

AREA EFFICIENT REALIZATION OF COEFFICIENT ARCHITECTURE FOR BIT-SERIAL FIR, IIR FILTERS AND COMBINATIONAL/SEQUENTIAL LOGIC STRUCTURE WITH ZERO LATENCY CLOCK OUTPUT**FIELD OF THE INVENTION**

The invention relates to area efficient realization of coefficient block [A] or architecture [A] with hardware sharing techniques and optimizations applied to this block. The block [A] is connected to coefficient lines CLin_0, CLin_1.....CLin_n and BLin_0, BLin_1,....BLin_n coming from block [E] and/or [F], to be connected to perform filtering operation or a mathematical computing operation with optimization in hardware and provides a zero latency output. The invention also gives the area minimal realization of digital filters based on coefficient block[A], when operated in bit serial fashion. The optimization techniques and structure of the present invention are good for linear digital filters typically a finite impulse response(FIR) filter, infinite impulse response filter(IIR) and for other filters and applications based on combinational logic consisting of delay element(T), multiplier(M), adder(SA) and subtractor (SS).

Brief description of the accompanying drawings

In the accompanying drawings:

Figure 1 shows the field of invention, applications of the device

Figure 2 shows the symbol of components used in device.

Figure 3 shows the description of the components used in device.

Figure 4 shows the bit serial FIR filter implementations.

Figure 5 shows an example of FIR filter.

Figure 6 shows one of the known minimization technique due to symmetry of coefficient.

This Sheet Has Been Replaced



- 2 -

Figure 7 shows the structure of existing/known implementation technique for example FIR filter.

Figure 8 shows the generalized structure of existing/known implementation technique of coefficient block.

Figure 9 shows a realization of the coefficient block for coefficient value close to power of two.

Figure 10 shows the optimization (a) for realizing the coefficient block for example FIR filter, of the present invention.

Figure 11 shows the optimization (b) for realizing the coefficient block for example FIR filter, of the present invention.

Figure 12 shows the optimization (c) for realizing the coefficient block for example FIR filter, of the present invention.

Figure 13 shows the generalized optimized structure, of the present invention.

Details of Elements/symbol used in the description

The basic components symbol used in design are shown in "Figure 2" of the drawings. In addition, explanation and usages of the device are done in the text below & depicted in "Figure 3" and "Figure 4" of the drawings.

Unit delay (T)

It is one bit delay element. It also performs function of a multiplier by a factor of 2. [e.g. For the serial input frame (0101011 in binary or 43 in integer representation), the output of this block is (01010110 in binary or 86 in integer representation). This element is usually a Flip-flop (D Flip-flop, J-K Flip-flop etc.).

Full adder (FA)

It performs binary addition. The inputs to this element are A, B, Cin (Carryin) while the outputs are Z and Cout (Carryout). The truth table for full adder functionality is shown in "Figure 3" of the drawings.

This Sheet Has Been Replaced



Full subtractor (FS)

It performs binary subtraction. The inputs to this element are A, B, Cin (Carryin) while the outputs are Z and Cout (Carryout). The truth table for full subtractor functionality is shown in "Figure 3" of the drawings.

Serial adder (SA) and Serial Subtractor (SS)

It performs addition/subtraction of two serial frame, $x_1(nT)$, $x_2(nT)$ to generate output $y(nT)$ represented as $x_1(nT)+x_2(nT)$ or $x_1(nT)-x_2(nT)$. The serial adder (or subtractor) is implemented using a full adder (or subtractor) with a Flip-Flop as shown in "Figure 3" of the drawings. The output Cout of [FA/FS] is delayed using the [T] element and is applied to Cin line of [FA/FS]. This enables the [FA/FS] and [T] together to function as serial adder (SA/SS), where A, B are the inputs to this element and Z is the output. (e.g of serial addition is as follows, if $x_1(nT) = 0110$ (6 in integer) and $x_2(nT) = 0111$ (7 in integer). Then $y(nT) = 01101$ (13 in integer representation).

Serial Multiplier (M)

It multiplies two serial input frame $X(nT)$ and m . The output is a function represented as $Y(nT) = X(nT) * m$. A serial coefficient multiplier(M) can be implemented by shift register using [T] elements and adder element [SA] (One shift means multiply by factor of 2). As shown in "Figure 3" of the drawings, the multiplier is formed by adding the outputs corresponding to ones in the binary representation of the coefficient.

Delay (Z^{-1})

- Delay by one frame of data is done by shift register (series of Flip-flops (T) connected to store and shift the input frame). The number of Unit delay (T) in one delay element is equal to the frame size of the input.

PRIOR ART OR EXISTING IMPLEMENTATION OF FILTER

The following description discusses the elements used for implementation of design and the existing implementations for digital filters. The proposed minimization is extendable to other applications such as Digital Signal Processing field and Digital designs.

From here onwards, all the illustration would be done with the preferred embodiment namely FIR filter which is extendable to other filters as described earlier. "Figure 4" shows the existing structure of bit serial FIR filter with coefficient lines CLin_0, CLin_1,.....CLin_n and the coefficient block [A] having the coefficients c(0), c(1), c(2),....c(n). The coefficient block is connected to delay element [Z^{-1}] and serial adders [SA] to form filter structure.

