-look ahead adder, carry skip adder, ripple carry adder and carry select adder for high speed 32 bit MAC unit.

Keywords **–**Carry-look ahead adder, Carry select adder, Carry skip adder, MAC unit, Ripple carry adder.

I. INTRODUCTION

The multiply and accumulate (MAC) unit is the main computed kernel in digital signal Processing architectures. The MAC unit determines the speed of overall system as it is always lies in the critical path. Developing high speed MAC is crucial for real time DSP application. In order to improve the speed of MAC unit, there are two major bottlenecks that need to be considered. The first one is the fast multiplication network and the second one is the accumulator. Both of these stages require addition of large operands that involve long path for carry propagation. A fast multiplication process consists of partial product generation, partial product reduction and final stage carry propagate adder. Accumulator consists of register and adder. Register hold the output of previous clock from adder. Holding output in accumulation register can reduce additional add instruction. An accumulator should be fast in response so it can be implemented with one of fast adder like Carry-look ahead adder or carry skip adder or carry select adder

II. ADDER ARCHITECTURE

2.1 Ripple Carry Adder

This is the simplest design in which the carry out of one bit is simply connected as the carry of the next [1]. It can be implemented as a combination circuit using n full adder in series and is called ripple carry adder.

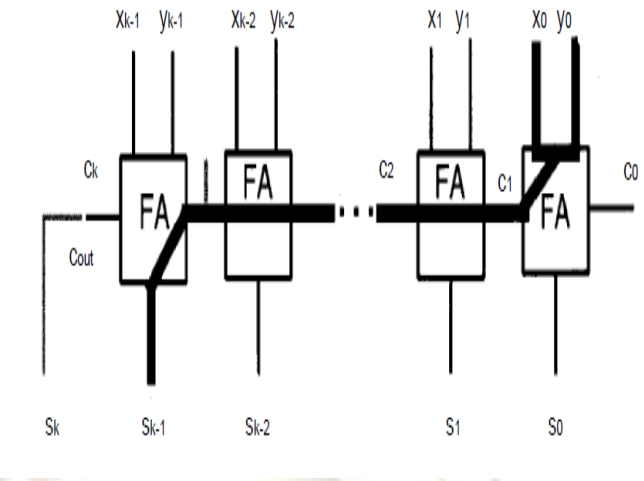

Fig.1 Critical path in k-bit RCA [3]

The latency of K-bit ripple carry adder can be derived by considering the worst case signal propagation path. As shown in fig.1 the critical path usually begins at the x0 or y0 input proceeds through the carry propagation chain to the leftmost FA and terminates at sk-1 output.

2.2 Carry-look ahead adder

The carry-look ahead adder [2] computes group generate signal as well as group propagate signals to avoid waiting for a ripple to determine if the first group generates a carry or not. From the point of view of carry propagation and the design of carry network the actual operand digits are not important. In the case of binary addition the generate, propagate and annihilate signals [3] [4] [5] are characterized by the following logic equations:

$$gi = xi \,.\, yi$$
$$pi = xi \oplus yi$$
$$ai = not(xi + yi)$$
$$ti = xi + yi$$

Carry-look ahead adder hardware may be designed as shown in fig.2. The carry-look ahead adder allows carry for each bit to be computed independently. Carry-look ahead logic consists of two logic levels in which AND gate followed by an OR gate.

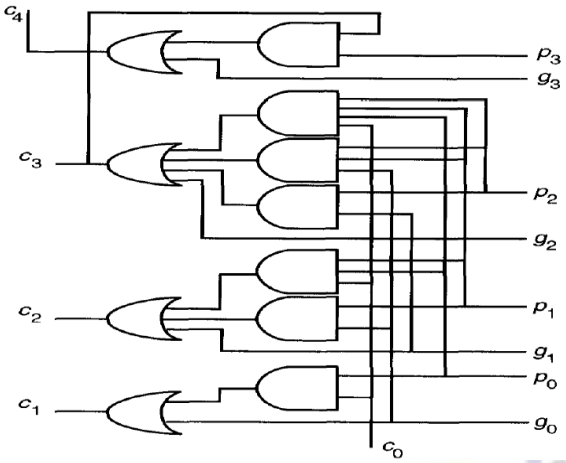

Fig.2 Carry-look ahead logic

2.3 Carry Skip Adder

The carry skip adder [3] [6] was invented for decimal arithmetic operations. The carry skip is an improvement over the ripple carry adder. By grouping the ripple cells together into blocks, it makes the carry signal available to the blocks further down the carry chain, earlier. The primary carry Ci coming into a block can go out of it unchanged if and only if, Xi and Yi are exclusive-or of each other. This means that corresponding bits of both operands within a block should be dissimilar.

