

High Speed and Low Power Dual Constant Multiplier

¹J. PRAVEEN, ²N. ARAVIND

¹DEPT. OF E.C.E, HYDERABAD, TELANGANA, INDIA

² ASSISTANT PROFESSOR, TEEGALA KRISHNA REDDY ENGINEERING COLLEGE, TELANGANA, INDIA

ABSTRACT: To avoid the problem of dual constant multiplication radix 2^r arithmetic is applied. Consider a number of M non negative constants having bit length N. To form the critical path we are going to determine the formulas for maximum number of additions, average number of additions and maximum number of cascaded additions. The problems that are predictable with problem size (M, N) is solved by the dual constant multiplication radix 2^r approach. This radix 2^r approach gives the sub linear routine complexity $O(M \times N/r)$ where r is the function of (M,N), compared to other published DCM algorithms this algorithm gives the shortest path in addition. In case high complexity problems are occurred it is solved by this radix 2^r DCM algorithms. At last by using the dual constant multiplication radix 2^r algorithm the power is low power is consumed and high speed is obtained.

KEY TERMS: dual constant multiplication, radix 2^r , cascaded additions.

1.INTRODUCTION

Multiple constant multiplication is an arithmetic operation which consists of set of multiples. This set of multiples consists of set of fixed point constants given as $\{C_0, C_1, C_2, \dots, C_{M-1}\}$. This constants consists of same fixed point variable and it is represented as X. The entire operation is based on the complexity of the transformations such as FIR, IIR, DSP transforms, crypto systems, LTI controllers. For the purpose of effective implementation MCM should avoid the rapid, compact and low power. To avoid this we will use costly multipliers. Coming to the hardware implementation we will use the multiplier less connections by using the additions, subtractions and left shifts. Coming to the addition and subtraction it has same area and speed cost but shifts are costless compared to both addition and subtraction. Here additions, subtractions and shifts are realized by the logic gates. So from this we can define that the MCM problem is the process of finding the minimum number of addition/subtraction operations. But the complexity of this process is very hard because the solutions that are obtained in this are very huge and optimal.

At last the multiple constant multiplication approach gives reasonable amount of time and sub optimal solutions. Up to now we have discussed about the MCM the improved version of this approach is RADIX 2^r SCM. In this compared to MCM the determination of the formula for adder-cost (Upb), average (Avg), and adder-depth (Ath) are low. In this the power will be consumed depending upon the value of N if the value of N increases then the power will be saved more.

Radix 2^r SCM algorithm gives the sub linear routine complexity with respect to N. In SCM radix 2^r algorithm the space required is very small. Basically there are two features of radix 2^r SCM based algorithm they are it consists of huge constants and lowest runtime complexity with non digit recording algorithm. Not only this one the more features of radix 2^r SCM is shown in below table 1.

Metrics	Equations
Adder cost	$Upb(r) = \left\lceil \frac{N+1}{r} \right\rceil + 2^{r-2} - 2$ with $r=r_1$ or $r=r_2$
Adder depth	$Ath(r) = \left\lceil \frac{N+1}{r} \right\rceil + r - 3$ with $r=r_1$ or $r=r_2$
Average	$-1 + Avg_{pp} + Avg_{om} \leq Avg(r) \leq -2 + Avg_{pp} + 2^{r-2}$ with $Avg_{pp} = (1 - 2^{-r}) \times \left\lceil \frac{N+1}{r} \right\rceil$ $Avg_{om} = \sum_{j=0}^{\left\lceil \frac{N+1}{r} \right\rceil - 1} \left\{ \sum_{k=1}^{2^{r-1}-1} P(m_{jk}) \times [1 - P(m_{jk})]^j \right\}$ $P(m_{jk}) = \frac{\log_2 \left[\frac{2^{r-1}}{2 \times k + 1} \right]}{2^{r-1}}$ and $r=r_1$ or $r=r_2$
	$r_1 = 2 \cdot W \left[\sqrt{(N+1) \cdot \log_2(2)} \right] / \log_2(2)$ $r_2 = W \left[4 \cdot (N+1) \cdot \log_2(2) \right] / \log_2(2)$

