ISSN 2277-2685
IJESR/October-December 2024/ Volume-14/Issue-4/94-107
Burra Akhila Goud *et. al.*, / International Journal of Engineering & Science Research

ARCHITECTURE OF HIGH-PERFORMANCE FOURIER TRANSFORM INFRARED FILTERS FOR STATIC AND DYNAMIC USE

Burra Akhila Goud[1] , Dr S Kishore Reddy[2] , P Vijnatha Raju[3] and Vasantha Naga Raju[4]
[1]M.Tech Student, ECE, VLSI System Design, Avanthi Institute of Engg. & Tech., Hyderabad, India.
[2]Associate Professor, HOD, ECE, VLSI System Design, Avanthi Institute of Engg. & Tech., India.
[3]Assistant Professor, ECE, VLSI System Design, Avanthi Institute of Engg. & Tech., Hyderabad, India.
[4]Assistant Professor, ECE, VLSI System Design, Avanthi Institute of Engg. & Tech., Hyderabad, India.

ABSTARCT: *The multiple constant multiplications (MCM) method, which results in significant computing savings, is made possible by transpose form finite-impulse response (FIR) filters. These filters are designed to be pipelined. Transpose form configuration is different from direct form configuration in that it does not permit block processing directly. Our goal in this study is to find out whether it is possible to build high-order FIR filters in a transpose-form configuration that can achieve area-delay efficiency for both fixed and reconfigurable applications. After conducting an extensive computational analysis of the FIR filter's transpose form configuration, we have created a flow diagram for a block FIR filter with optimized register complexity. We provide a generalized formulation of a transpose form FIR filter block. Our universal multiplier-based architecture for the transpose form block filter has been designed for reconfigurable applications. An further low-complexity method for the block implementation of fixed FIR filters using the MCM scheme is also detailed. For medium or large filter lengths, the proposed structure uses much less energy per sample (EPS) and area delay product (ADP) than the current block implementation of direct-form structure. , for short-length filters, the block implementation of direct-form FIR structure uses much less EPS and ADP. In comparison to the best-available FIR filter structure for reconfigurable applications, the proposed structure with a block size of 4 and a filter length of 64 uses 42% fewer active semiconductor photons (ADP) and 40% less effective semiconductor photons (EPS). Compared to the existing direct-form block FIR structure, the proposed structure utilizes 13% less ADP and 12.8% less EPS while maintaining the same filter length and block size.*

I. INTRODUCTION

A few examples of applications that need multiplying by a collection of constants are shown in this section. For the purpose of providing effective solutions to the issue of constant coefficient multiplication in logic circuits, the methodologies that are provided in this thesis may be used in the process of building hardware that is capable of carrying out these duties. When you multiply by a constant matrix or vector, you are doing the identical operation as when you multiply by a constant set. An example of this would be the scalar projection of an onto b, which can be obtained by doing the dot product of a and b (or vice versa). The process of multiplying by a constant matrix consists of nothing more than executing the dot product of a variable vector, the components of which serve as the inputs, and a number of constant vectors, which together make up the matrix. There are several situations in which you would consider the process of multiplication by a constant matrix to be a linear change in coordinates. It is necessary to multiply the RGB (red, green, and blue) colour space by a constant 3x3 matrix convert it into the YUV (yellow, ultraviolet, and chroma) colour scheme. Because the human eye is more sensitive to brightness than it is to colour (chroma), we are able to compress the data in the U and V components with very minimal distortion that is visible to the naked eye. JPEG is the format that employs this for photo compression, whereas MPEG is the one that uses it for video compression. Because the majority of display devices need RGB colour transmission, colour space conversion is an activity that is an essential necessity. Multiplying by a set of constants is a technique that is used in any circumstance that requires a linear translation of coordinates that has been specified. A step that is shared by the Discrete Fourier Transform (DFT) and other Fourier-related transforms, such as the Discrete Cosine Transform, is the process of multiplying by constants. Quite a few discrete linear signal modifications are characterised by this property. Spectrum analysis is a process that includes computing the percentage of the input signal that was provided by a sinusoid at frequency f (for various values of f). It is general knowledge that the discrete Fourier transform (DFT) and transformations related to Fourier may be an effective tool for doing spectrum analysis. that these modifications are functionally equivalent to linear coordinate transformations, they are advantageous for signal compression. The objective of the conversion from the input signal to the coordinate basis is to cram as much information (or energy) as possible into a small number of components. In cases when there is a little amount of distortion, we are able to compress or simply omit the low energy components. In the field of communications, signal compression is necessary for reducing the amount of energy that is used during the transmission of data. When it comes to portable electronic devices, this is of the highest significance since the battery life of these devices is a significant element. Consider the bulk of pictures in the actual world are composed of low frequency information rather than pixels that are created at random. This can help you get a deeper comprehension of signal compression. Visualising a picture in its frequency domain enables us to compress high-frequency data with little distortion, which is a significant benefit. More than that, there is a vulnerability that affects both MPEG and JPEG. This means that the human ear is less sensitive than the ears of other animals when it comes to hearing high-frequency noises. High frequencies are compressed more significantly than low frequencies in the MP3 format, which takes advantage of this phenomenon by compressing high frequencies more heavily. The process of compressing the signal begins with the first stage, which is to move it into the frequency domain. After the compression has been removed, we will need to transfer from the frequency domain to the time domain decompress the audio and listen to it. Once again, a linear signal conversion that is based on constant multiplication is necessary for this method. There are a number of linear signal transformations, like the Fast Fourier Transform (FFT), that take fact that they may reuse intermediate terms in their computations. With an input signal of length n, the number of multiplications decreases from O(n2) to O(n log n), resulting in a decrease in the execution time. In the current state of affairs, it is still essential to do multiplication by constants for this. As an example, the Fast Fourier Transform (FFT) involves multiplying a constant 2x2 matrix by each "butterfly" express complex values in rectangular coordinates. This is done because to addition is somewhat difficult to do in polar coordinates.