Stating the FIR filter equation in time and frequency domain

$$Y(n) = c(0) X(n) + c(1) X(n-1) + c(2) X(n-2) + \dots c(n) X(0)$$

$$Y(z) = X(z) [c(0) + c(1) Z^{-1} + c(2) Z^{-2} + c(3) Z^{-3} + c(4) Z^{-4} + c(5) Z^{-5} + c(6) Z^{-6} + \dots + c(n) Z^{-n}]$$

where X, Y are the input and output respectively and c(0), c(1).....c(n) represent the coefficients value which defines the characteristics of the filter and each delay [Z^{-1}] block represents one sample delay. The filter equation can be implemented in two ways as shown in "Figure 4" of the drawings

In implementation 1, coefficient lines CLin_0, CLin_1,.....CLin_n are common and connected to input X[n]. The output lines CLout_0, CLout_1,.....CLout_n are connected to block [E], consisting of delay element [Z^{-1}] and serial adders [SA] elements. The structure makes easy realization of share-able multiplier in the coefficient block [A]. An example of share-able multiplier with coefficient values 3,11 is illustrated in "Figure 4". The realization of these coefficient separately

would require $4[T]$, $3[SA]$ elements. By virtue of $CLin_0$, $CLin_1, \dots$ being common, the hardware is realized using $3[T]$, $2[SA]$ elements. Another feature of the structure is that the structure inherently requires more storage area, represented by $[Z^{-1}]$, as compared to implementation 2, since the storage is done after the multiplication. For input frame of n bit and coefficient of size m bit, the storage area of each delay element $[Z^{-1}]$ is $(m+n)$. The total storage space of the delay elements is $(m+n) * (\text{number of coefficients} - 1)$.

In implementation 2, the coefficient line $CLin_0$, $CLin_1, \dots$ are not common. By virtue of connectivity of different input lines to the coefficient elements $[c(0), c(1), \dots]$, the realization of coefficient block $[A]$ using share-able elements is not present. Another feature of this structure is that it inherently requires lesser storage space, represented as $[Z^{-1}]$, as unlike in previous implementation, here the storage is done before multiplication. For input frame of m bit and coefficient of size n bit, the storage area of each delay element $[Z^{-1}]$ is (m) . The total storage space is $(m) * (\text{number of coefficients} - 1)$.

The invention reduces the hardware of the coefficient block $[A]$ by having share-able elements in coefficients, even if the coefficient lines $CLin_0$, $CLin_1, \dots$ are not commonly connected. For existing configuration as shown in "Figure 7" and "Figure 8", the share-ability of hardware in block $[A]$ is a limitation.

Also, as described in previous section, implementation 2 is area efficient with respect to implementation 1 due to reduced delay elements size. Over and above this by having share-able multiplier or reduced coefficient block $[A]$, which are the key features of the invention, implementation 2 becomes still more area-efficient. This reduction is extendable to other filter based on coefficient block $[A]$ as stated in the first section. The present invention operates on integer valued coefficient.

Further, to quote Norsworthy and Crochiere (Delta-Sigma Data Converters IEEE press pp-435, copyright 1997)

"Bit-serial architecture reduce the interprocessor communication down to 1 bit. Generally the number of processors is very large, but because each processor is so small, the overall economy is very high. Bit serial architectures are usually most effective for filters having a few state variables, such as IIR filters and the wave-digital filters. For this reason, bit- serial techniques are less frequently applied to FIR structures, especially when the filter length is relatively long....."

However, the present invention applies optimization techniques for reducing the area in large sized coefficients by applying a number of optimizations in FIR/IIR filter structures

To elaborate the applicant's optimization techniques, consider an FIR filter with symmetrical coefficient as 5, 14, 25, 30, 25, 14, and 5. Though, the size of the coefficients in this example is small, it is enough to elaborate the minimization proposals. Stating the FIR filter equation in time and frequency domain

$$Y(n) = c(0) X(n) + c(1) X(n-1) + c(2) X(n-2) + \dots c(n) X(0)$$

$$Y(z) = X(z) [c(0) + c(1) Z^{-1} + c(2) Z^{-2} + c(3) Z^{-3} + c(4) Z^{-4} + c(5) Z^{-5} + c(6) Z^{-6} + \dots + c(n) Z^{-n}]$$

where X, Y are the input and output respectively and c(0), c(1).....c(n) represent the coefficients value.

Using the coefficient values in above equation

$$Y(n) = 5 X(n) + 14 X(n-1) + 25 X(n-2) + 30 X(n-3) + 25 X(n-4) + 14 X(n-5) + 5 X(n-6)$$

$$Y(z) = X(z) [5 + 14 Z^{-1} + 25 Z^{-2} + 30 Z^{-3} + 25 Z^{-4} + 14 Z^{-5} + 5 Z^{-6}] \quad (\text{EQ 1})$$

The Existing Method and Minimization

"Figure 5" of the drawings shows FIR filter structure of implementation 2. This figure illustrates the realization of FIR filter represented by "Equation 1".

In one of the known optimization technique, is taken advantage of the symmetry in the coefficients. The streams which have to be multiplied with the same coefficients can be added first and then multiplied. For a large filter structure, this leads to a reduction by 45% in the coefficient block. (see "Figure 6" of the accompanying drawings)

This is done by restructuring the equation as under:

$$Y(z) = X(z)[5*(1+Z^{-6}) + 14*(Z^{-1}+Z^{-5}) + 25*(Z^{-2}+Z^{-4}) + 30 * Z^{-3}] \quad (\text{EQ 2})$$

For the rest of the optimization proposals it will be talking about only the multiplier adder series which is shown in the dotted box referred to as coefficient block [A]. "Figure 7" of the drawings shows the traditional way of implementation of the example structure for block [A], wherein S1 to S4 represent the lines connected to delay block $[Z^{-1}]$ through line CLin_0 to CLin_6 depicted in "Figure 6" of the drawings. The Lines S1 to S4 are separately connected to [T] element for performing a multiplication by "a factor of 2" and (SA) is being used to perform serial addition of data. This represents the multiplier less realization of filter coefficient block [A] where the property of flip-flop (T) as multiplier of two is used.

