Spvryan's International Journal of Engineering Sciences & Technology (SEST)
ISSN : 2394-0905

Design of Algorithm and Architecture of FIR Filter

SONAWANE SARIKA RAMESH sonsarika@gmail.com,mb.no.09422703522

DR.S.T. GANDHE, PHD,PRINICPAL OF SITRC NASIK, mail.id:principal@sitrc.org

PROF. G.M. PHADE,PHD PERCUVING, HOD E&TC SITRC NASIK, mail id:gayatri..phade@sitrc.org

PROF.P.A.DHULEKAR,PHD SCHOLLER, ME COORDINATOR SITRC NASIK, mail id:pravin.dhulekar@sitrc.org

***Abstract* : From many decades, many efficient algorithms and architectures have been introduced for the design of low complexity bit-parallel multiple constant multiplications (MCM) operation which dominates the complexity of many digital signal processing systems. On the other hand, little attention has been given to the digit-serial MCM design that offers alternative low complexity MCM operations albeit at the cost of an increased delay. In this paper, we address the problem of optimizing the gate-level area in digit-serial MCM designs and introduce high level synthesis algorithms, design architectures. Experimental results show the efficiency of the proposed optimization algorithms and of the digit-serial MCM architectures in the design of digit-serial MCM operations and finite impulse response filters.**

***Index Terms*—0–1 integer linear programming (ILP), digit-serial arithmetic, finite impulse response (FIR) filters, gate-level area optimization, multiple constant multiplications.**

I. INTRODUCTION

Finite impulse response (FIR) filters are of great importance in digital signal processing (DSP) systems since their characteristics in linear-phase and feed-forward implementations make them very useful for building stable high-performance filters. Although both architectures have similar complexity in hardware, the transposed form is generally preferred because of its higher performance and power efficiency. The multiplier block of the digital FIR filter in its transposed where the multiplication of filter coefficients with the filter input is realized, has significant impact on the complexity and performance of the design because a large number of constant multiplications are required. This is generally known as the multiple constant multiplications (MCM) operation and is also a central operation and performance bottleneck in many other DSP systems such as fast Fourier transforms, discrete cosine transforms (DCTs), and error-correcting codes. Although area-, delay-, and power-efficient multiplier architectures, such as Wallace and modified Booth multipliers, have been proposed, the full flexibility of a multiplier is not necessary for the constant multiplications, since filter coefficients are fixed and determined beforehand by the DSP algorithms. Hence, the multiplication of filter coefficients with the input data is generally implemented under a shift addition architecture where each constant multiplication is realized using addition/subtraction and shift operations in an MCM operation For the shift-adds implementation of constant multiplications, a straightforward method, generally known as digit based rn at the gate level. In this paper, we initially determine the gate-level implementation costs of digit-serial addition, subtraction, and left shift operations used in the shift-adds design of digit-serial MCM operations. Then, we introduce the exact CSE algorithm that formalizes the gate-level area optimization problem as a 0–1 integer linear programming (ILP) problem when constants are defined under a particular number representation. We also present a new optimization model that reduces the 0–1 ILP problem size significantly and, consequently, the runtime of a generic 0–1 ILP solver. Since there are still instances which the exact CSE algorithm cannot handle, we describe the approximate GB algorithm that iteratively finds the "best" partial product which leads to the optimal area in digit-serial MCM design at the gate level. This paper also introduces a computer-aided design (CAD) tool called SAFIR which generates the hardware descriptions of digit-serial MCM operations and FIR filters based on a design architecture and implements these circuits using a commercial logic synthesis tool. In SAFIR, the digit-serial constant multiplications can be implemented under the shift adds architecture, and also can be designed using generic digit serial constant multipliers. Experimental results on a comprehensive set of instances show that the solutions of algorithms introduced in this paper lead to significant improvements in area of digit-serial MCM designs compared to those obtained using the algorithms designed for the MCM problem. The digit-serial FIR filter designs obtained by SAFIR also indicate that the realization of the multiplier block of a digit-serial FIR filter under the shift adds architecture significantly reduces the area of digit-serial FIR filters with respect to those designed using digit-serial constant multipliers Additionally, it is observed that the optimal tradeoff between area and delay in digit-serial FIR filter designs can be explored by changing the digit size d. In the last two decades, many efficient algorithms and architectures have been introduced for the design of low complexity bit-parallel multiple constant multiplications (MCM) operation which dominates the complexity of many digital signal processing systems. On the other hand, little attention has been given to the digit-serial MCM design that offers alternative low complexity MCM operations albeit at the cost of an increased delay. In this paper, we address the problem of optimizing the gate-level area in digit-serial MCM designs and introduce high level synthesis algorithms, design architectures by using Verilog HDL and modelsim.

