ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

Implementation of Efficient Parallel Decimal Multiplier Using Verilog

Ch.Tejasri[1], Dr.B.Naga Jyothi[2]

M.Tech Student, Department of Electronics and Communication Engineering, DMS SVH College of Engineering, Machilipatnam, Andhra Pradesh, India[1]

Professor, Department of Electronics and Communication Engineering, DMS SVH College of Engineering, Machilipatnam, Andhra Pradesh, India[2]

ABSTRACT: This paper includes the implementation of decimal multipliers which are arranged in parallel, with the idea of reducing delay. By reducing the delay, multiplication speeds up. This can be done by reducing the number of partial products. This is because delay depends on number of partial products added up in Accumulation Unit. The partial products are generated in parallel by using Signed digit radix-10 recoding of the multiplier and from multiplicand multiples. Repeated additions give Multiplication results. A 32X32-bit parallel decimal multiplier is designed by using three different adders to reduce the delay. By using Regular adder the maximum combinational path delay is 54.293ns, while for Modified adder delay it is 51.962ns and for Reduced delay adder is 41.483ns. This is designed in Verilog HDL, simulated and synthesised using Xilinx 14.7.

KEYWORDS: Delay,Multiplication, Partial products, Regular Adder(RA) Modified Adder(MA), Reduced Delay Adder(RDA), Recoding.

I. INTRODUCTION

In practical applications multiplication is essential component. There is a need of decimal operations in applications like banking operations, financial analysis and so on. . This paper describes the delay comparison of parallel decimal multiplier by using Regular, Modified and Reduced delay BCD adders. Multiplication operation mainly consists of partial products generation and accumulation of those partial products. Efficient implementation of decimal parallel multiplication is done by parallel generation of partial products and reduction of those partial products. This idea can be used to speed up the multiplication process. Digital multipliers are widely used than analog multipliers because the inputs are multiplied as it is done in mathematics.

Decimal multiplication is more complex than binary multiplication mainly for two reasons: The range of decimal digits (0 to 9), which increments the number of multiplicand multiples and the inefficiency of representing decimal values in systems based on binary logic using BCD (since only 9 out of the 16 possible 4-bit combinations represent a valid decimal digit). These issues complicate the generation and reduction of partial products. So the decimal digit range is reduced to {-5 to 5} from {0 to 9} and also generate a sign bit. This recoding leads to minimized calculations as only five multiples are required to be calculated and the negative multiples are obtained using 2's compliment approach for the signed numbers. But for Unsigned numbers, the partial products are generated by AND gates. Full adders are used for addition of generated partial products.

II. LITERATURE SURVEY

In [1],for 16X16-bit input high speed Parallel BCD multiplier a Reduced delay BCD adder has less delay as compared to CLA adder. In [2], the delay is shown to be less in parallel multiplication rather than sequential method. Decimal multiplication is considered as one of the most complicated operations, which requires high-cost hardware

DOI: 10.15680/IJIRCCE.2017. 0505055

implementation. Therefore, the processor industry has opted to use sequential decimal multipliers to reduce the high cost of parallel architectures [3]. The Carry look ahead adder permits the design of a parallel decimal arithmetic unit which is competitive to a binary arithmetic unit in terms of performance [4] and the problem of Multioperand Parallel Decimal Addition with an approach that uses binary arithmetic and suggested adoption of binary-coded decimal (BCD) numbers in [5]. A fixed-point decimal multiplication is proposed that utilizes a simple recoding scheme to produce signed-magnitude representations of the operands thereby greatly simplifying the process of generating partial products for each multiplier digit. The fixed-point decimal multiplication utilizes decimal carry save addition to reduce the critical path delay [6]. Introduced and analysed are three different techniques for performing fast decimal addition on multiple binary coded decimal (BCD) operands [7].