II. LITERATURE REVIEW

Within the realm of digital signal processing, words such as "time domain" and "filter input and output signals" are often used. This occurs that periodic sampling is often used to create signals. , there are more avenues to choose samples. In terms of popularity, equal space interval sampling is the second most common approach of them. By way of illustration, consider the process of obtaining data from a series of strain sensors that are spaced one centimetre apart down the length of an aircraft wing. Although there is the potential for a great number of other domains, the most prevalent ones are time and space. In discrete signal processing, the word "time domain" may refer to either the span of time over which samples were collected or, more generally speaking, any domain over which samples were collected. Every linear filter has three primary characteristics: the impulse response, the step response, and the frequency response

The whole filter data set is included into each of these replies, but in a manner that is slightly distinct from the others. Given one of the three, it is possible to compute the other two of them immediately since they are both fixed. It is essential to note that these three examples illustrate how the filter could behave in a variety of circumstances. This is the most fundamental method of applying a digital filter, which involves convolutionally blending the input signal with the impulse response of the filter. By using this method, it is feasible to design any linear filter that is even remotely conceivable. refer to this particular impulse response, filter designers make phrase "filter kernel." Through the use of recursion, digital filters may also be created in another method. When generating the output samples, a convolution filter first takes the input samples, then assigns weights to each of those samples, and then combines all of those weights together. A form of this concept is known as recursive filters, which not only take in input points but also make use of values that have been computed from the output value in the past. A recursive filter is defined by a set of recursion coefficients rather than by a filter kernel. Regardless of the manner in which the filter is implemented, it is important to bear in mind that every linear filter has an impulse response. This is true at least for the time being. It is possible to discover the impulse response of the recursive filter by simply feeding it an impulse and seeing what it produces as a result. Recursive filters produce sinusoids with amplitudes that gradually decrease, and these sinusoids are the impulse responses of the filters. In theory, this has the effect of producing an impulse response that is indefinitely complex. In spite of this, it is possible to disregard the remaining samples since the amplitude will ultimately be lower than the round-off noise of the equipment. It is beca features that recursive filters contain that they have been renamed as "IIR" filters. This is in contrast to convolutional filters are referred to as FIR filters. The behaviour of a system in response to an impulse is referred to as its impulse reaction. , step replies are generated whenever steps are employed as inputs. In light of the step is the integral multiple of the impulse response being considered. locate the step answer, below are two methods: (1) see the outcome, a step waveform must be sent through the filter.

Consider the response that appears out of nowhere in the second position. Utilising the density-functional theory (DFT) of the impulse response is one method that may be accomplished get the frequency response.

III. PROPOSED SYSTEM

Designers of embedded systems have been grappling with the challenging problem of balancing run time and power consumption for a considerable amount of time. perform the computationally expensive processing of visual data, such as photographs and motion pictures, embedded systems often make use of DSP techniques. There is a possibility that the energy and time requirements of embedded devices might be satisfied by a specialised hardware implementation of these algorithms. The act of multiplying a variable by a set of constants is the foundation for a wide variety of techniques, including digital filtering, image processing, linear transformations, and many more. It is possible that improving these multiplications might be beneficial to a great number of design objectives, such as the use of electric power or space. The term that we use to refer to this issue is really an MCM, which stands for multiple constant multiplication question.

A novel strategy is presented here solve the MCM issue. In comparison to all of the other methods that have been disclosed up to this point, our approach is superior in terms of the total amount of computations that are required to solve the problem. A further advantage of the new method is that it is more adaptable. We begin with a more in-depth introduction to the subject matter ensure that our contribution is easily discernible and is placed within the framework of the research that has been done before.

3.1.1 Single Constant Multiplication (SCM)

The equation y = tx is obtained by multiplying x by a known integer or fixed-point constant t. Decomposing this expression further shows additions, subtractions, and binary shifts. In [Cappello and Steiglitz 1984], it is shown that the single constant multiplication (SCM) issue, which is defined as the process of finding the decomposition that requires the least amount of operations, is an NP-complete mathematical problem. that a fixed-point multiplication is similar to multiplying by an integer and then shifting the result to the right, we are able to assume that the constants are integers without compromising our ability to generalise. The SCM issue is comparable to the addition chain problem [Knuth 1969], which requires multiplying by a constant using adds alone. However, this problem is distinct from the SCM problem. In the event that shifts are provided authorisation, the issue as well as the techniques that are used to remedy it undergo a transformation
When multiplication is described as an add-and-shift operation, the first step is to shift the inputs, which are then translated from the binary representation of the constant t into shifts that are generated by the process. When t is equal to 71, for instance, the equation 71x = 10001112x is equivalent to the expression $x \geq 6 + x \geq 2 + x \geq 1 + x$. Three more components are required for it. Subtracting from the nearest constant that is composed of 1s (i.e., of the form 2n − 1) is another method for breaking down the multiplication into shifts and subtracts. This method

involves converting the 0s into shifts and then subtracting from the nearby constant

71x = 1001112x = (x ≥ 7 − x) − x ≥ 5 − x ≥ 4 − x ≥ 3, for every x ≥ 7 this is the calculation. The answers for combining the best features of these two algorithms are 2b + O(1) adds/subtracts, where b is the bitwidth of t. These solutions represent the worst-case scenario and the average-case scenario, respectively.