Mathematically, the restructured equation according to the structure is stated as

$$Y(nT) = (4+1)S1 + (8+4+2)S2 + (16+8+1)S3 + (16+8+4+2)S4 \quad (\text{EQ 3})$$

In this implementation, S1, S2, S3, S4 lines are not commonly connected. Hence this restricts to achieve a share-able hardware in coefficient block [A]. Thus all the

function/operations of this block represent unique hardware. The elements required by the terms are listed as

First term = 2[T], 1[SA] elements

Second term = 3[T], 2[SA] elements

Third term = 4[T], 2[SA] elements

Fourth term = 4[T], 3[SA] elements

Final addition of all the four term would require 3[SA] elements.

The generalized structure of "The Existing Method and Minimization" is depicted in "Figure 8". In the structure, each column represents a coefficient value. The [T] elements, shown as T1_1 to T1_m in column 1, defines connectivity with line S1. In similar fashion, [T] elements, shown as Tn_1 to Tn_m in column n, defines the connectivity with line Sn.

The presence of one of the elements in columns 1 to n (i.e T1_1 to T1_m, T2_1 to T2_m.....Tn_1 to Tn_m) is determined by coefficient value. Thus depending on coefficient value on lines S1 to Sn, the number of [T] element in a column is determined. Also the number of serial adders/subtractor [SA/ SS] in columns is represented as (SA1_1 to SA1_m, SA2_1 to SA2_m.....SAN_1 to SAN_m). The presence of one of these elements is again defined by the coefficient value.

In the structure, the [T] elements are arranged in shift register form. The input to first [T] element is connected to one of the S line. While the input to [SA/ SS] is connected from input S* and/or one of the output of [T] elements of shift register, depending on the coefficient value. Finally, using SAe_1 to SAe_n-1 elements, the addition/subtraction of [SA/SS] of all the coefficient terms depicted in columns is done. The final output is the output of last addition/subtraction[SA/SS].

- 9 -

Among the lines S1 to Sn, the [T] elements are not share-able and also the [SA] in each column are also not share-able. Thus limited minimization is possible in this structure.

DETAILED DESCRIPTION OF THE INVENTION

The device reduces the hardware of the coefficient block [A] by having share-able elements in coefficient block [A], even in the implementation where the coefficient lines CLin_0, CLin_1,.....are not commonly connected (shown as architecture [A]). This reduced hardware in coefficient block when applied in implementation2 ("Figure 4") makes it still more area- efficient. This reduction is extendable to other filter based on coefficient block [A], as stated in the first section.

The device of the present invention reduces the area by approximately 30- 50% of "Figure 7" of the drawings by reducing the number of components. The optimization techniques are illustrated mathematically and towards the end of this section generalized equation and structure of the device is presented.

Accordingly, the present invention (Fig. 13) provides a device for providing an area efficient realization of coefficient, said device comprising of architecture [A] with hardware sharing techniques and optimization applied to this architecture, the said architecture [A] is connected to coefficient lines CLin_0, CLin_1.....CLin_n and/or BLin_0, BLin_1,....BLin_n coming from block [E] and/or [F], to be connected to perform filtering operation or a mathematical computing operation with optimization in hardware and provides a zero latency output, the said architecture [A] has serial input bit lines as S1, S2,.....Sn. [where n represents the number of coefficients of the filter] and the addition terms of the equation $[(a_0*S_1+b_0*S_2+....+k_0*S_n), (a_1*S_1+b_1*S_2+.....+k_1*S_n).....(a_m*S_1+b_m*S_2+.....+k_m*S_n)]$ are represented as blocks [B] , the values of a0, b0,..... etc. are represented as [(+ /-)1 or 0], the said

- 10 -

Block [B] is a combinational-sequential block consisting of serial adders (SA) & serial subtractors (SS) elements, the connection of elements (SA/SS) to S1, S2....Sn lines and interconnection of the elements (SA, SS) depend on the value of coefficients, the SA/SS elements are arranged in matrix form SA0_0 to SA0_n in bit position 0 and SA1_1 to SA1_n in bit position 1 and similarly SAM_1 to SAM_n in bit position m, the presence of one of these elements is defined by coefficient value, the output of each block [B] is connected to [T] elements through line b_1, b_2,....b_m, the number of T elements depends on the size of maximum coefficient and is share-able for all the coefficient in the coefficient architecture [A], the output of element [T] is connected to one of the inputs of combinational logic of block [B] of next bit position (i.e connected to input of element (SA or SS) of block [B] depending upon the sign value+/-), Lines t_1, t_2,.....t_m are used to mark the interconnections from cluster [C] to [B], in the said structure [A], all the elements in block [B] are clustered together as block [D] and all the unit delay elements {T[1], T[2].....T[m]} are clustered together in block [C], thereby separating the combinational-sequential and sequential logic, while the sequential elements [T] of block [C] are common for all the coefficients and are share-able and positioned at end position of each block [B], the Block [D] has combinational-sequential element block [B] which are essentially SA, SS, In the structure the hardware within block [B] are share- able across various [B] blocks and also within block [D]. The final output is taken from the output of the elements of last bit position.