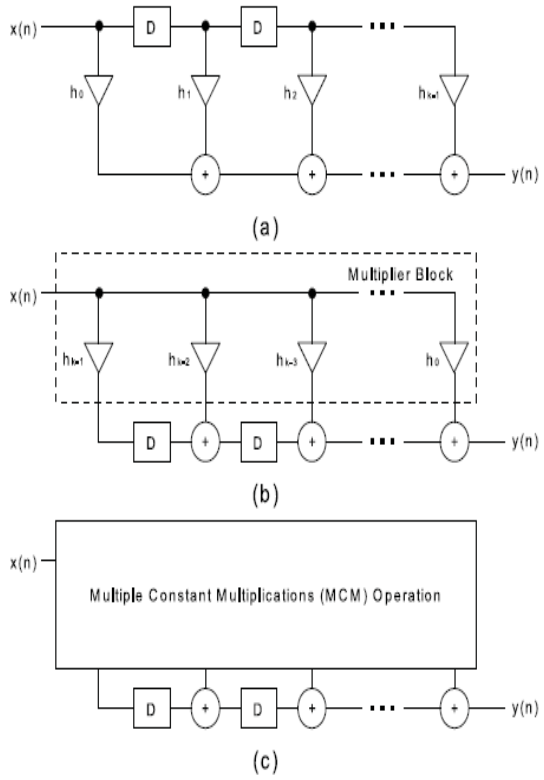

Fig.1. FIR filters implementations. (a) Direct form. (b) Transposed form with generic multipliers. (c) Transposed form with an MCM block.

A. MOTIVATION

As the scale of integration keeps growing, more and more sophisticated signal processing systems are being implemented on a VLSI chip.These signal processing applications not only demand great computation capacity but also consume considerable amount of energy. While performance and Area remain to be the two major design tolls, power consumption has become a critical concern in today's VLSI system design. The need for low-power VLSI system arises from two main forces. First, with the steady growth of operating frequency and processing capacity per chip, large currents have to be delivered and the heat due to large power consumption must be removed by proper cooling techniques. Second, battery life in portable electronic devices is limited. Low power design directly leads to prolonged operation time in these portable devices. Addition usually impacts widely the overall performance of digital systems and a crucial arithmetic function. In electronic applications adders are most widely used. Applications where these are used are multipliers, DSP to execute various algorithms like FFT, FIR and IIR. Wherever concept of multiplication comes adders come in to the picture. As we know millions of instructions per second are performed in microprocessors. So, speed of operation is the most important constraint to be considered while designing multipliers. Due to device portability miniaturization of device should be high and power consumption should be low. Devices like Mobile, Laptops etc. require more battery backup.

So, a VLSI designer has to optimize these three parameters in a design. These constraints are very difficult to achieve so depending on demand or application some compromise between constraints has to be made. Ripple carry adders exhibits the most compact design but the slowest in speed. Whereas carry look ahead is the fastest one but consumes more area. Carry select adders act as a compromise between the two adders. In 2002, a new concept of hybrid adders is presented to speed up addition process by Wang et al. that gives hybrid carry look-ahead/carry select adders design. In 2008, low power multipliers based on new hybrid full adders is presented.

DESIGN of area- and power-efficient high-speed data path logic systems are one of the most substantial areas of research in VLSI system design. In digital adders, the speed of addition is limited by the time required to propagate a carry through the adder. The sum for each bit position in an elementary adder is generated sequentially only after the previous bit position has been summed and a carry propagated into the next position.The CSLA is used in many computational systems to alleviate the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum.

