International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 10
June 2016

Design of Digit-Serial FIR Filters Using GB Algorithm

***N.MANASA** ****V.VIJAYA** *****G.BABU**

*M.TECH Dept of ECE, VAAGDEVI COLLEGE OF ENGINEERING

**Asso. prof Dept of ECE, VAAGDEVI COLLEGE OF ENGINEERING

***Asso. prof Dept of ECE, VAAGDEVI COLLEGE OF ENGINEERING

Abstract:

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. In this paper, we address the problem of optimizing the gate-level area and delay in digit-serial MCM designs, for that purpose we introduce high level CSE and GB algorithms. 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, CSE and GB algorithms, multiple constant multiplications.

1. 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 [1]. These are mainly consists of multiplication of a vector of input samples with a set of constant coefficients is known as MCM operations.

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 10
June 2016

Multiple constant multiplications (MCM) are typical operations in digital signal processing (DSP) as well as in the design of finite-impulse-response (FIR) filters, as shown in Fig.1 (a).

Multiplications are expensive in terms of area and power consumption, when implemented in hardware. The relative cost of an adder and a multiplier in hardware, depends on the adder and multiplier architectures. For example, a *k x k* array multiplier has k times the logic (and area) and twice the latency of the slowest ripple carry adder. Since the values of the coefficients are known beforehand, the full flexibility of a multiplier is not necessary, and it can be more efficiently implemented by converting it into a sequence of additions/subtractions and shift operations are shown in fig.1 (b).

For the shift-adds implementation of constant multiplications, a straightforward method, generally

known as digit based recoding [2], initially defines the constants in binary.

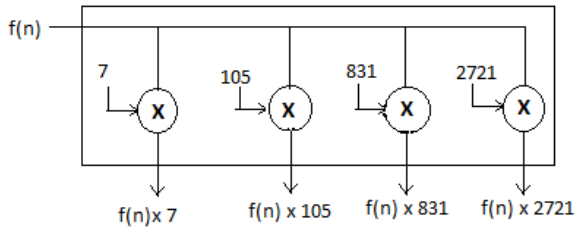

Figure 1(a): A multiplier-based MCM example

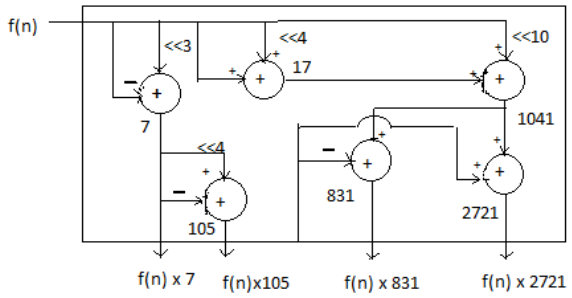

Figure 1(b): A multiplierless-based MCM example

Then, for each "1" in the binary representation of the constant, according to its bit position, it shifts the variable and adds up the shifted variables to obtain the result. As a simple example, consider the constant multiplications 29x and 43*x*. Their decompositions in binary are listed as follows:

$29x = (11101)_{bin}x = x<<4 + x<<3 + x<<2 + x$

$43x = (101011)_{bin}x = x<<5 + x<<3 + x<<1 + x$

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 10
June 2016

Which requires six addition operations as illustrated in Fig.2 (a)

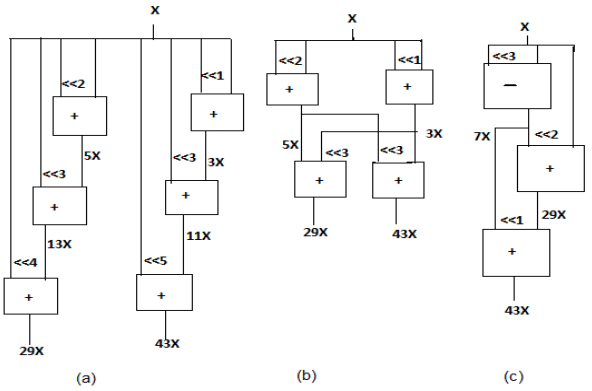

Fig. 2. Shift-adds implementations of 29*x* and 43*x*. (a) Without partial product sharing [2] and with partial product sharing. (b) Exact CSE algorithm [5]. (c) Exact GB algorithm [8].