An embodiment of the invention or Optimization (a)

Continuing the same example of FIR filter & using "Equation 3" of previous section.

$$y(nT) = 5 * S1 + 14 * S2 + 25 * S3 + 30 * S4$$

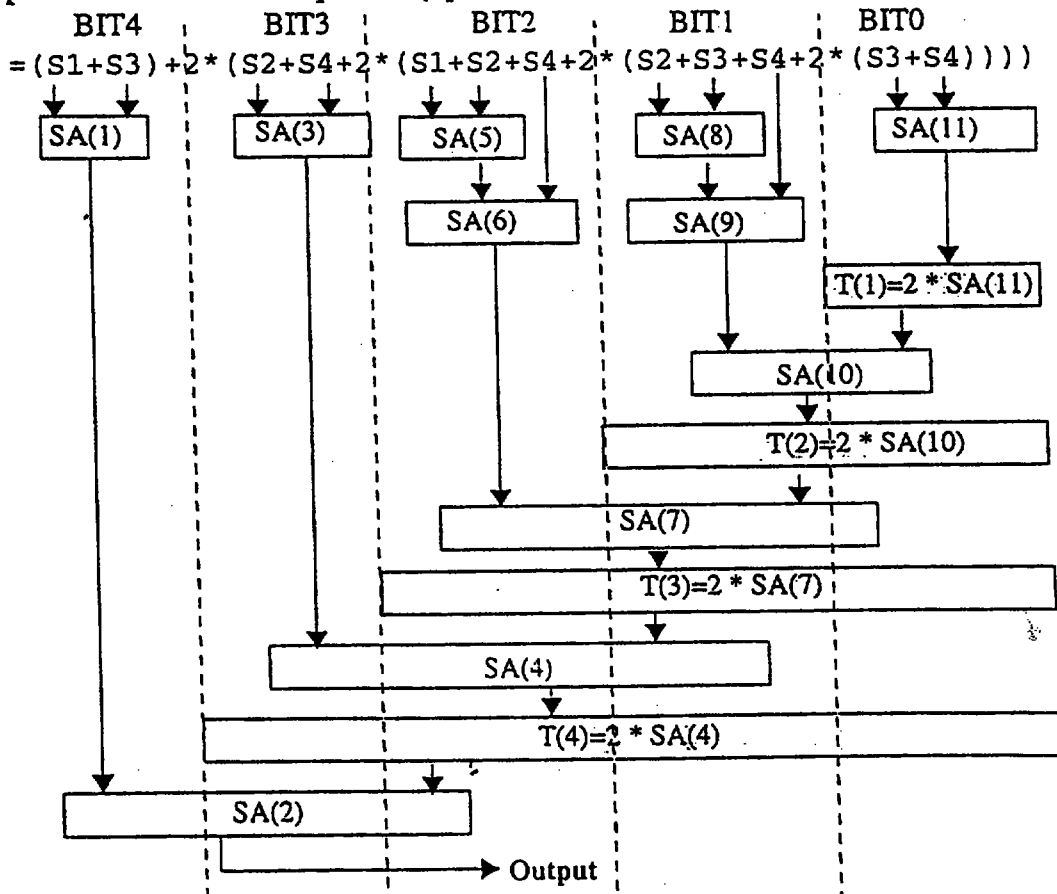
$$Y(nT) = (4+1)S1 + (8+4+2)S2 + (16+8+1)S3 + (16+8+4+2)S4$$

we proceed to share the shift registers (multiply by 2) of the design.

$$\begin{aligned}
 &= (S3+S4)*16+(S2+S3+S4)*8+(S1+S2+S4)*4+(S2+S4)*2+(S1+S3) \\
 &= (S1+S3)+2*(S2+S4+2*(S1+S2+S4+2*(S2+S3+S4+2*(S3+S4)))) \quad (EQ\ 4)
 \end{aligned}$$

The implementation flow for this equation is presented below this text paragraph and the hardware implementation is shown in "Figure 10" of the drawings. In the flow of implementation, S1, S2, S3, S4 represents four inputs. The primary additions are done using serial adders SA(1), SA(3), SA(5), SA(8), SA(11) representing addition of terms S1+S3, S2+S4, S1+S2, S2+S3, S3+S4. While the secondary and tertiary additions are done using the adders SA(6), SA(9), SA(10), SA(7), SA(4), SA(2). The multiplication by factor of two is done using the elements T(1), T(2), T(3), T(4).

Implementation flow of equation {optimization(a)}



This Sheet Has Been Replaced



As shown in the above implementation flowchart, the equation defines the bit position as BIT0 to BIT4, which is the position of "multiplication by power of two". (e.g BIT0 represents multiplication by 2^0). At BIT0 position addition of $S3+S4$ is performed and the output is terminated at T(1). The output of T(1) defines the next bit position BIT1, which performs addition of $S2+S3+S4$ and output of T(1) by using the [SA]. The output of this addition is again terminated at T(2). The structure is repeated in next BIT positions. The final addition of BIT position BIT4 gives the output of the coefficient block [A].