TABLE. 1 RADIX- 2^r SCM

The radix 2^r SCM is the base number system that is properly given to bit length N of constant to obtain an optimal adder cost r_1 and lower adder depth r_2 . Depending upon the design requirement the expression of r is chosen. r_1 is used when area is targeted and r_2 is used when speed and power is concerned. The main purpose of this process is to first apply the arithmetic operation to the MCM problem. The determination of the formula for adder-cost (Upb), average (Avg), and adder-depth (Ath) are low this one is similar to the determination of the SCM. Next is to implementation of actual circuit by

using the application of radix 2^f MCM optimization of bench mark FIR filter.

II. RADIX 2^f MCM

The improved version of radix 2^f SCM is radix 2^f MCM. In this a single variable X is multiplied with the set of M constants $\{C_0, C_1, C_2, \dots, C_{M-1}\}$ having equal bit size N. Here the each bit constant C_i is given as $[(N+1)/r^1]$ as well as bit length is given as r^1+1 . The partial production generation is obtained by adding the maximum number of partial products and maximum of $2^{f-2}-1$ non trivial partial products. The below figure (1) shows the partitioning of n bit constants of radix $2r$ SCM and radix $2r$ MCM.

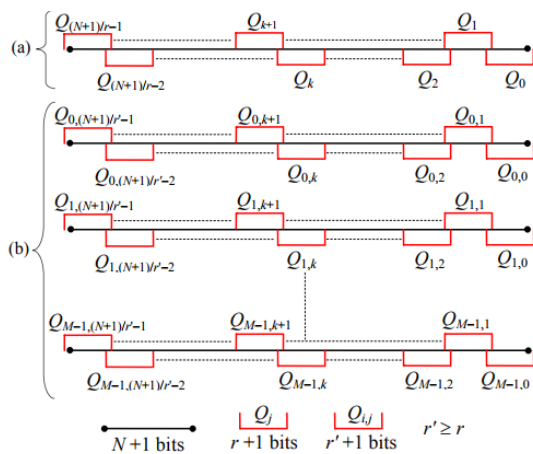


FIG. 1 SLICE PARTITIONING OF N-BIT CONSTANTS IN RADIX-2R. (A) RADIX-2R SCM, (B) RADIX-2R MCM.

The analytical expression for adder and the adder depth is shown in above figure (1). Here in this the bit size is same but coming to the applications of the MCM the transposed form of the FIR filter coefficients will have different bit size. Let us consider that the constant C_i having bit size N_i , so from this the number of partial products for a set of M constants is given as $\sum[(N+1)/r^1]$.

III. DUAL CONSTANT LOGIC

Basically the dual logic constant gives the high performance and less delay compared to the other multipliers. This is shown in below figure(2). It is depend upon the logical operations that are executed. It is divided into three layers having particular bit size. In case if it consists of 2 bit size then there will be 3 parts and for 4 bit size there will be 5 parts. Depending upon the bit size parts will be doubled in the architecture. The important thing is that the entire architecture depends up on the third layer because the third layer is depend upon the two operations. Here

the one level of operation is performed next to the other. In this third layer the both AND and XOR operations are performed. In the AND gate one gate is used to perform the operation coming to the XOR gate five gates are used to perform the operation. So first preference in this architecture is given to the AND gate operation and second preference is given to the XOR operation. so from this we can say that in this entire operation the both AND and XOR operations plays very important role.

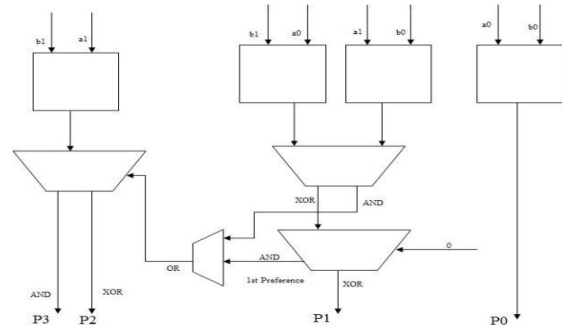


FIG. 2 2 X 2 ARCHITECTURE OF DUAL CONSTANT LOGIC

From the below flow chart we can observe that it processes by taking two input values. This two input values are used for the purpose of reading which is known as multiplicand. Now the input registers reads the values of both multiplicand and the multiplier. These are passed through the dual logic level multiplier.