By re-coding the number into the canonical signed digit (CSD) form, a more effective digit-based technique divides the information into addition and subtraction, and it also enables for negative numbers to be equal to one [Avizienis 1961]. By using CSD, we are able to cut down on the amount of add/subtract operations that are necessary for the example that was shown before to only two:

The correct solution is 1001001CSDx, which is equivalent to 10001112x + x to the power of three.

The optimum decomposition employs a different network architecture than the CSD de-composition, as you can see in the previous sentence. By virtue of digital approaches such as CSD only take into consideration a specific kind of network design, it is reasonable to assume that they provide outcomes that are less than optimal. , the exhaustive search techniques described in [Dempster and Macleod 1994; Gustafsson et al. 2002] take into account every conceivable network topology locate the best decompositions

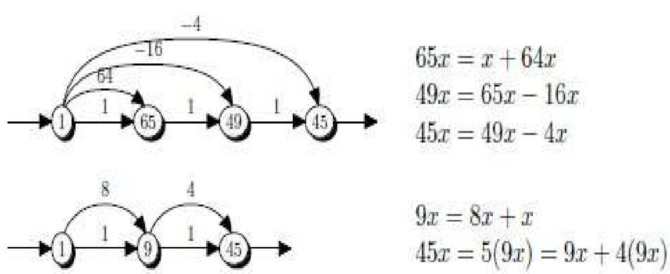

Fig. 3.1. The result should be obtained, and then it should be multiplied by 45 using the two sets of computations (optimal, bottom) and the three sets (CSD, top).

While the edges represent shifts that are labelled with the proper scaling (a two-power), the vertices represent operations that are performed by adding and subtracting and are marked with their outputs. When the scale displays a negative value, it is essential to do a subtraction.

3.1.2 A decrease in the number of adders by graph dependency techniques

If you want to cut down on the number of adders, you have a variety of options. Before incorporating the following algorithms into CSE, the project's objective is to first test and research them, then assess how well they function, and then include them.

ISSN 2277-2685
IJESR/October-December 2024/ Volume-14/Issue-4/94-107
Burra Akhila Goud *et. al.,* / International Journal of Engineering & Science Research

3.1.3 Bull Horrocks (BH) Algorithm [1]

Through the use of this strategy, which is a graph-based design technique, simple procedures such as convolution may be carried out with a reduced number of arithmetic operations. In an attempt to visually display the outputs of the constant multipliers, this approach takes common parts wherever it is practical to do so. As an example, the expression w2[n] = 17x[n] is produced in Figure 1 by associating the outputs of the multipliers with the letter w and the output of the second multiplier with the letter w2. The effective weighting of the signal x[n] at each vertex is represented by the integer value that is next to each vertex. We can see from this figure that there are lower numbers that can be multiplied by 17, and even smaller values that can be multiplied by further coefficients. This is something that we can see that is possible.

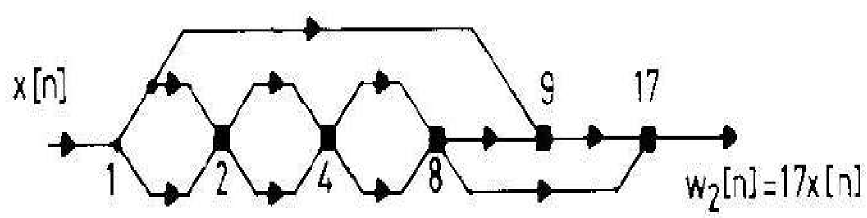

Fig. 3.2: Explanation of the graph for w2[n] = 17x[n,] starting from [1]

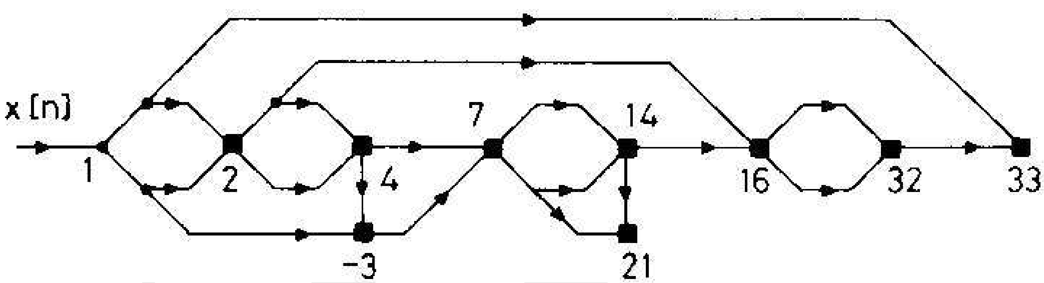

Fig. 3.3: Representation of the graphs 1, 7, 16, 21, and 33, taken from [1]

Here we have an alternative example with 7, 16, 21, and 33 as coefficients. Figure 4.3 shows how additional coefficients might make advantage of the intermediate outcomes.