B. OBJECTIVE

Main objective is to eliminate multiplier block and introducing MCM architecture in digit serial FIR filter for the reduction of multiplication in the form of shift and add operations

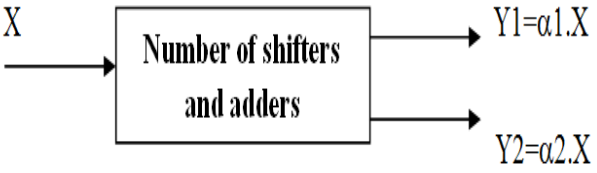

Fig.2 X denotes input, α1 and α2 are filter coefficients, Y1 and Y2 are the outputs

In this paper, we initially determine the gate-level implementation costs of digit-serial addition, subtraction, and left shift operations used in the shift-adds design of digit-serial MCM operations. Then, we introduce the exact CSE algorithm that formalizes the gate-level area optimization problem as a 0–1 integer linear programming (ILP) problem when constants are defined under a particular number representation. We also present a new optimization model that reduces the 0–1

ILP problem size significantly and, consequently, the runtime of a generic 0–1 ILP solver. Since there are still instances which the exact CSE algorithm cannot handle, we describe the approximate GB algorithm that iteratively finds the "best" partial product which leads to the optimal area in digit-serial MCM design at the gate level. This paper also introduces a computer-aided design (CAD) tool called SAFIR which generates the hardware descriptions of digit-serial MCM operations and FIR filters based on a design architecture and implements these circuits using a commercial logic synthesis tool. In SAFIR, the digit-serial constant multiplications can be implemented under the shift adds architecture, and also can be designed using generic digit serial constant multipliers.

C. EXISTING SYSTEM

Multiple constant multiplication (MCM) constitutes a typical fixed-point arithmetic operation in digital signal Processing. It is the focus of a lot of research on high-speed and low power devices in communication systems and signal processing systems. In multiplier less MCM, multipliers are replaced by simpler components such as adders and hard-wired shifts (adders in our paper include also subtractors as their hardware costs are similar). By using the
Negative digits (subractor in circuit) in their signed-digit representations, coefficients may be synthesized with fewer adders; therefore the area and power consumption of the circuit can be reduced. An example of a multiplier-based and a multiplier less based MCM implementations respectively, wherein 4 multiplications are replaced by 6 adders and 6 hard-wired shifts. Such
Multiplier less MCMs are utilized, for example, in the design of finite-impulse response Filters.

II. PROPOSED SYSTEM ALGORITHM:CSE AND GB ALGORITHM

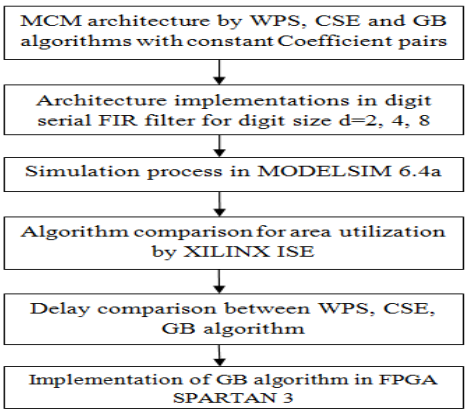

Fig 4.1 flow map for proposed system design

A. CSE ALGORITHM

An algorithm for efficient solution of the multiple constant multiplication problem. Common sub expression elimination (CSE) as a way to tackle the MCM problem was already proposed by various authors primarily as a possible method for the optimization of finite-duration impulse response (FIR) filter area through the reduction of the multiplier block logic a number of other applications in which the MCM transformation can be successfully applied were proposed. In this work, we will introduce an algorithm able to solve the CSE problem in an efficient way. The idea of CSE can be demonstrated on a FIR filter design. The optimization procedure targets the minimization of the multiplier block area. After expressing the coefficients in a canonical signed digit (CSD) format in order to reduce the total number of nonzero bits (thus also the additions/subtractions necessary), an add shift expansion is performed. The goal of CSE is to identify the bit patterns that are present in the coefficient set more than once. Since it is sufficient to implement the calculation of the multiple identical expressions only once, the resources necessary for these operations can be shared.