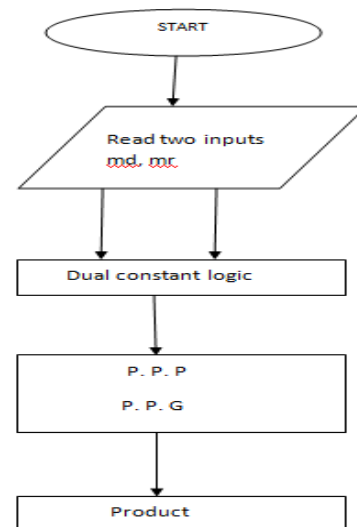


FIG. 3 FLOW CHART OF DUAL CONSTANT LOGIC

As studied earlier that in the dual logic level multiplier two operations will be performed. One is AND operation and the another is XOR operation. Now, The product will be selected by the logic family after the completion of the multiplication process in the multiplier. The entire operation is seen in below figure. 3.

IV. RESULTS

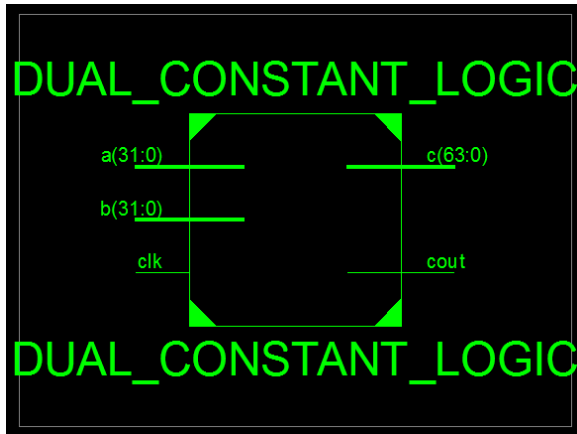


FIG. 4. RTL SCHEMATIC

The below figure (4) shows the schematic view of the RTL. From the figure (5) we can observe the absolute values of the DCM.

Device Utilization Summary (estimated values)		
Logic Utilization	Used	Available
Number of Slice LUTs	116	46560
Number of fully used LUT-FF pairs	0	116
Number of bonded IOBs	130	240

FIG. 5. SUMMARY REPORT OF DCM

From figure (6) we can observe the technology view of the DCM and figure (7) shows the output of DCM.

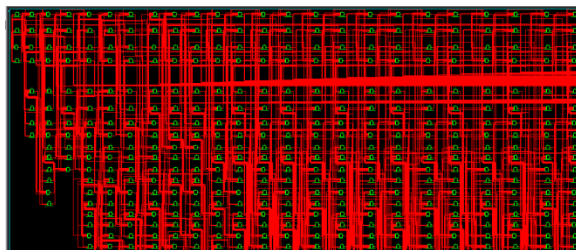


FIG. 6. TECHNOLOGY VIEW OF THE DCM

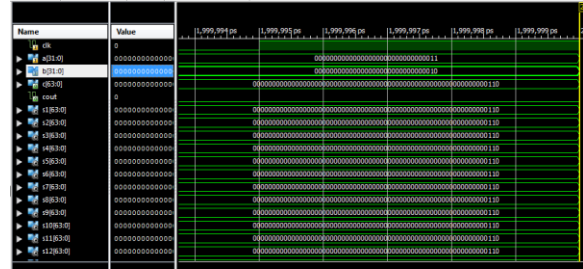


FIG. 7. OUTPUT OF DCM

V. CONCLUSION

As discussed earlier that to avoid the problem of dual constant multiplication first radix 2^f arithmetic is used. After for effective implementation we use the SCM based radix 2^f is used. For more effective implementation we uses the radix 2^f DCM is used. In this the DCM uses costly multipliers for low power consumption and high speed. so compared to others it is highly predictable and shorter adder path is developed. At last it gives exact values for the adder cost and adder depth bounds.