3.1.4 The n-dimensional reduced adder graph (RAG-n) algorithm [3]

The RAG-n method is composed of two distinct pieces. A heuristic approach is used in the second portion of the computation, whereas the first part is an accurate procedure. Should the collection of coefficients be entirely synthesised, it is possible to determine the adder cost that is the least expensive in the first section. For each coefficient, a look-up table is used in Section 2 of the investigation. To summarise, the algorithm is comprised of the following phases in their entirety:

Reduce the coefficients of each set to their odd fundamentals complete the process.
A single coefficient may be used to assess all expenses, and the cost lookup table can be utilised to do so.
Eliminate the essentials that are provided at no cost.

Construct a graph that illustrates the basics that have been chosen.
The task at hand is to determine the power-of-two multiples of the basic sums that are included inside the graph set. In the event that you want to bypass the fundamentals, just return to step

3.1.5 Multiple Constant Multiplication (MCM)

Figure 3 illustrates the difficulty of multiplying x by a large number of constants t1,..., tn in a so-called multiplier block. This problem is an enlargement of the SCM problem. It is feasible to deconstruct a multiple constant multiplier block into less operations than the total of the operation counts of the single constant decompositions. This is achievable that it is allowed to share the intermediate outcomes of the constant decompositions. Multiple Constant Multiplication (MCM) presents a challenge in that it requires the user to locate the breakdown that requires the fewest operations.There is a correlation between the number of constants and the potential savings that may be achieved via the exchange of intermediate findings.

Both the multiplierless implementation of digital finite impulse response (FIR) filters (Bull and Horrocks 1991) and matrix-vector products with a fixed matrix (P¨uschel et al. 2004; Chen et al. 2002; Liang and Tran 2001) are examples of applications that specifically involve the MCM problem. Within these applications, linear signal transforms such as the discrete Fourier or discrete cosine transform are included. Every single input sample is multiplied by all of the n taps that are used in an n-tap FIR filter. Discrete Fourier and trigonometric transform approaches, , require the simultaneous multiplication of two constants and include rotations of a size of two by two.

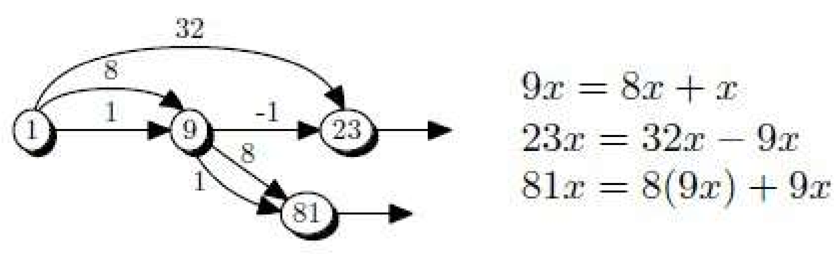

Fig. 3.4. Multiplier block with constants 23 and 81.

Fig. 5 represents a block that serves as an illustration of multipliers. There are only three addition and subtraction operations required to perform the parallel multiplication of 23 and 81. This is despite the individual optimum decompositions of 23 and 81 need two operations each.
Tummelshammer et al. (2004) wrote a second work in which they discussed multiplexed multiple constant multiplication from a different perspective. In this particular instance, the control logic of the multiplier block switches multiplexers accomplish multiplication by having different constants. This method allows for the combination of consecutive multipliers to be accomplished.

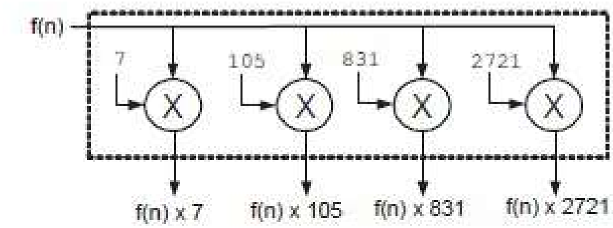

Figure 3.5(a): A case study of MCM using multipliers

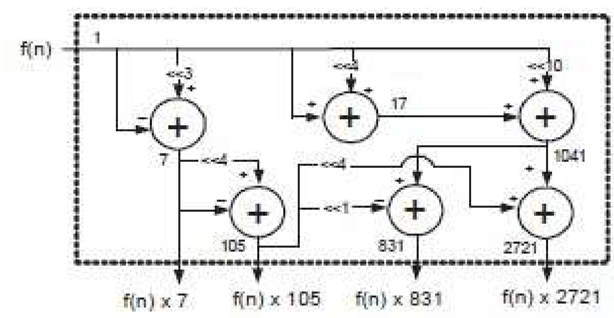

Figure 3.5(b): A case of multiplier-based MCM

The constants are first converted into binary using a basic technique known as digit-based recoding [2], which is followed by the construction of constant multiplications via the use of shift-adds. First, we add up all of the shifted variables, and then we move the variable by one bit for every "1" that appears in the binary form of the constant. This gives us the result. For simplicity's sake, let's consider about the constant multiplications 29x and 43x. An inventory of their binary breakdowns is shown in the following passage.