The algorithms designed for the MCM problem can be categorized in two classes: common sub expression elimination (CSE) algorithms [3]–[5] and graph-based (GB) algorithm [6]–[8].

The proposed algorithm that optimally solves this maximal sharing problem. This problem has been the subject of extensive research in recent years. Two key strategies have had a large impact in the optimization of MCMs. One is to consider not only adders, but also subtracter to combine partial terms, thus increasing the opportunity for the sharing of common sub expressions.

The second is the usage of the Canonical Sign Digit (CSD) representation for the coefficients. This representation minimizes the number of non-zero digits; hence the maximal sub expression sharing search starts from a minimal level of complexity.

In a recent paper, Park [9] proposes the usage of a Minimal Signed Digit (MSD) representation for the coefficients. The MSD representation is obtained from the CSD representation by relaxing the requirement that there cannot be two consecutive non-zero digits. Under the MSD representation, a given numerical value can have multiple representations. However, in all of them, the number of non-zero digits is the same as the CSD representation. The algorithm proposed in [9] exploits the redundancy of the MSD representation by choosing the MSD instance that leads to a maximal sharing in the implementation of efficient FIR filters.

To the best of our knowledge, all previous solutions to this problem have been heuristic, providing no indication as to how far from the optimum their solution is. We

propose an exact algorithm that is feasible for many real situations. We model this problem as a Boolean network that covers all possible partial terms which may be used to generate the set of coefficients in the MCM instance. The inputs to this network are shifted versions of the value that serves as input to the MCM operation. Each adder and subtracter used to generate a given partial term is represented as an AND gate. All partial terms that represent the same numerical value are ORed together. There is a single output which is an AND over all the coefficients in the MCM. We cast this problem into a 0-1 Integer Linear Programming (ILP) problem by requiring: that the output is asserted,

meaning that all coefficients are covered by the set of partial terms found; while minimizing the total number of AND gates that evaluate to one, *i.e.*, the number of adders/subtracters.

We have applied this algorithm to coefficients represented in binary, CSD and MSD representations. Note that the redundancy of the MSD representation can be readily incorporated in our model, where the equivalent MSD representations are simply new inputs to the OR gate that generates a given coefficient.

Returning to our example in Fig. 2, the exact CSE algorithm of [9] gives a solution with four operations by finding the most common partial products $3x = (11)_{\text{bin}}x$ and $5x = (101)_{\text{bin}}x$ when constants are defined under binary, as illustrated in Fig. 2(b). On the other hand, the exact GB algorithm [6] finds a solution with the minimum number of operations by sharing the common partial product $7x$ in both multiplications, as shown in Fig. 2(c). Note that the partial product $7x = (111)_{\text{bin}}x$ cannot be extracted from the binary representation of $43x$ in the exact CSE algorithm [5].

2. Digit-Serial Arithmetic

In digit-serial arithmetic, data words are divided into digits, with a digit size of d bits, which are processed in one clock cycle. The special cases of the digit-serial computation, called bit-serial and bit-parallel processing, occur when the digit size d is equal to 1 and input data word length, respectively. The digit-serial computation plays an important

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 10
June 2016

role when the bit-serial implementations cannot meet delay requirements and the bit-parallel designs require excessive hardware. Thus, an optimal tradeoff between area and delay can be obtained by changing the digit size parameter (*d*). The fundamental digit-serial operations were introduced in [10]. The digit-serial addition, subtraction, and left shift operations are depicted in Figure 3 when *d* is equal to 3. Notice from Figure 3(a) that in a digit-serial addition operation, in general, the number of required full adders (FAs) is equal to *d* and the number of necessary D flip-flops is always 1. The subtraction operation (Figure 3(b)) is implemented using 2's complement, requiring the initialization of the D flip-flop with 1 and additional *d* inverter gates with respect to the digit-serial addition operation.

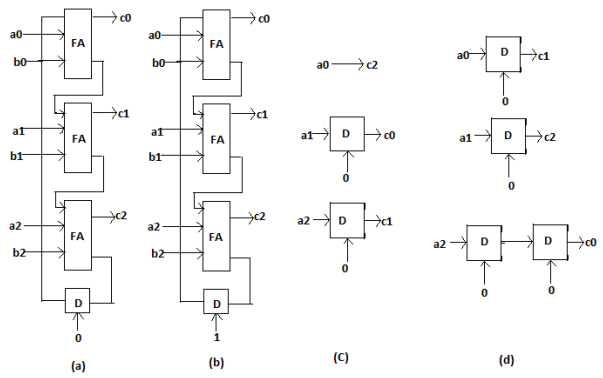

Figure 3: The digit-serial operations when *d* is 3: (a) addition operation; (b) subtraction operation; (c) left shift by 2 times;(d)left shift by 4 times.