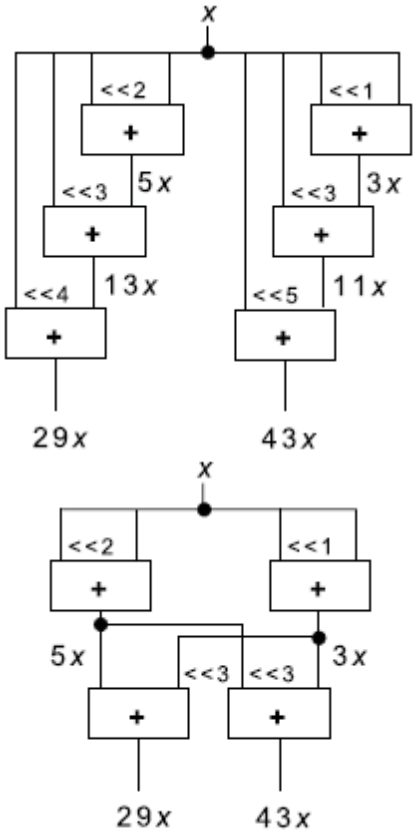

Fig 4.2 WSE and CSE Algorithm

B. EXACT GB ALGORITHM

The optimization of gate-level area problem in digit-serial MCM design is an NP-complete problem due to the NP-completeness of the MCM problem. Thus, naturally, there will be always 0–1 ILP problems generated by the exact CSE algorithm that current 0–1 ILP solvers find difficult to handle. Hence, the GB heuristic algorithms, which obtain a good solution using less computational resources, are indispensable. In our approximate algorithm called MINAS-DS, as done in algorithms designed for the MCM problem given in Definition 1, we find the fewest number of intermediate constants such that all the target and intermediate constants are synthesized using a single operation. However, while selecting an intermediate constant for the implementation of the not yet synthesized target constants in each iteration, we favor the one among the possible intermediate constants that can be synthesized using the least hardware and will enable us to implement the not-yet synthesized target constants in a smaller area with the available constants. After the set of target and intermediate constants that realizes the MCM operation is found, each constant is synthesized using an A-operation that yields the minimum area in the digit-serial MCM design.the area of the digit-serial MCM operation is determined as the total gate-level implementation cost of each digit-serial addition, subtraction, and shift operation under the digit size parameter d as described in Section II-D. The preprocessing phase of the algorithm is the same as that of the exact CSE algorithm, and its main part and routines are given. The right shift of an A-operation is assumed to be zero.