VI. REFERENCES

- [1] C. F. Kerry, "Digital signature standard (DSS)," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, FIPS PUB 186-4, 2013.
- [2] IEEE Standard Specifications for Public-Key Cryptography, IEEE Standard 1363-2000, Aug. 2000, pp. 1–228.
- [3] H. Fan and Y. Dai, "Fast bit-parallel GF(2n) multiplier for all trinomials," IEEE Trans. Comput., vol. 54, no. 4, pp. 485–490, Apr. 2005.
- [4] A. Cilardo, "Fast parallel GF(2m) polynomial multiplication for all degrees," IEEE Trans. Comput., vol. 62, no. 5, pp. 929–943, May 2013.
- [5] T. Beth and D. Gollman, "Algorithm engineering for public key algorithms," IEEE J. Sel. Areas Commun., vol. 7, no. 4, pp. 458–466, May 1989.
- [6] L. Song and K. K. Parhi, "Efficient finite field serial/parallel multiplication," in Proc. Int. Conf. Appl. Specific Syst., Archit. Processors (ASAP), Aug. 1996, pp. 72–82.
- [7] M. Nikooghadam and A. Zakerolhosseini, "Utilization of pipeline technique in AOP based multipliers with parallel inputs," J. Signal Process. Syst., vol. 72, no. 1, pp. 57–62, Jul. 2013.
- [8] B. Sunar and C. K. Koc, "Mastrovito multiplier for all trinomials," IEEE Trans. Comput., vol. 48, no. 5, pp. 522–527, May 1999.
- [9] H. Wu, "Bit-parallel finite field multiplier and squarer using polynomial basis," IEEE Trans. Comput., vol. 51, no. 7, pp. 750–758, Jul. 2002.
- [10] S.-M. Park, K.-Y. Chang, D. Hong, and C. Seo, "New efficient bit parallel polynomial basis

multiplier for special pentanomials,” Integr., VLSI J., vol. 47, no. 1, pp. 130–139, Jan. 2014.

[11] Y. Li and Y. Chen, “New bit-parallel Montgomery multiplier for trinomials using squaring operation,” Integr., VLSI J., vol. 52, pp. 142–155, Jan. 2016.

[12] A. G. Wassal, M. A. Hassan, and M. I. Elmasry, “Low-power design of finite field multipliers for wireless applications,” in Proc. 8th Great Lakes Symp. VLSI (GLSVLSI), Feb. 1998, pp. 19–25.

[13] A. Zakerolhosseini and M. Nikooghadam, “Low-power and high-speed design of a versatile bit-serial multiplier in finite fields GF(2^m),” Integr., VLSI J., vol. 46, no. 2, pp. 211–217, Mar. 2013.

[14] J. Grossschadl, “A low-power bit-serial multiplier for finite fields GF(2^m),” in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), vol. 4, May 2001, pp. 37–40.

[15] L. Gao and K. K. Parhi, “Custom VLSI design of efficient low latency and low power finite field multiplier for Reed–Solomon codec,” in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), vol. 4, May 2001, pp. 574–577.



J. PRAVEEN, Completed M.Tech degree in VLSI System Design from Briilaint Institute Of Engineering Techonolgy, Jntu, Hyderabad. received the b.tech degree in Electronics And Communication Engineering from Teegala Krishna Reddy College Of Engineering And Techonolgy, Jntu, Hyderabad, in 2010, His area of interest is Low Power V.L.S.I, and Digital communication.



N. ARAVIND working as Assistant Professor in Teegala Krishna Reddy Engineering College. Completed m.tech degree in VLSI system design from CVSR College Of Engineering, Jntu, Hyderabad, received the B.Tech degree in Electronics And Communication Engineering from SLCS Institute Of Engineering And Technology, Jntu, Hyderabad, in 2010, research interests include Digital Communication, Wireless Communication And Frontend Logic Design And Hardware/Software Co-Frication.