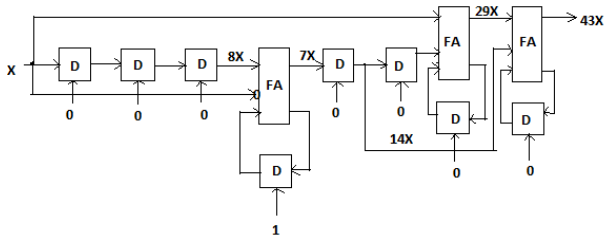

Figure 4: Bit-serial realization of shift-adds implementation of 29*x* and 43*x* given in Figure 2(c).

As an example on digit-serial realization of constant multiplications under the shift-adds architecture, Figure 4 illustrates the bit-serial implementation of 29*x* and 43*x* obtained by the exact GB algorithm [8] given in Figure 2(c). The network includes 2 bit serial additions, 1 bit-serial subtraction, and 5 D flip-flops for all the left shift operations. Observe from Figure 4 that at each clock cycle, one bit of the input data *x* is applied to the

network input and one bit of the constant multiplication output is computed. Note that the digit-serial design of the MCM operation occupies significantly less area when compared to its bit-parallel design and the area of the design is not dependent on the bit-width of the input data. However, the

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 10
June 2016

latency of the MCM computation is increased due to the serial processing. Suppose that *x* is a 16-bit input value. To obtain the actual output of 29*x* and 43*x* in the bit-serial network of Figure 4, 21 and 22 clock cycles are required respectively. Thus, necessary bits must be appended to the input data *x*, *i.e.*, 0s, if *x* is an unsigned input or sign bits, otherwise. Moreover, in the case of the conversion of the outputs obtained in digit-serial to the bit parallel format, storage elements and control logic are required.

Note that while the sharing of addition/subtraction operations reduces the complexity of the digit-serial MCM design (since each addition and subtraction operation requires a digit-serial operation), the sharing of shift operations for a constant multiplication reduces the number of D flip-flops, and consequently, the design area. Observe from Figure 4 that two D flip-flops cascaded serially to generate the left shift of 7*x* by two can also generate the left shift of 7*x* by one without adding any hardware cost.

GB algorithm can be applied for any coefficient pair combinations. Hence GB algorithm is used and number of operations is reduced drastically than other algorithms.

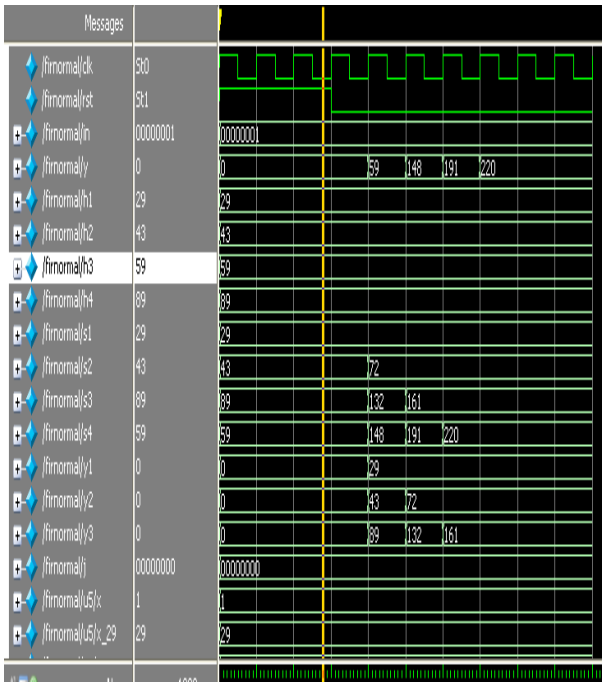

Fig5. Output for FIR filter with digit based recoding algorithm

Four filter coefficient 29,43,59,89 values are taken for digit serial FIR filter design. X(n) is taken as a input sequence and Y(n) is taken as output sequence.