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

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

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

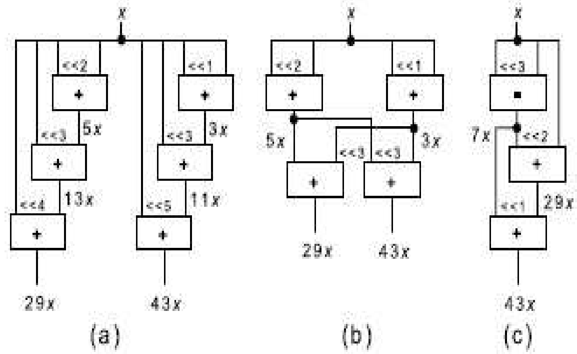

Fig. 3.5 The 29x and 43x implementations alter the additions in their respective ways. (a) Goods that are partially or completely shared [2-3]. Method of CSE that is accurate (b) [5]. b) The accuracy of the GB algorithm [in the sixth place].There are two primary groups of techniques to the MCM problem: the graph-based (GB) approach [6]-[8] and the common subexpression elimination (CSE) algorithms [3]-[5]. Both of these examples are examples of methodologies.

The best possible answer to the issue of maximal sharing is presented in this article. In recent years, there has been an overwhelming amount of study conducted on this subject. There have been two major techniques that have had a significant impact on the optimisation of MCMs. It is possible to combine partial words by employing subtractors rather than adders significantly increase the possibility of sharing subexpressions that are related to one another.

To represent the coefficients, the CSD format is used, which brings us to the second explanation. Due to the limited number of non-zero digits, the maximum subexpression sharing search in this encoding requires a low level of complexity. This is something that must be taken into consideration.An investigation that was conducted not too long ago by Park [9] suggests that the coefficients need to be represented in a form known as Minimal Signed Digit (MSD). By removing the constraint that there must be two consecutive non-zero digits, it is feasible to deduce the MSD representation from the CSD representation. This is accomplished by determining the MSD representation. When it comes to a certain numerical value, the MSD format makes it possible to have many representations of that number. , the number of digits in the CSD representation that are not zero remains the same throughout all of our examples. By using the redundancy of the MSD representation, the strategy that is suggested in [9] selects the MSD instance that results in the maximum amount of sharing, which enables the development of efficient FIR filters.Heuristics, which do not describe the degree to which the actual answer departs from the ideal, have been used in every prior effort to solve this issue, as far as we are aware. They have been used in every single one of those attempts. The technique that we use is precise and may be applied to a wide variety of real-world issues. describe this issue, we construct a Boolean network that contains all of the potential partial terms for the coefficient set represented by the MCM instance. The MCM process uses the values that flow into this network as inputs, and the values that flow into this network are modified copies of those values. Every single adder and subtractor that is used in the process of producing a particular partial term is represented by an AND gate. The conditional operator (OR) is used to combine any partial words that have the same numerical value. An AND operation is performed on all of the coefficients, which is the sole output of the MCM. This problem is transformed into a 0-1 Integer Linear Programming (ILP) problem by applying the constraint that the output be confirmed or that the collection of partial terms acquired contain all coefficients. This allows us to turn this issue into an ILP problem. At the same time, we are looking to cut down on the overall amount of AND gates or adders/subtracters that evaluate to one.

Every single one of the coefficients that are expressed in binary, CSD, and MSD formats have been processed using this procedure. It is important to point out that our model readily incorporates the redundancy of the MSD representation. This is because the matching MSD representations are simply fresh inputs to the OR gate, which is responsible for producing a specific coefficient.

When constants are expressed under binary, the most frequent partial products are 3x = (11)binx and 5x = (101)binx, respectively, according to the precise CSE technique of [9]. This is a return to our earlier example, which can be found in Figure 2. In Figure 2(b), this is shown. Using this method results in a solution that requires four different processes. , achieve the smallest amount of operations that are feasible, the precise GB approach [6] use the same partial product 7x in both multiplications, as seen in Figure 2(c). When utilising the precise CSE approach, it is important to keep in mind that the partial product 7x = (111)binx cannot be obtained from the 43x binary representation [5].

3.2 DIGIT-SERIAL ARITHMETIC

The method of digit-serial arithmetic involves the division of data words into digits that have a size of d bits, and the processing of each digit takes place within a single clock cycle. Bit-serial processing and bit-parallel processing are two specific examples of digit-serial computing, respectively. These scenarios occur when the value of d is equal to one and the length of the input data word is also equal to one. In situations when the hardware requirements for bit-parallel designs are too high and bit-serial implementations are unable to fulfil latency requirements, the significance of digit-serial computing becomes even more significant. Play about with the digit size option (d) to find the optimal balance between latency and area. This is one technique to discover the sweet spot. The digit-serial operations that are considered to be the most basic were stated in [8]. Figure 3 illustrates the digit-serial operations of addition, subtraction, and left shift when d = 3. These operations are represented in the table below. The number of full adders (FAs) that are required for a digit-serial addition operation is usually d, but the number of D flip-flops that are required for such an operation is always 1. This is shown in Figure 3(a). Figure 3(b) illustrates the subtraction operation, which necessitates the complement of two. Because of this, the operation of digit-serial addition necessitates the inclusion of d inverter gates in addition to the initialisation of the D flip-flop with 1.