Implementation of hardware is shown in "Figure 10", wherein the input line S1 to S4 represent the lines connected to delay block $[Z^{-1}]$ through coefficient line Clin_0 to CLin_6 depicted in "Figure 6" of the drawings. The Lines S1 to S4 are connected to block [B] for performing the serial addition/ subtraction, for which [SA], [SS] elements are used within block[B]. The input to [B] block is connected to line S1 to S4 and also from [T] elements as would be explained later in this section. The output of each block [B] is terminated with [T] element, which represents the block [B] output being multiplied by "a factor of 2". Each [T] elements defines bit position marked as BIT1, BIT2, BIT3, BIT4. The output b_1 of block [B] which is at bit position BIT0 is fed to the input of the T(1), in turn the output line t_1 of element [T(1)] is fed to next section of block[B]. The connectivity is done in similar fashion for other [T] blocks. Thus all addition/subtraction in block [B] defines a bit position before getting multiplied by "a factor of 2" and changing to next bit position. The block [B] at final bit position represents the output of the coefficient block [B].

In the structure, all [T] elements are represented as block[C] wherein the flip- flop [T] representing multiplication by "a factor of 2", is share-able among various

coefficient values and their number is determined by maximum coefficient value. This is in contrast to "Figure 7" of existing structure where the elements are not share-able between S1 to S4 lines.

The number of flip-flops (T) in "Figure 7" is 13 vs. number of flip-flops(T) in current proposal is 4. Also for both the implementation, the number of the one-bit serial adders (SA) remains the same [11 in each case]. In present minimization, approximate area calculations is = 11 serial adder + 4 T = 26 Units, whereas the area after previous minimization is 11 serial adder + 13 T = 35 units. (assuming 1 Unit = 1 FA = 2HA = 1T and serial adder/serial subtractor (SA/SS) = 2 Units). This is approximately 26% saving in area in "Figure 10" as compared to "Figure 7".

For filter having large size coefficient, this leads to a drastic reduction in the area (30-50% of the coefficient block).

Another embodiment of the invention or Optimization (b)

This optimization reduces the hardware of block [D] which essentially consists of (SA) and (SS) elements. Beginning with "Equation 4" and finding out the common additive factors

$$A1 = S2 + S4$$

$$A2 = S3 + S4$$

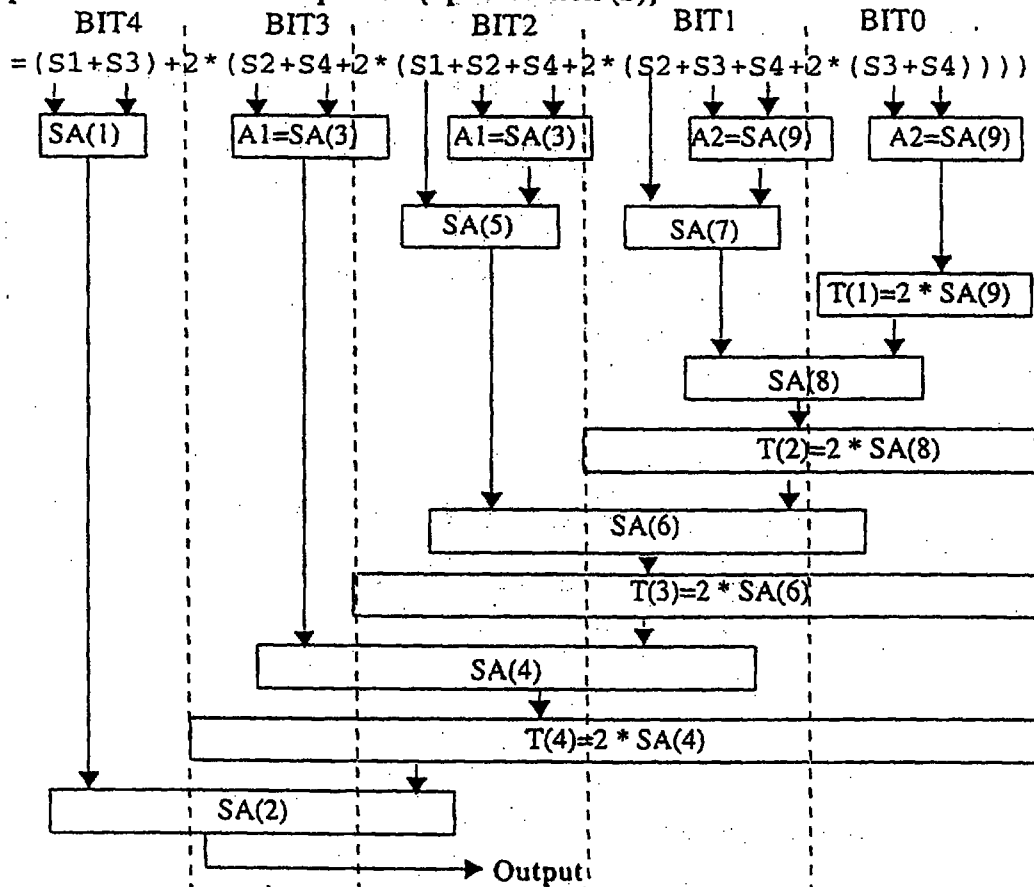
The "Equation 4" can be further reduced as

$$y(nT) = (S1 + S3) + 2 * (A1 + 2 * (S1 + A1 + 2 * (S2 + A2 + 2 * A2))) \quad (\text{EQ 5})$$

The flow of implementation of equation is illustrated below and is self explanatory. Here S1, S2, S3, S4 represents four inputs. The primary addition is done using serial adders SA(1), SA(3), SA(9) representing addition of terms S1+S3, S2+S4, S3+S4. While the secondary and tertiary addition is done using the

adders SA(5), SA(7), SA(8), SA(6), SA(4), SA(2). The multiplication by factor of two is done using the elements T(1), T(2), T(3), T(4).