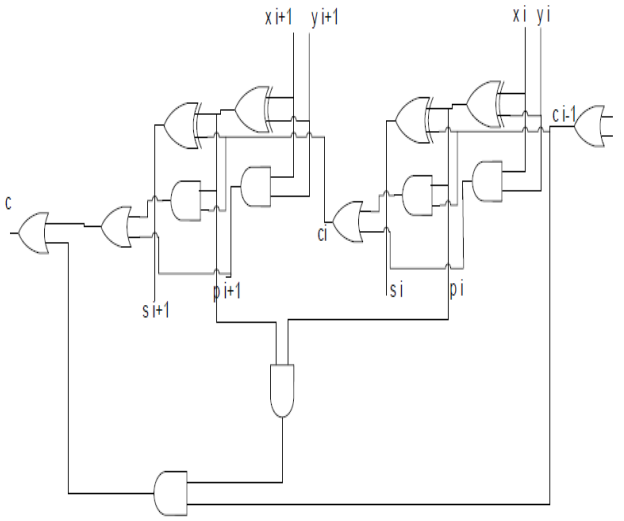

Fig.3 Carry Skip logic using gates [7]

2.4 Carry Select Adder

The carry select adder [3] comes in the category of conditional sum adder. Blocks of bits are added in two ways - assuming an incoming carry of 0 or 1, with the correct output select later as the blocks true carry in become known. With each level of selection, the number of known output bits doubles and leading to a logarithmic number of levels and thus logarithmic time addition.

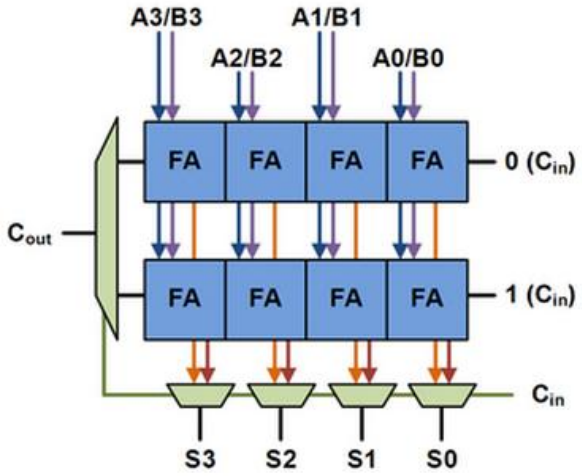

Fig.4 Basic building block of carry select adder

III. SIMULATION RESULTS

All the adders are coded with VHDL, simulated and synthesized by Xilinx ISE 8.1i and Modelsim SE 6.3f. For 32 bit MAC unit, 64 bit adder is used.

3.1 Ripple Carry Adder

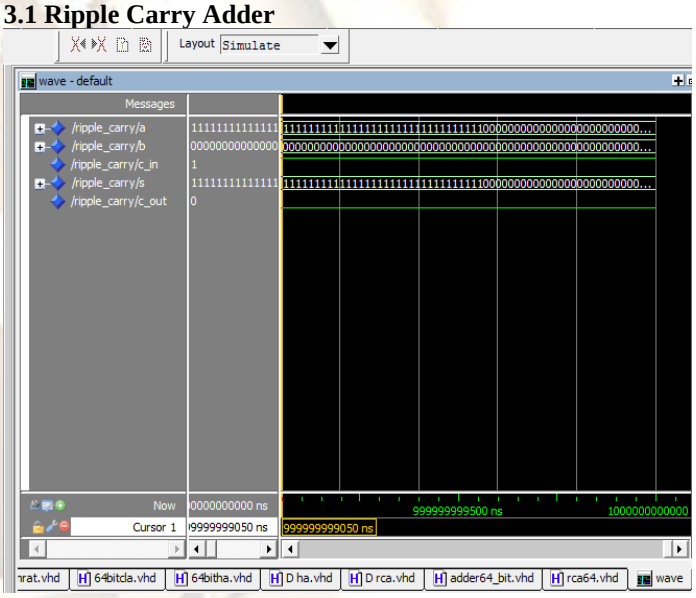

Fig.5 Simulation result of 64 bit Ripple carry adder

Size	16-bits	32-bits	64-bits
No. of Slices	19	37	74
4 input LUTs	33	65	129
Bonded IOs	50	98	194
Level of logic	18	34	66
Logic delay (%)	48.2	43.3	40.3
Route delay (%)	51.8	56.7	59.7
Max. comb. delay	31.698	55.634	70.506

Table 1 Synthesis results of Ripple carry adder

3.2 Carry-look ahead adder

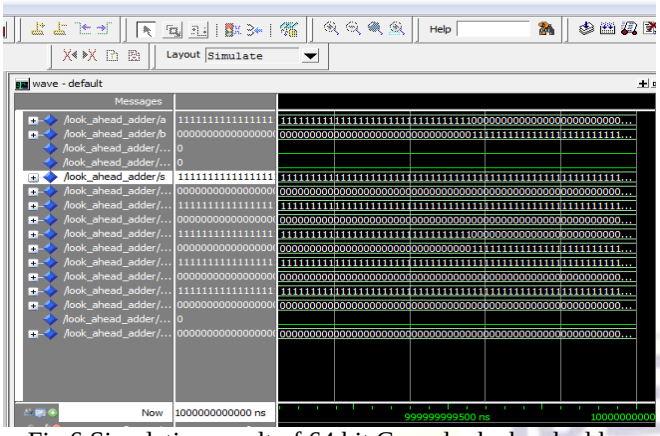

Fig.6 Simulation result of 64 bit Carry look ahead adder

Size	16-bits	32-bits	64-bits
No. of Slices	29	56	113
4 input LUTs	54	104	210
Bonded IOs	50	98	194
Level of logic	11	19	35
Logic delay (%)	65.1	61.9	57.4
Route delay (%)	34.9	38.1	42.6
Max. comb. delay	16.626	26.608	48.31