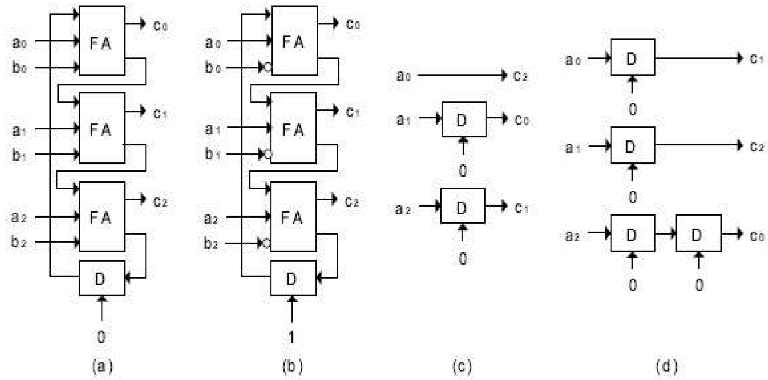

Figure 3.6: The serial-digit operations that are performed when d equals three are as follows: a left shift (d) performed twice, a subtraction performed three times (b), and an addition operation (a).

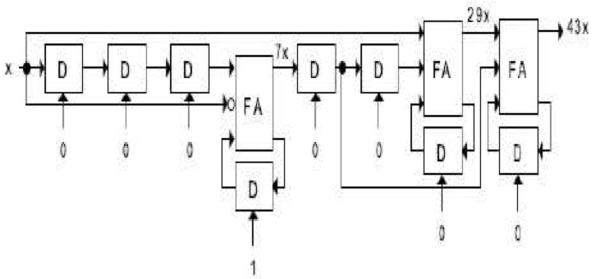

Figure3.7 : Bit-serial realization of shift-adds implementation of 29*x* and 43*x* given in Figure 2(c).

The bit-serial implementation of 29x and 43x is shown in Figure 4 as an example of the digit-serial realisation of constant multiplications under the shift-adds architecture. This is accomplished by using the same GB method [2] as is shown in Figure 2(c). Five D flip-flops, one bit-serial subtraction, and two bit serial additions are used by the network carry out all left shift operations. As shown in Figure 4, the network input is given one bit of input data x for each clock cycle, and the output of the constant multiplication is calculated using one bit of input data. The digit-serial design of the MCM operation is much smaller than the bit-parallel design, and the size of the design is not connected to the bit-width of the input data. This is something that should be taken into consideration.

It is important to note that although sharing shift operations for a constant multiplication decreases the number of D flip-flops and, as a result, the design area, sharing addition/subtraction operations simplifies the design of a digit-serial MCM (that each operation includes a digit-serial operation). Furthermore, as seen in Figure 4, two D flip-flops that are cascaded in a serial fashion to produce a 7x by two left shift are also capable of producing a 7x by one left shift without the need for any additional hardware.

ISSN 2277-2685
IJESR/October-December 2024/ Volume-14/Issue-4/94-107
Burra Akhila Goud *et. al.*, / International Journal of Engineering & Science Research