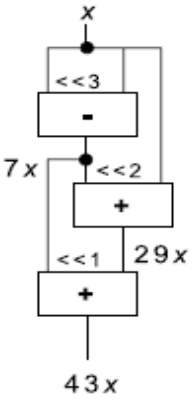

Fig 4.3 GB Algorithm

III. EXPERIMENTAL RESULTS

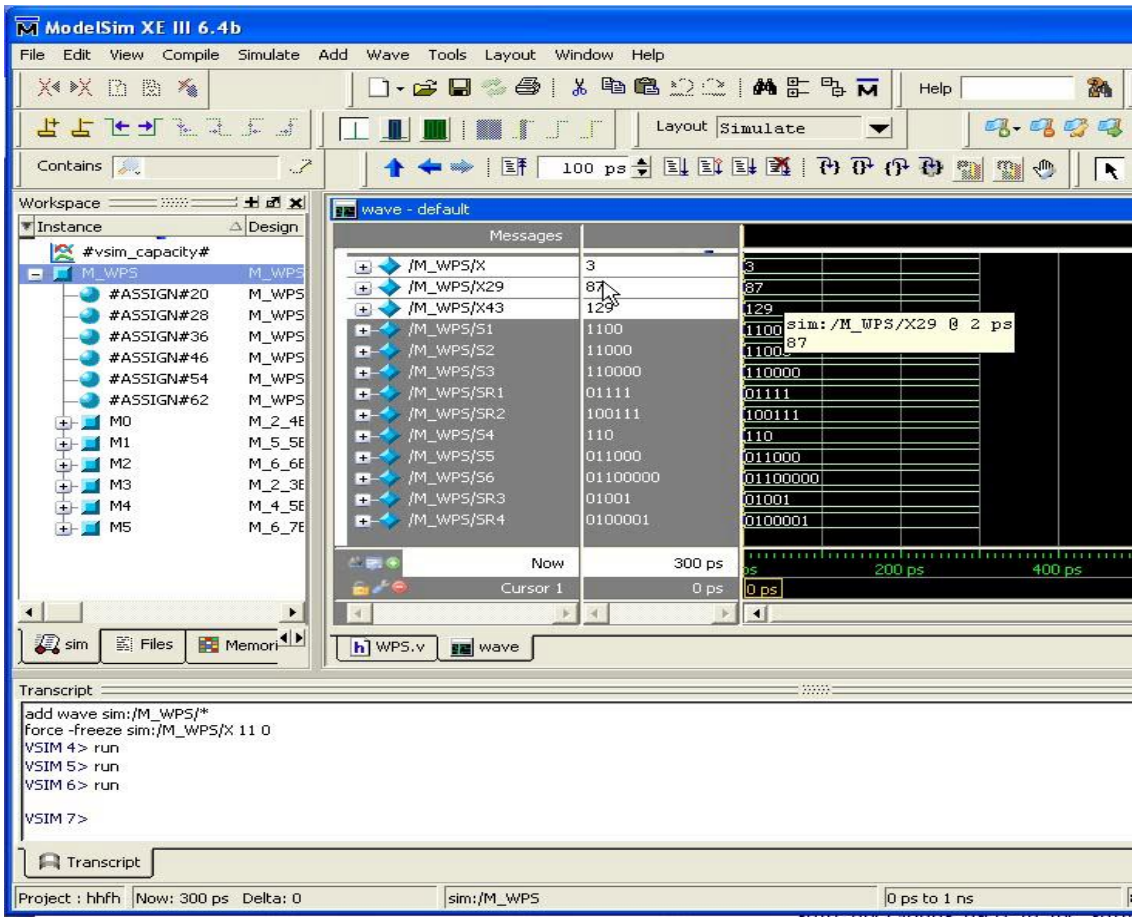

Fig 5.1 Output of WPS Algorithm

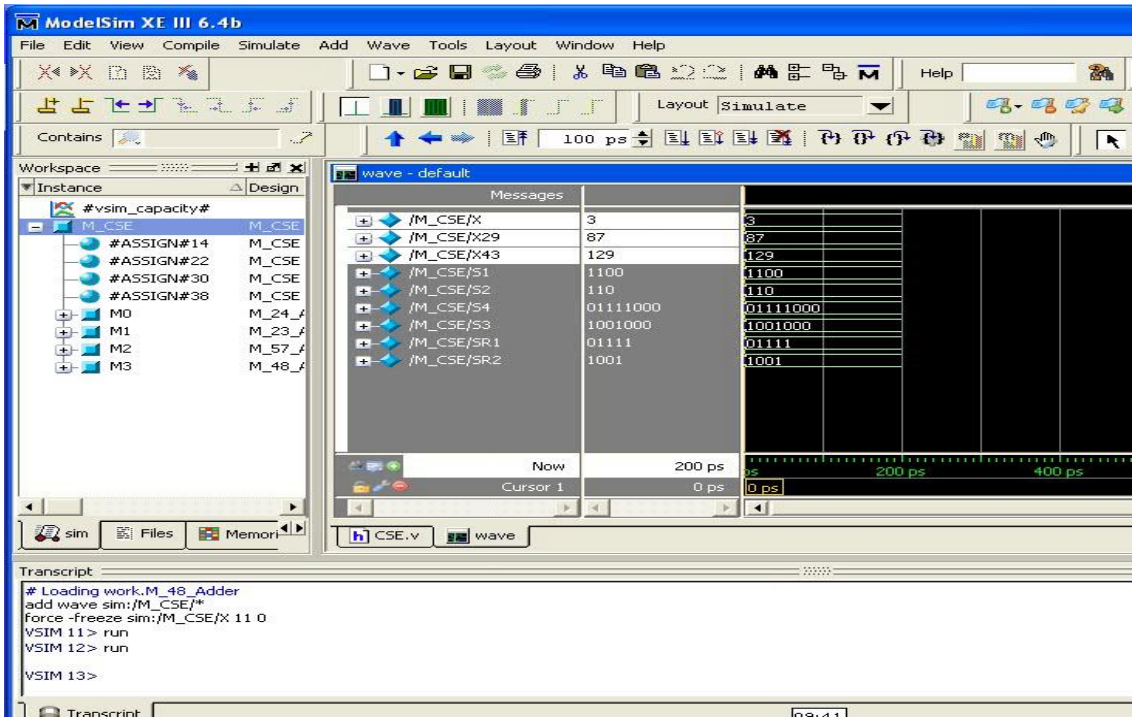

Fig 5.2 Output of CSE Algorithm

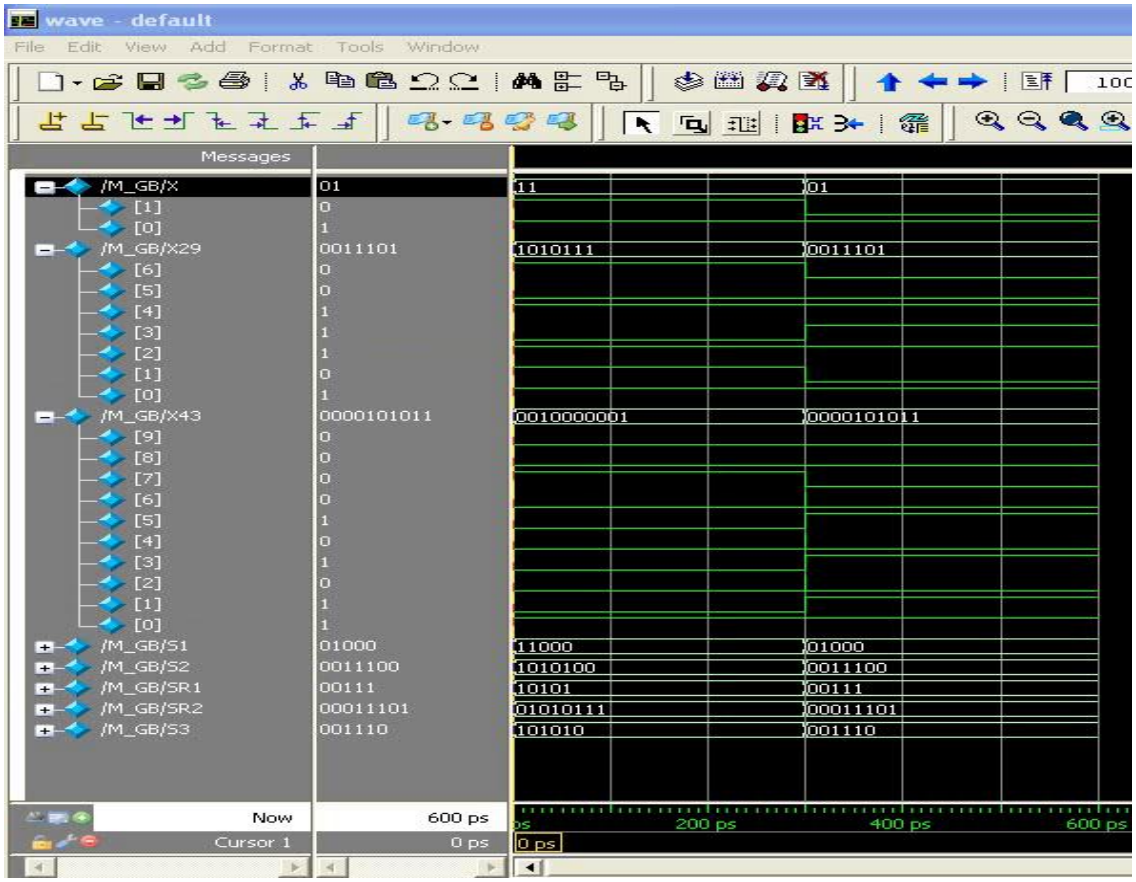

Fig 5.3 Output of GB Algorithm

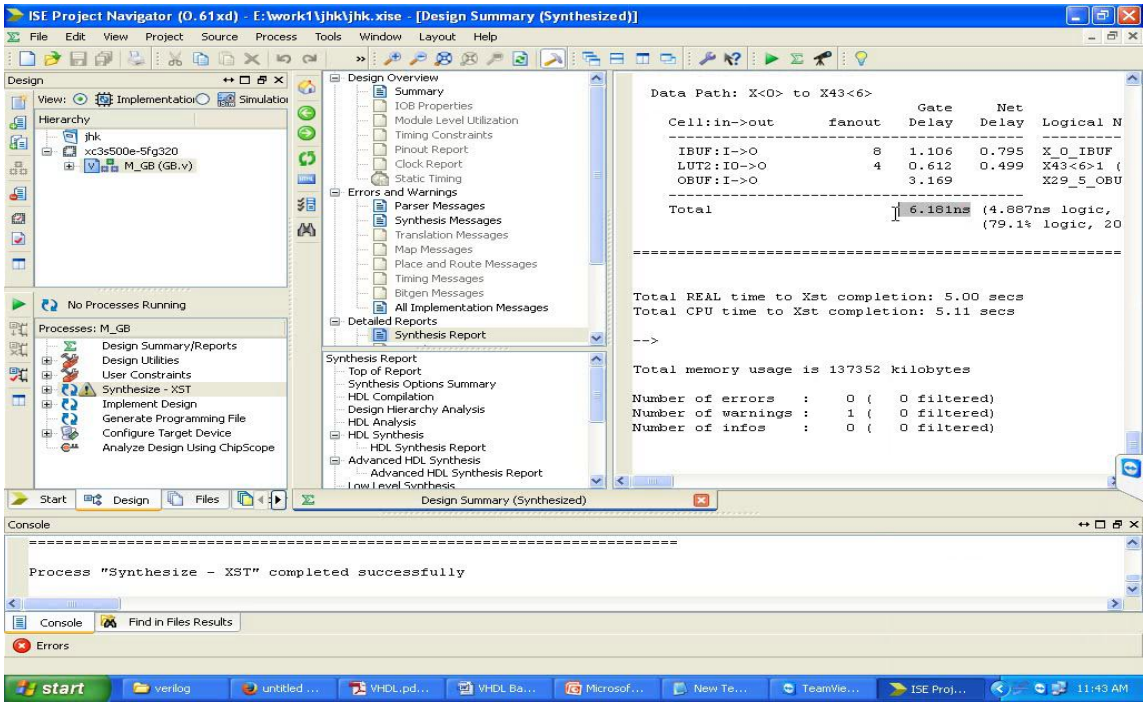

Fig 5.4 Output of XILINX Algorithm

IV. CONCLUSION

In this paper, we introduced the 0–1 ILP formalization for designing digit-serial MCM operation with optimal area at the gate level by considering the implementation costs of digit-serial addition, subtraction, and shift operations. Since there are still instances with which the exact CSE algorithm cannot cope, we also