Implementation flow of equation (optimization (b))



This Sheet Has Been Replaced



"Figure 11" shows the implementation of the structure, wherein the input line S1 to S4 represent the lines, connected to delay block $[Z^{-1}]$ through coefficient lines CLin_0 to CLin_6 depicted in "Figure 6" of the drawings. The Lines S1 to S4 are connected to block [B] for performing the serial addition/subtraction, for which (SA), (SS) elements are used within block[B]. The input to [B] block is from line S1 to S4 and also from [T] elements. The output of each block [B] is terminated with a [T] block, which represents the block [B] output being multiplied by factor of 2. The output b_1 of block [B] which is at bit position BIT0, is fed to the input of the T(1), in turn the output t_1 of [T(1)] is fed to next section of block[B]. Thus all addition defines a bit position before getting multiplied by factor of 2. All such [T] termination is represented by block[C].

The optimizations in reducing the hardware of block [D] are done. The output b_1 representing the bit position BIT0 and addition term A2, is connected to T[1] and also fed to the next block [B], hence reducing the adder count by 1. Also the output of adders SA(3) of block [B] in bit position BIT3, is fed at two points. One to the input of adders SA(4) which eventually terminates at [T4] element and other to the input of adder SA(5), hence reducing the adder count further by 1.

Note how A1 and A2 are shared in the Structure. Comparing the hardware implementation of "Figure 10" and "Figure 11", the number of adders are minimized by having common adders A1, A2. This optimization is purely dependent on finding common addition terms among coefficients.

In present minimization, approximate area calculations is = 9 serial adder (SA) + 4 (T) = 22 Units, whereas the area of the existing minimization of "Figure 7" is 11 (SA) + 13 (T) = 35 units. (assuming 1 Unit = 1 FA = 2HA = 1T and serial adder (SA) = 2 Units). Thus compared to the existing minimization, the "Optimization (a)" and "Optimization (b)", combined has resulted in 37% saving in area (13 / 35

* 100). "Optimization (b)" is an improvement of 15% in area (of the coefficient block) over "Optimization (a)"

Yet another embodiment of the invention or Optimization (c)

In realization of block [D], further optimization is done by realizing the coefficient value using subtraction instead of addition. This is good for numbers which have values closer to power of 2. (e.g. for realization of coefficient value 63, the realization $(63=64-1)$ is better than $(63=32+16+8+4+2+1)$. In first case the number of subtractor is 1 while in second case the number of adders are 5.

To illustrate this by an example, consider the coefficient values as 5, 25, -48, -63). Writing the FIR equation using these coefficient values.

Arranging the terms with 63 as $(32+16+8+4+2+1)$

$$\begin{aligned} y(nT) &= 5 * S1 + 25 * S2 - 48 * S3 - 63 * S4 \\ &= (1+4)*S1 + (16+8+1)*S2 - (32+16)*S3 - (32+16+8+4+2+1)*S4 \\ &= (S1+S2-S4)+2*(-S4+2*(S1-S4+2*(S2-S4+2*(S2-S4-(S3+2(S3+S4)))))) \quad (\text{EQ 6}) \end{aligned}$$

Alternately arranging the terms with 63 as $(64-1)$, the first equation reduces

$$\begin{aligned} &= (1+4)*S1 + (16+8+1)*S2 - (32+16)*S3 + (1-64)*S4 \\ &= (S1+S2+S4)+2*(2*(S1+2*(S2+2*(S2-S3-2*(S3+2*S4)))) \quad (\text{EQ 7}) \end{aligned}$$

The realization of "Equation 6" and "Equation 7" is shown in "Figure 9" and "Figure 12" respectively.

In these realizations, the number of [T] elements is one more in "Equation 7" due to presence of term "64". However, the number of adders are less in the structure represented by "Equation 7" than by "Equation 6". This is because the number of adders are less in former case. Comparing the area of the two realization, From "Equation 6", Area obtained is $5 T+6 SA+6 SS = 29$ Units. While from

"Equation 7" , representing optimization(c), results an area calculation of $= 6T + 6SA + 2SS = 22$ Units. (assuming 1 Unit = 1 FA = 2HA = 1T and SA=SS= 2 Units). This is an improvement by 24% in reducing area of coefficient block for the current example.

Thus, the optimization reduces area for realization of negative coefficient. This optimization is also efficient realization of coefficients having value close to power of two. Further minimization is possible by taking subtraction as common factor and using addition instead of subtraction wherever possible in the realization. This result in an improvement in area, due to the fact that area for a subtractor is more than the area of an adder.

GENERALIZED STRUCTURE OF THE INVENTION

The invention provides an area efficient realization of filter coefficient block[A] applicable to filters devices such as FIR, IIR and other filter structures). This architecture is also applicable to combinational and sequential logic consisting of adder, subtractors, multipliers and flip flop [T]. This architecture is realized using the elements serial adders (SA), serial subtraction (SS) and flip-flop [T].

Generalized structure of the present invention is depicted in "Figure 13". The generalized equation of the structure is also calculated here.