IV. RESULTS

4.1 SIMULATION RESULT

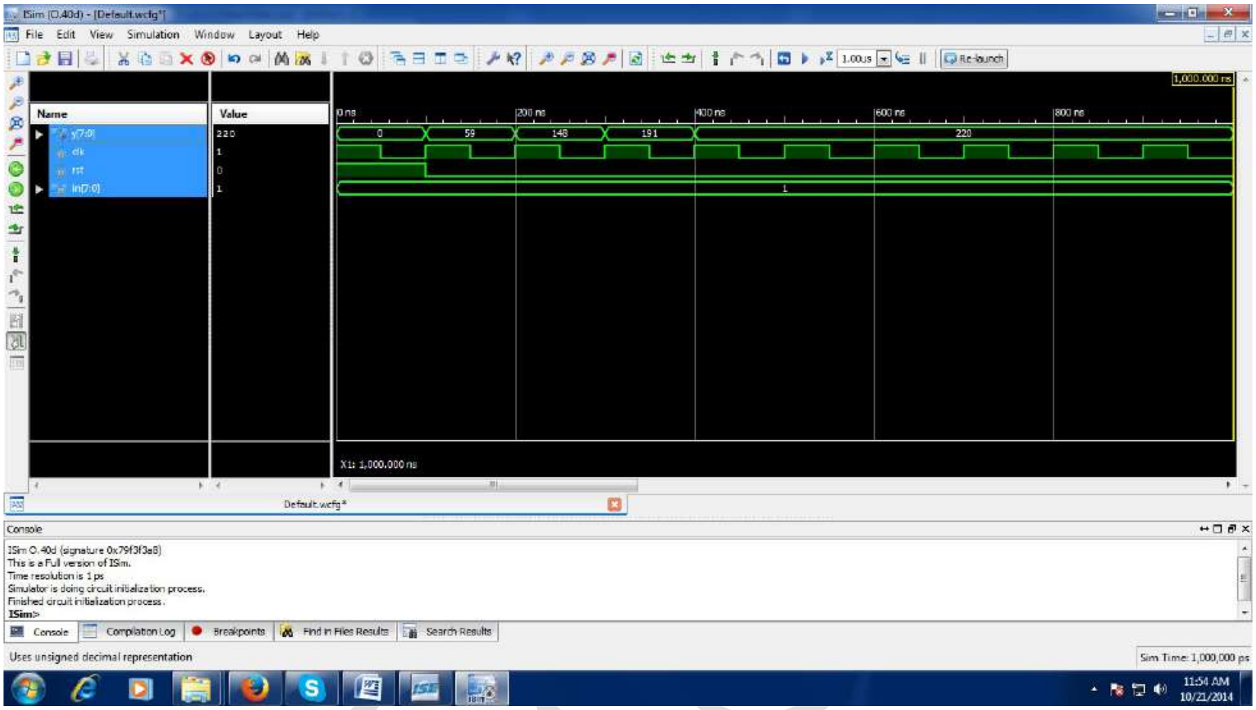

Fig 4.1 Encoder

In Fig 4.1 shows the output of the encoder. A fifteen-bit codeword is produced as a result of this process, which involves adding the priority bits to seven bits of information internally.

4.2 BLOCK DIAGRAM

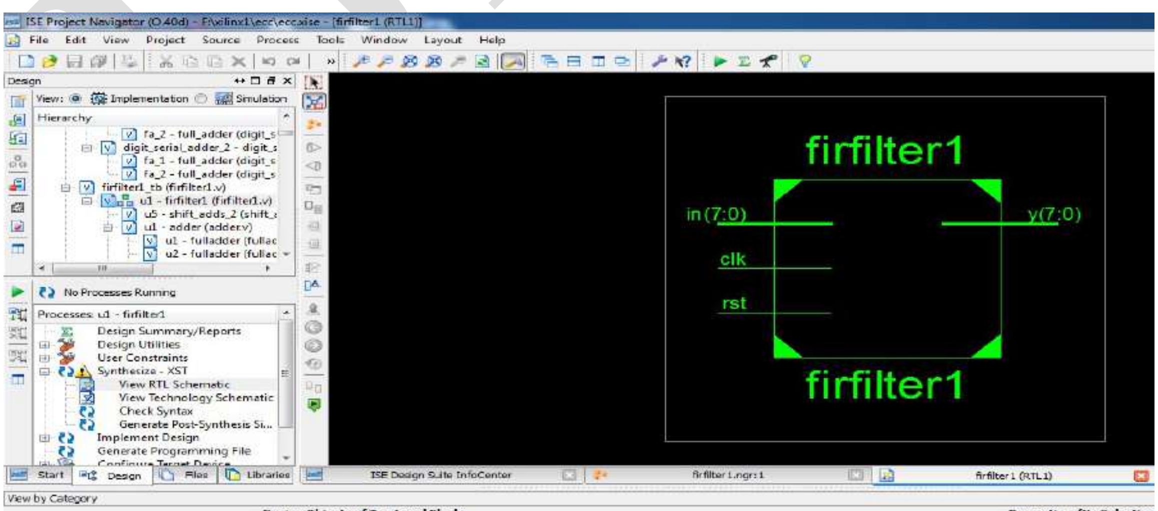

Fig 4.2 Block diagram

ISSN 2277-2685
IJESR/October-December 2024/ Volume-14/Issue-4/94-107
Burra Akhila Goud *et. al.*, / International Journal of Engineering & Science Research