proposed an approximate GB algorithm that finds the best partial products in each iteration which yield the optimal gate-level area in digit-serial MCM design. This paper also introduced the design architectures for the digit-serial MCM operation and a CAD tool for the realization of digit-serial MCM operations and FIR filters.

REFERENCES:

[1] L. Wanhammar, *DSP Integrated Circuits*. New York: Academic, 1999.

[2] C. Wallace, "*A suggestion for a fast multiplier*," IEEE Trans. Electron. Comput., vol. 13, no. 1, pp. 14–17, Feb. 1964.

[3] W. Gallagher and E. Swartzlander, "*High radix booth multipliers using reduced area adder trees*" in Proc. Asilomar Conf. Signals, Syst. Comput., vol. 1. Pacific Grove, CA, Oct.–Nov. 1994, pp. 545–549.

[4] J. McClellan, T. Parks, and L. Rabiner, "*A computer program for designing optimum FIR linear phase digital filters*" IEEE Trans. Audio Electroacoust., vol. 21, no. 6, pp. 506–526, Dec.1973.

[5] H. Nguyen and A. Chatterjee, "*Number-splitting with shift-and-add decomposition for power and hardware optimization in linear DSP synthesis*" IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 8, no. 4, pp. 419–424, Aug. 2000.

[6] M. Ercegovac and T. Lang, Digital Arithmetic. San Mateo, CA: Morgan Kaufmann, 2003.

[7] R. Hartley, "*Subexpression sharing in filters using canonic signed digit multipliers*" IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 43, no. 10, pp. 677–688, Oct. 1996.