Beginning with the generalized equation of FIR filter coefficient block(A)

$$y(nT) = a * S1 + b * S2 + c * S3 +k * Sn.....(1)$$

where a, b,....k represents filter coefficients. S1, S2..... represents bit lines corresponding to the coefficients.

Now, representing each coefficient as addition of terms arranged in power of two and applying it to the equation:

$$y(nT) = (2^m \cdot a_m + \dots 2^1 \cdot a_1 + 2^0 \cdot a_0) \cdot S_1 + (2^m \cdot b_m + \dots 2^1 \cdot b_1 + 2^0 \cdot b_0) \cdot S_2 + (2^m \cdot c_m + \dots 2^1 \cdot c_1 + 2^0 \cdot c_0) \cdot S_3 + \dots + (2^m \cdot k_m + \dots 2^1 \cdot k_1 + 2^0 \cdot k_0) \cdot S_n$$

Further, taking "2" as common factor we get the generalized equation for architecture as:

$$\begin{aligned} Y(nT) = & (a_0 \cdot S_1 + b_0 \cdot S_2 + \dots + k_0 \cdot S_n) \\ & + 2^1 (a_1 \cdot S_1 + b_1 \cdot S_2 + \dots + k_1 \cdot S_n) + \\ & 2^1 ((a_2 \cdot S_1 + b_2 \cdot S_2 + \dots + k_2 \cdot S_n) + \\ & 2^1 ((a_3 \cdot S_1 + b_3 \cdot S_2 + \dots + k_3 \cdot S_n) + \dots + \\ & 2^1 ((a_m \cdot S_1 + b_m \cdot S_2 + \dots + k_m \cdot S_n)))) \end{aligned}$$

where $a_0, a_1, \dots, a_m$ and $b_0, b_1, \dots, b_m$ and $k_0, k_1, \dots, k_m$ represents the sign of coefficients [i.e they have value (+ / - 1) or 0]. The architecture realization in "Figure 13" is done using the sequential elements like unit delays [T] and combinational elements such as serial adder (SA) and serial subtractor (SS). The highlights of the architecture are:

- 1) A common share-able [T] elements for all the coefficients. The maximum number of [T] element is equal to next integer value of "log of the maximum value of coefficient" in the coefficient block [A]
- 2) Area-optimizations in reducing the combinational logic [D] (i.e optimizations applied on serial adders(SA), serial subtractor (SS) as stated in previous section).

In "Figure 13", the input serial data is present on bit line $S_1, S_2, \dots, S_n$. [where n represents the number of coefficients of the filter] The addition terms of the equation. $[(a_0 \cdot S_1 + b_0 \cdot S_2 + \dots + k_0 \cdot S_n), (a_1 \cdot S_1 + b_1 \cdot S_2 + \dots + k_1 \cdot S_n), \dots, (a_m \cdot S_1 + b_m \cdot S_2 + \dots + k_m \cdot S_n)]$ are represented as blocks [B]. Block [B] is a combinational-sequential block consisting of serial adders (SA) and serial subtractors (SS) elements. Since the values $a_0, b_0, \dots$ etc. represents value $[(+ / -)1$ or 0]. The connection of elements (SA/SS) to $S_1, S_2, \dots, S_n$ lines and interconnection of the elements (SA, SS) depend on the value of coefficients [This is because the value of

coefficient determines the value of $a_0, a_1, \dots$ etc. and hence it defines the interconnections between them]. The output of each block [B] is multiplied by two using [T] elements {The elements $T[1], T[2], \dots, T[m]$ are used for multiplication by factor of 2}. The number of T elements depends on the size of maximum coefficient and is share-able for all the coefficient in the coefficient block [A]. Thus in the structure the final outputs of all the blocks [B] are terminated at unit delay elements [T] (connected through $b_1, b_2, \dots, b_m$).

In the structure, all the elements [B] are clustered together as [D] and all the unit delay elements $\{T[1], T[2], \dots, T[m]\}$ are clustered together in [C]. The sequential [C] and combinational-sequential logic [D] are quite separated in this architecture. The input of the unit delay element [T] is final output of block [B] and the output of element [T] is connected to the one of the inputs of combinational logic of block [B] of next bit position (i.e connected to input of element (SA or SS) of block [B] depending upon the sign value+/-). The interconnections from cluster [C] to [B] is represented as $t_1, t_2, \dots, t_m$. The bit positions of serial data frame are marked as BIT0, BIT1, ..., BITm.

In the generalized structure, flip-flops[T] of all the coefficient are share-able and the number of flip-flops[T] are limited to the coefficient which has the maximum value. Also optimization can be applied in block [D]. The gain in area when compared with the existing design is illustrated below.

Hardware reduction in block [C]

Before beginning to prove the statement, we proceed to formularize the calculation of number of flip-flop (T) for structure of "The Existing Method & Minimization" in "Figure 7" and "Figure 8" of the drawings. The number of flip-flops in the coefficient block depends on the size of all the coefficients. The approximate and

- 20 -

pessimistic formula for calculation of total flip-flops (T) in coeff. block is [= average size of coefficient * number of coefficient], where average size of coefficient can be calculated pessimistically as (Maximum coefficient size / 2). (Refer "The Existing Method and Minimization", "Figure 8" of the drawings). Applying this formula, to example of "Figure 7" for verification, where coefficient (5, 14, 25, 30) are represented in 4,5,6,6 bits (using signed notation). According to formula, average size of coefficient is $(6 / 2) = 3$ and total number of flip-flop = $3 * 4 = 12$. This is pessimistic as compared to the implementation (where total number of flip-flops are 13, refer "Figure 7").