Table 2 Synthesis results of Carry-look ahead adder

3.3 Carry select adder

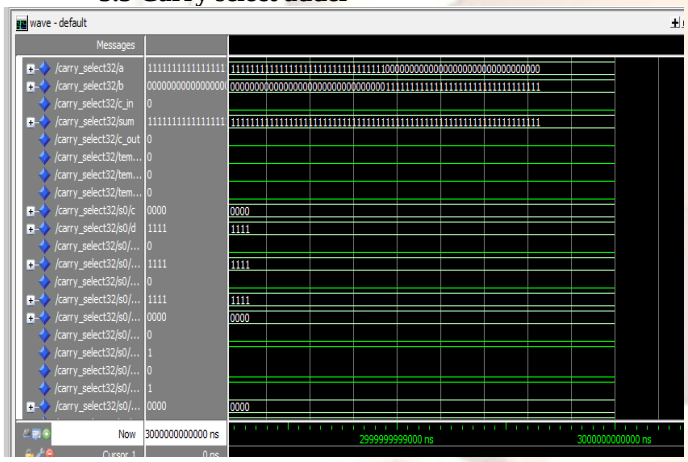

Fig.7 Simulation result of 64 bit Carry select adder

Size	16-bits	32-bits	64-bits
No. of Slices	19	37	74
4 input LUTs	33	64	128
Bonded IOs	50	98	194
Level of logic	15	27	66
Logic delay (%)	65.2	62.1	61.8
Route delay (%)	34.8	37.9	38.2
Max. comb. delay	21.072	36.782	71.209

Table 3 synthesis results of Carry Select adder

3.4 Carry Skip adder

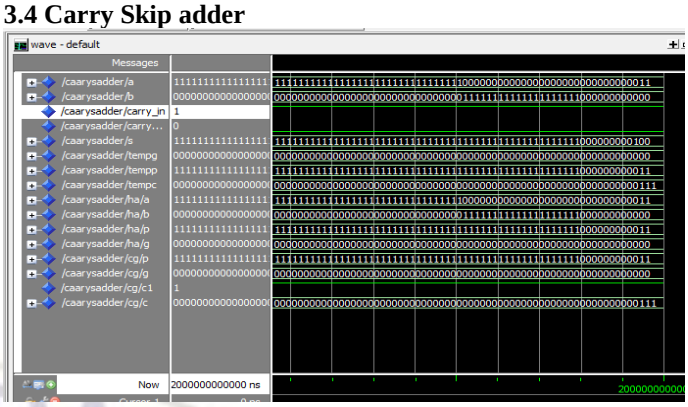

Fig.8 Simulation result of 64 bit Carry Skip adder

Size	16-bits	32-bits	64-bits
No. of Slices	30	61	120
4 input LUTs	55	112	223
Bonded IOs	50	98	194
Level of logic	24	44	90
Logic delay (%)	66.3	63.5	63.2
Route delay (%)	33.7	36.5	36.8
Max. comb. delay	24.368	43.412	80.923

Table 4 Synthesis results of Carry Skip adder

IV. CONCLUSION

The Ripple carry adder is simple in design but it is suitable for only addition of less width operand because the delay goes linearly with width of operands. The Carry skip adder requires a linear area that is hardly larger than the area required by the ripple carry adder. Delay of Carry-look ahead adder is less as compare to other. A carry-look ahead adder is much faster than ripple carry adder but it requires comparatively large area. For 32 bit MAC unit we can use Carry-look ahead adder for high speed multiplication and accumulation. In fact the multiplier with carry-look ahead adder has approximately twice the speed of multiplier.

References

[1] Koc, CK., "*RSA Hardware Implementation", RSA Laboratories*, RSA Data Security, Inc. 1996.

[2] A. Weinberger; J.L Smith, "A Logic for High-Speed Addition*", National Bureau of Standards, Circulation 591*, pp.3-12, 1958.

[3] B. Parhami, "Computer Arithmetic, Algorithm and Hardware Design", *Oxford University Press, New York*, pp.73-137, 2000.

[4] Neil. H. E. Weste, "*Principle of CMOS VLSI Design"* Adison-Wesley 1998.

[5] Raahemifar, K. and Ahmadi, M., "Fast carry look ahead adder" *IEEE Canadian Conference on Electrical and computer Engineering*, 1999.

[6] Kantabutra, V., "Designing Optimum one-level Carry-Skip Adders", *IEEE Transactions on computers*, pp.759, June 1993.

[7] K.C Chang, "*Digital system design with VHDL and Synthesis*" An integrated Approach IEE Computer Society, pp 408-437, 1999.