III. Proposed model

The RTL Schematic of the proposed model for the efficient parallel decimal multiplication is shown in Fig.1.By using three different adders Multiplier is designed to produce less delay. Human beings have preferred decimal numbers for all calculations although binary numbers are used as default base in all computers because of the storage and speed efficiency of binary hardware. Subsequently, the designers have preferred binary computers because of the speed and the simplicity of binary arithmetic. Now-a-days, demand for the decimal arithmetic hardware in financial and commercial applications is increasing.This model gives the simple calculation of partial products and the final product is obtained by their addition.

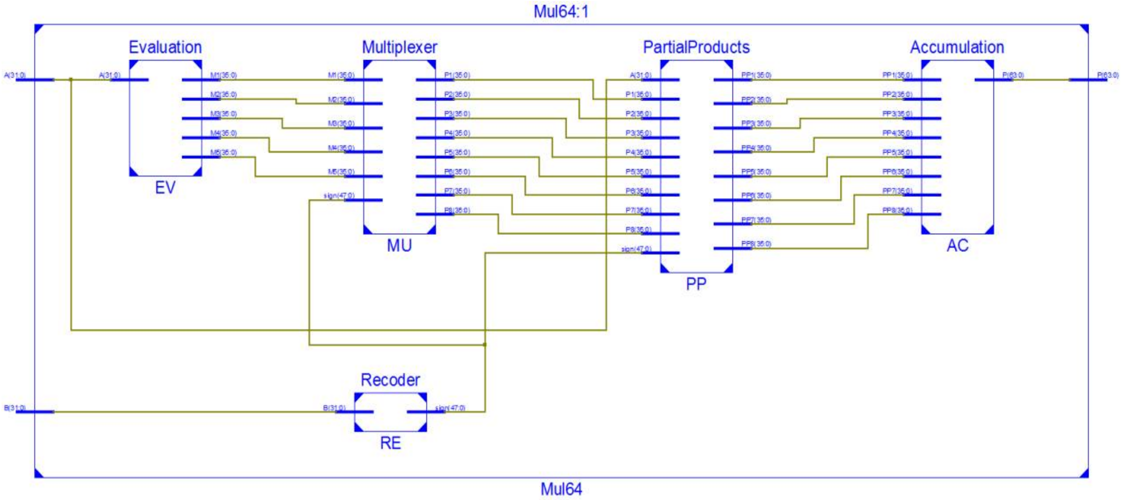


Fig.1.Parallel Decimal Multiplier

A. *Evaluation Block(EV):*

The input of this block is Multiplicand A which is a 32-bit input. This can be used to calculate the multiplicand multiples 1A, 2A, 3A, 4A and 5A for calculating the final multiplication product. The inputs A and B are Multiplicand and Multiplier respectively & in BCD form. These multiplicand multiples are produced by Regular adder, Modified adder and Reduced delay adder. These adders are used in both the Evaluation block and Accumulation block.

Regular adder is a Ripple carry adder. The carry should propagate from LSB to MSB so the main result should wait for these calculations from LSB to MSB. This leads to more time taking process &finally the delay is more.The Modified adder is a Carry Look Ahead Adder&has less delay path than ripple carry adder. In this adder a pre-correction stage is required but the post-correction utilization is eliminated. In pre-correction stage, first the input A and B are added and if the result is above or equals to eight an addition of 0110 is done for correcting the sum. If the result is invalid combinations(1110 or 1111), they are further represented as 8 and 9 respectively. The Reduced delay Adder is a Kogge adder. Ithas three stages, i.e., Pre Processing, Carry Look Ahead networking and Post Processing. Every adder has a Logicaldelay and Route delay. Logical delay is the time required for the entire operation and Route delay is the path delay from source to destination. For Kogge stone adder Route delay is small so it produces less delay as compared to other adders.