3. SIMULATION RESULTS

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 10
June 2016

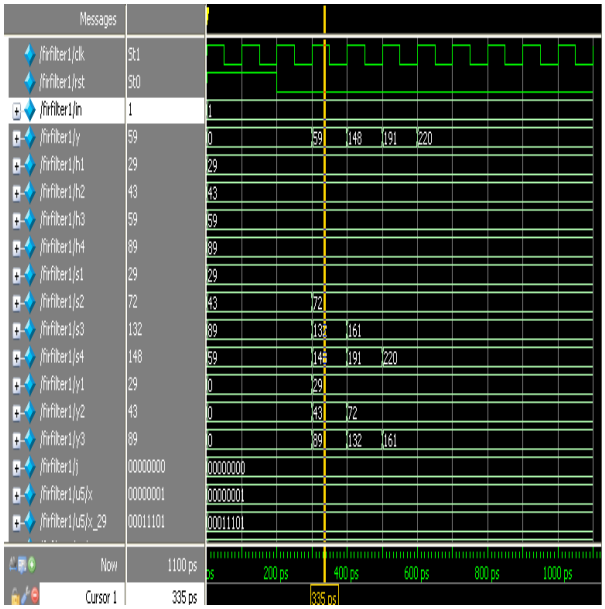

Fig6. Output for FIR filter with CSE algorithm

Fig.5 shows 4 tap FIR filter with without partial product sharing (Digit based recoding) algorithm and Fig.6 displays 4 tap filter with CSE algorithm.

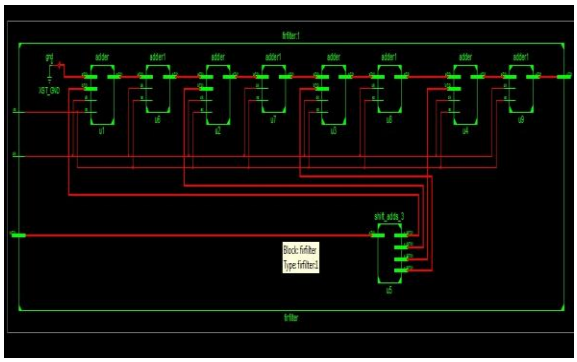

Fig8. RTL schematic for FIR filter with GB algorithm

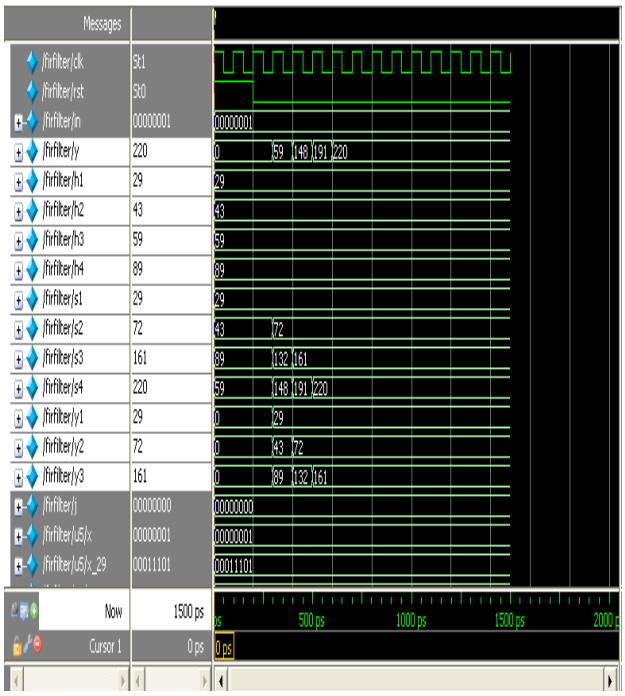

Fig7. Output for FIR filter with GB algorithm

Fig.7 displays 4 tap filter with GB algorithm. This simulation result was displayed by modelsim software. These are the simulation results displayed by modelsim software.

After completion of simulation process in Modelsim tool, synthesis process is takes place to calculate gate count and delay report. Fig.8 shows the RTL schematic view of FIR filter with MCM architecture.