4.2.1 RTL

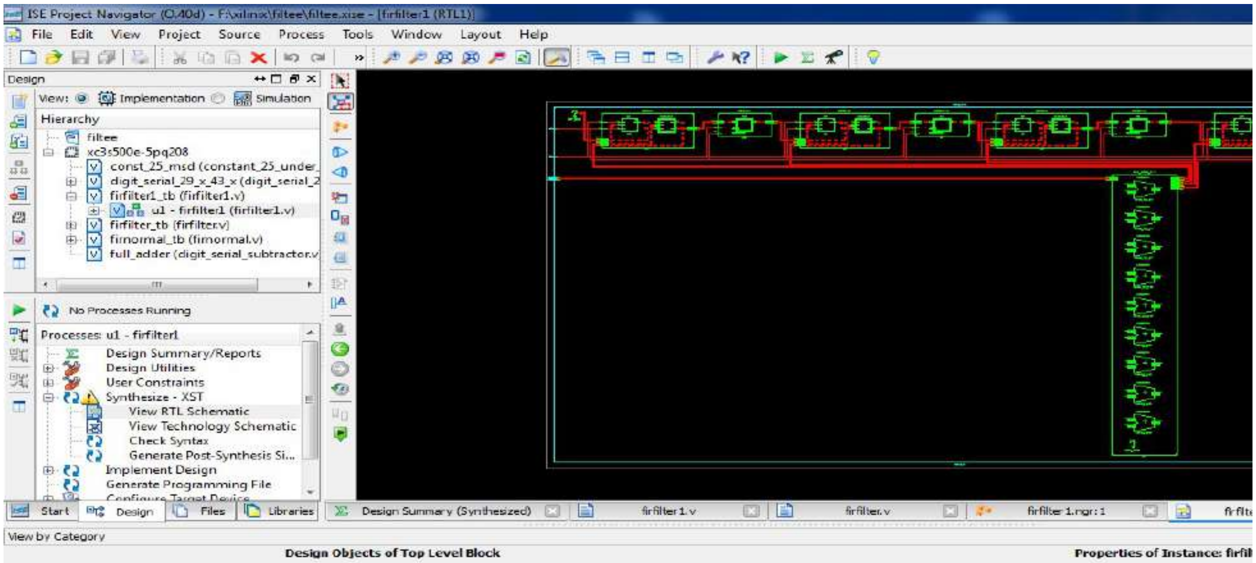

Fig 4.3 RTL

4.2.2 TECHNOLOGY

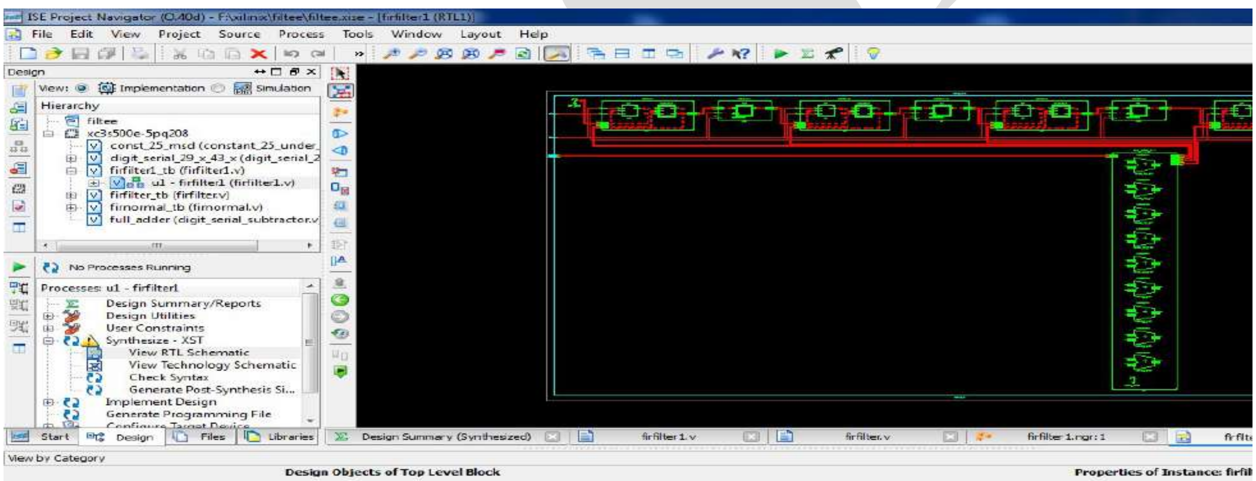

Fig 4.4 technology

V. CONCLUSION

In this study, a unique strategy was provided to solve the optimisation issue of multiple constant multiplication. This problem requires multiplying a variable by a collection of constants, and this research offered a possible solution. The common subexpression reduction technique is the foundation of this strategy, which combines an exhaustive search for multiple pattern identification with a steepest descent approach for pattern selection. The findings indicate that acceptable runtimes may be achieved while simultaneously lowering the amount of mathematical operations or hardware operations required to carry out certain jobs. The suggested approach might be considered an extension of the 2-bit pattern optimisation techniques described in [4]. This is because the proposed method does not impose any limits on the existing patterns. When it comes to FIR filter

ISSN 2277-2685
IJESR/October-December 2024/ Volume-14/Issue-4/94-107
Burra Akhila Goud *et. al.,* / International Journal of Engineering & Science Research

optimisation, our method surpasses other research that uses data that is freely accessible to the public, or at the very least, it comes close. The most significant benefit of the technique is that it is a universal idea that allows it to be used to a variety of different issues. This is in addition to the applications that are suggested in this research, which include the design of FIR filters and the optimisation of linear transforms.

FUTURE SCOPE

Both the amount of space and the amount of power that is used need to be reduced even more.

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.

[8] L. Aksoy, E. Gunes, and P. Flores, "Search algorithms for the multiple constant multiplications problem: Exact and approximate," *J. Microprocess. Microsyst.*, vol. 34, no. 5, pp. 151–162, Aug. 2010.

[9] P. Cappello and K. Steiglitz. Some Complexity Issues in Digital Signal Processing. *IEEE Trans. on Acoustics, Speech, and Signal Processing*,32(5):1037.1041, October 1984.

[10] R. Hartley and P. Corbett. Digit-Serial Processing Techniques. *IEEE TCAS II*, 37(6):707-719, 1990.

[11] L. Aksoy, E. Gunes, and P. Flores. Search Algorithms for the Multiple Constant Multiplications Problem: Exact and Approximate. *Elsevier Journal on Microprocessors and Microsystems*, 34:151-162, 2010.

[12] L. Aksoy, C. Lazzari, E. Costa, P. Flores, and J. Monteiro, "Efficient shift-adds design of digit-serial multiple constant multiplications," in *Proc. Great Lakes Symp. VLSI*, 2011, pp. 61–66.

[13] P. Flores, J. Monteiro, and E. Costa, "An exact algorithm for the maximal sharing of partial terms in multiple constant multiplications," in *Proc. Int. Conf. Comput.-Aided Design*, Nov. 2005, pp. 13–16.