B. *RecoderBlock(RE):*

The decimal digits {-5 to 5} are recoded from {0 to 9} by using signed digit Radix-10 method. The numbers 1to 5 doesn't require recoding but from 6 to 9 are recoded as -4 to-1. Every decimal digit from 0 to 9 in BCD numbers is considered by 4-bits so this can be recoded as 6-bits by hot one encoding technique. In 6-bits,MSB represents sign bitand the other 5-bits represent the digit. If the BCD number is equal to or greater than five, then the sign bit becomes negative otherwise it is positive. Another five digitsare treated as selection bits at the output of the recoding block.

C. *Multiplexer(MU):*

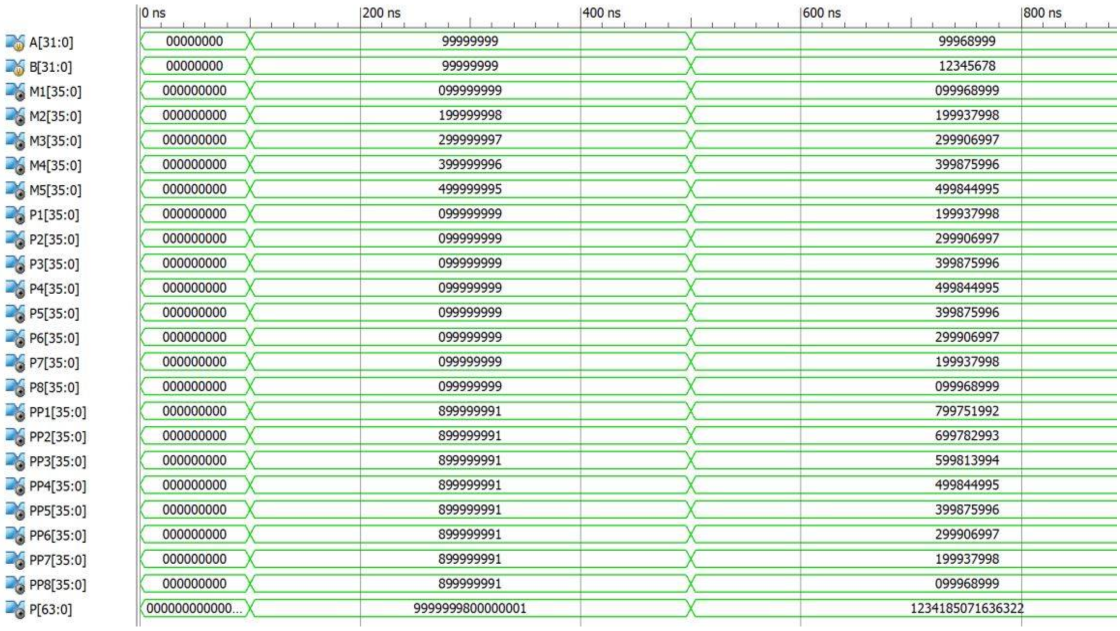
Multiplexer is a data selector which selects the multiplicand multiples 1A, 2A, 3A, 4A and5Afrom recoding unit by using the selection bits. If the sign bit is positive it chooses the positive multiples otherwiseif the sign bit is negative it selects negative multiples.

D. *Partial products(PP)and Accumulation Block(AC):*

The partial products P1 and P2…P8,PP1and PP2…PP8 are generatedand the final result is obtained by left shifting and adding those partial products. Finally the result is obtained from Accumulation unit. If the input digit is of size n then the partial products length is of n+1 and the final product length may be maximum of 2n length.