3.1. FIR FILTER DEVICE UTILIZATION REPORT

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 10
June 2016

FIR filter with digit size d=4 was synthesized separately for device utilization in terms of gate count with each different algorithms. Fig 9 and Fig.10 shows the device utilization report of FIR filter with GB algorithm. Table1

```
Device utilization summary:
---------------------------

Selected Device : 3s500efg320-4

 Number of Slices:                       87  out of   4656     1%
 Number of Slice Flip Flops:             36  out of   9312     0%
 Number of 4 input LUTs:                161  out of   9312     1%
 Number of IOs:                          18
 Number of bonded IOBs:                  18  out of    232     7%
 Number of GCLKs:                         1  out of     24     4%

---------------------------
Partition Resource Summary:
---------------------------
```

Fig 9. Device utilization report

explains that the FIR filter with GB algorithm utilized less number of gate count and delay than other two algorithms.

3.2. FIR FILTER DELAY SYNTHESIS REPORT

```
-------------------------------------------------------------------------
Timing constraint: Default OFFSET IN BEFORE for Clock 'clk'
  Total number of paths / destination ports: 8980 / 72
-------------------------------------------------------------------------
Offset:              12.875ns (Levels of Logic = 11)
  Source:            in<0> (PAD)
  Destination:       u3/u9/temp (FF)
  Destination Clock: clk rising

  Data Path: in<0> to u3/u9/temp
                              Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             32   1.218   1.437  in_0_IBUF (in_0_IBUF)
     LUT3:I0->O             6   0.704   0.748  u5/Msub_reg_2_Madd_cy<2>1 (u
     LUT3:I1->O             1   0.704   0.000  u5/Msub_x_59_lut<5> (u5/Msub
     MUXCY:S->O             1   0.464   0.000  u5/Msub_x_59_cy<5> (u5/Msub_
     XORCY:CI->O            6   0.804   0.673  u5/Msub_x_59_xor<6> (h3<6>)
     LUT4:I3->O             1   0.704   0.000  u5/Madd_x_89_lut<6> (u5/Madd
     MUXCY:S->O             0   0.464   0.000  u5/Madd_x_89_cy<6> (u5/Madd_
     XORCY:CI->O            3   0.804   0.610  u5/Madd_x_89_xor<7> (h4<7>)
     LUT4:I1->O             2   0.704   0.526  u3/u7/co1_SW0 (N7)
     LUT3:I1->O             1   0.704   0.595  u3/u6/co1_SW1 (N51)
     LUT3:I0->O             1   0.704   0.000  u3/u8/co1 (u3/c8)
     FDR:D                      0.308          u3/u9/temp
    ----------------------------------------
    Total                      12.875ns (8.286ns logic, 4.589ns route)
                                        (64.4% logic, 35.6% route)

=========================================================================
```

Fig 10. Delay synthesis report

Table -1: Delay and Gate Count Comparison

FIR Filter	Delay	DeviceUtilization
Normal method	14.203	323
CSE Algorithm	15.528	321
GB Algorithm	12.875	299

4. CONCLUSION

Thus the implementation of digit serial FIR filter was implemented with low complexity

MCM architectures using GB algorithm. Device utilization and delay values are compared for hardware implementation. Hence this MCM approach drastically reduces the system complexity, area and delay and FPGA hardware real time implementation has performed with spartan3 version. Future enhancement of this paper is to design MCM architecture with more coefficient pairs for FIR filter implementation.

REFERENCES

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

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

[3] 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.

[4] I.-C. Park and H.-J. Kang, "Digital filter synthesis based on minimal signed digit representation," in Proc. DAC, 2001, pp. 468–473.

[5] L. Aksoy, E. Costa, P. Flores, and J. Monteiro, "Exact and approximate algorithms for the optimization of area and delay in multiple constant multiplications," IEEE Trans. Comput.-Aided Design Integr. Circuits
Syst., vol. 27, no. 6, pp. 1013–1026, Jun. 2008.

[6] A. Dempster and M. Macleod, "Use of minimum-adder multiplier blocks in FIR digital filters," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 42, no. 9, pp. 569–577, Sep. 1995.

AUTHOR1:-

*N.MANASA completed her B.Tech GANAPATHY ENGINEERIMG COLLEGE in 2014 and M.Tech completed in VAAGDEVI COLLEGE OF ENGINEERING

AUTHOR2:-

**V.VIJAYA working as Assoc. prof in Dept of ECE, VAAGDEVI COLLEGE OF ENGINEERING

AUTHOR3:-

***MR.G.BABU working as Assoc. prof in Dept of ECE, VAAGDEVI COLLEGE OF ENGINEERING