IV. Simulation results

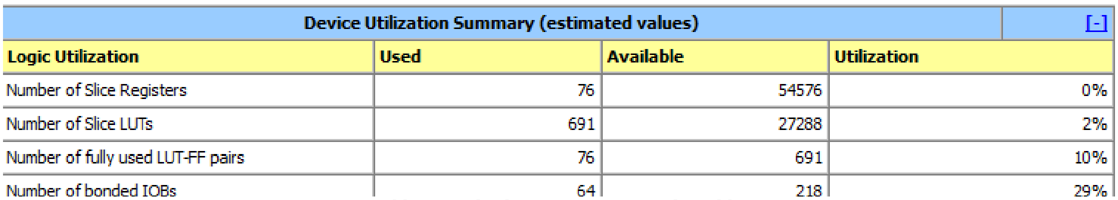
A 32-bit multiplier is designed using Verilog HDL and simulated using ISIM simulator. Fig.2 shows the simulated waveforms which are generated from Evaluation unit, intermediate partial products and the final result. It is designed in Verilog HDL, synthesized and implemented using Xilinx 14.7 design suite with specifications of spartan6(XC6SLX45-CSG324).



A[31:0]	00000000	99999999	99968999
B[31:0]	00000000	99999999	12345678
M1[35:0]	000000000	099999999	099968999
M2[35:0]	000000000	199999998	199937998
M3[35:0]	000000000	299999997	299906997
M4[35:0]	000000000	399999996	399875996
M5[35:0]	000000000	499999995	499844995
P1[35:0]	000000000	099999999	199937998
P2[35:0]	000000000	099999999	299906997
P3[35:0]	000000000	099999999	399875996
P4[35:0]	000000000	099999999	499844995
P5[35:0]	000000000	099999999	399875996
P6[35:0]	000000000	099999999	299906997
P7[35:0]	000000000	099999999	199937998
P8[35:0]	000000000	099999999	099968999
PP1[35:0]	000000000	899999991	799751992
PP2[35:0]	000000000	899999991	699782993
PP3[35:0]	000000000	899999991	599813994
PP4[35:0]	000000000	899999991	499844995
PP5[35:0]	000000000	899999991	399875996
PP6[35:0]	000000000	899999991	299906997
PP7[35:0]	000000000	899999991	199937998
PP8[35:0]	000000000	899999991	099968999
P[63:0]	000000000000...	9999999800000001	1234185071636322

Fig.2. Simulation Results of 32X32-bit Multiplication

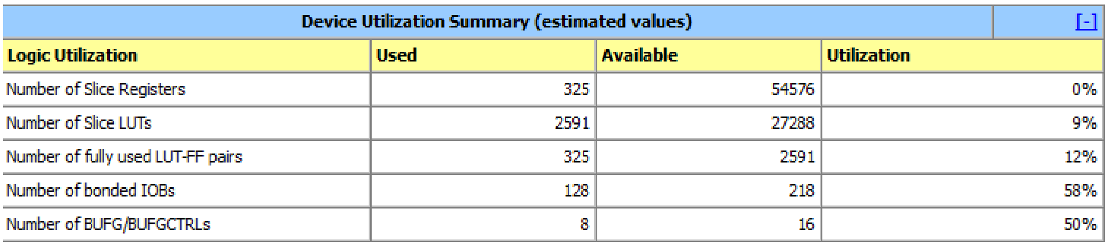
In figure 2, A is 4 digit decimal value whose multiplicand multiples are calculated. M1, M2, M3, M4, M5 represents multiplicand multiples. A and B represents the multiplicand and multiplier respectively.P1,P2…P7,P8, PP1, PP2…PP7 and PP8 represents partial products and P represents final product. Table 1,Table 2 and Table 3 represent the device utilization summary for Regular adder, Modified adder and Reduced delay BCD adder respectively.



Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	76	54576	0%
Number of Slice LUTs	691	27288	2%
Number of fully used LUT-FF pairs	76	691	10%
Number of bonded IOBs	64	218	29%

Table 1. Multiplication using Regular adder

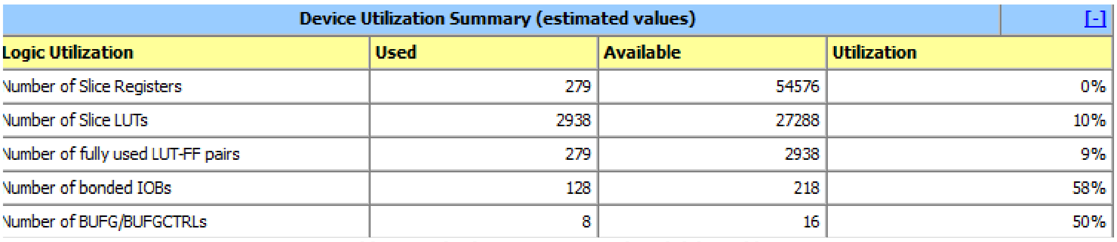
The Device Utilization Summary allows us to quickly access design overview information, number of LUTs and Slice Register.Maximum Combinational path delay observed is 54.293ns for Regular adder.



Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	325	54576	0%
Number of Slice LUTs	2591	27288	9%
Number of fully used LUT-FF pairs	325	2591	12%
Number of bonded IOBs	128	218	58%
Number of BUFG/BUFGCTRLs	8	16	50%

Table 2:.Multiplication using Modified adder

Maximum Combinational path delay observed is 51.962ns for Modified adder.



Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	279	54576	0%
Number of Slice LUTs	2938	27288	10%
Number of fully used LUT-FF pairs	279	2938	9%
Number of bonded IOBs	128	218	58%
Number of BUFG/BUFGCTRLs	8	16	50%

Table 3. Multiplication using Reduced delay adder

Maximum Combinational path delay observed is 41.483ns for Reduced delay adder. Table 4 shows the Performance analysis of parallel decimal multiplier in terms of maximum combinational path delay and number of LUTs. By compare the multiplication delays, Reduced delay adder has given the minimum path delay.

Parameters	Regular adder	Modified adder	Reduced delay adder
Maximum Combinational path delays(in ns)	54.293	51.962	41.483
Device Utilization summary(number of LUTs)	2215	2591	2938

Table 4: Performance analysis of 32X32 bit multiplier

V. CONCLUSION

We have designed the parallel decimal multiplier by using three adders. The Reduced delayadder (41.483ns) gives less delay as compared to Regular adder(54.293ns)and Modified adder(51.962). A 23.6% reduction for Reduceddelay adder as compared toRegular adder and 20.2% reduction forReduced delay adder as compared toModified adder is observed from the synthesis results.

REFERENCES

1. Ch.Tejasri and B.NagaJyothi, 'Design and Performance Comparison of High Speed Parallel BCD Multiplier', International Journal Of Innovative Research In Science, Engineering And Technology, Vol. 6,Issue 4, pp.5703-5707, April 2017.
2. T. Lang and A. Nannarelli, 'A Radix-10 Combinational Multiplier', Proc. 40th Asilomar Conf. Signals, Systems, and Computers, pp. 313-317, Oct. 2006.
3. M. Kaivani A, Liu Han and Seok -Bum Ko,'Improved design of high- frequency sequential decimal multiplier', Electronics Letters, Vol. 50 No. 7, pp. 558–560,March 2014.
4. Schmookler .M and Weinberger. A, 'High Speed Decimal Addition', IEEE Trans. Computers, Vol. 20, No. 8, pp. 862-866, Aug. 1971.
5. L. Dadda, 'Multioperand parallel decimal adder: A mixed binary and BCD approach', IEEE Trans. Compute., Vol. 56, No. 10, pp. 1320–1328, Oct. 2007.
6. Erle, M.A., Schwarz, E.M., and Schulte, M.J.,'Decimal multiplication with efficient partial product generation', Proc.*17th IEEE Symp. ComputerArithmetic*, USA, pp. 21–28, June 2005.
7. R.D. Kenney and M.J. Schulte, 'High-Speed MultioperandDecimal Adders', IEEE Trans. Computers, Vol. 54, No. 8, pp.953- 963, Aug. 2005.

DOI: 10.15680/IJIRCCE.2017. 0505055