Similarly, the approximate formula for calculation of total adders (SA) in coeff. block for "The Existing Method and Minimization", and "Detailed Description of the Invention" in "Figure 8" and "Figure 13" is [=adders per coefficient * number of coefficient]. Adders per coefficient solely depend on the value of coefficient. Assuming number of adders as (=number of coefficients * maximum coefficient size / 2).

Now, as an example, the Applicants provide herebelow use of the above mentioned formulae of previous two paragraphs in filter of 20 coefficient. Assume the maximum coefficient value is represented in 16 bits (e.g. maximum coefficient value is +32767 or -32768 in 2's complement representation). Average size of the coefficient approximated by the formula is $(16/2=8 \text{ bit})$. In "The Existing Method and Minimization", the total number of flip-flop (T) required for implementation is $8 * 20 = 160$. In contrast to this "Detailed Description of the Invention" would require only 16 Flip-Flops (The number of flip-flops of all the coefficient are share- able and are limited to the coefficient which has the maximum value). Assuming in worst case that there is no optimization of adders, the number of

adders in both the cases are same and are equal to $8 * 20 = 160$. (Refer "Figure 8" and "Figure 13" of the drawings)

Area calculation for "The Existing Method and Minimization", "Figure 8" of the drawings is $160 T + 160 SA = 480$. Area calculation for "Detailed Description of the Invention", "Figure 13" is $16 T + 160 SA = 336$. This is an improvement of 30% $[(480-336)/480]$ over "The Existing Method and Minimization". (assuming 1 Unit = 1 FA = 2HA = 1T and serial adder/serial subtractor (SA/SS) = 2 Units). The area gain by structure could be as high as 50% (of the coefficient block) for big filter where minimization of adders and other minimization optimization(a), optimization(b) and optimization(c), as discussed earlier, are applied.

The preferred embodiment of the invention is also supported by a real example of filter coefficient device. This is referred to as optimization(a) and shown in "Figure 10" of the drawings and discussed in previous section. Area calculation for "Figure 10" is $11SA + 4T = 26$ units while area of "Figure 7" of the drawings (existing implementation) is $11SA + 13T = 35$ units. This results a gain of 26% in area (of coefficient design block) for this example design as compared to existing implementation, supporting the generalized statement.

Hardware reduction in block [D]

The optimizations in block [D] are referred to as optimization (b), optimization (c) and shown in "Figure 11", "Figure 12" of the drawings. In optimization (b), beside sharing the flip-flop (T) for all the coefficients and sharing the common adders (SA) techniques is done in cluster [D]. This is due to presence of (SA), (SS) in block [D] and separate clustering of elements in block [D] and [C].

- 22 -

This is illustrated using the example of previous section Using $A1 = S2+S4$, $A2 = S3+S4$ in this example. Total area for this example after the optimization is $9SA + 4T = 22$ Units*. This, when combined with optimization(a) results in area-gain of 37% in area (of coefficient design block) for this example design as compared to existing implementation (where area was 35 units).

The optimization(c) as described before can be further applied to cluster [D]. That is beside optimization(a) and optimization(b), the technique of realizing the coefficient value using subtraction (SS) instead of addition(SA) is used here. This saves quite in area when the coefficient value is close to power of 2. (e.g. for realization of coefficient value 63, the realization ($63=64-1$) is better than ($63 = 32+16+8+4+2+1$). In first case the number of subtractor is 1 while in second case the number of adders are 5.). The two cases are illustrated in previous section with example and are shown in "Figure 12" and "Figure 9" of the drawings. The area calculation without optimization and using adders ("Figure 9" of the drawings) = 29 Units*. While optimization applied ("Figure 12" of the drawings) results in area = 22 Units*. This is an improvement by 24% in reducing area of coefficient block for this example.

With all the optimization applied, the invention while in use results in an area improvement of 30-50% of the coefficient block design or combinational logic consisting of adders, subtractor, multiplier and unit delays [T].

*Note that the input to adders in [B] are interchangeable e.g adders SA(5), SA(6) inputs could be interchanged. Also the signals t_1 , t_2 etc. can be connected to any input of adders of block [B] of next bit position.

This Sheet Has Been Replaced



- 23 -

****For approximate area calculation following assumption is made (1 Unit of Area = 1 FA = 2HA = 1T & SA=SS= 2 Units of Area).**

The present invention is most economical in terms of area of coefficient block/ architecture. In fact, the present invention provides an area improvement of 30-50% of the coefficient block design or combinational logic consisting of adders, subtractor, multiplier and unit delays.

This Sheet Has Been Replaced



SUBSTITUTE SHEET (RULE 26)



CLAIMS

1. A device for providing an area efficient realization of coefficient, said device comprising of architecture [A] with hardware sharing techniques and optimization applied to this architecture, the said architecture [A] is connected to coefficient lines CLin_0, CLin_1.....CLin_n and/or BLin_0, BLin_1,....BLin_n coming from block [E] and/or [F], to be connected to perform filtering operation or a mathematical computing operation with optimization in hardware and provides a zero latency output, the said architecture [A] has serial input bit lines as S1, S2,.....Sn. [where n represents the number of coefficients of the filter] and the addition terms of the equation $[(a_0*S1+b_0*S2+....+k_0*Sn), (a_1*S1+b_1*S2+.....+k_1*Sn).....(a_m*S1+b_m*S2+.....+k_m*Sn)]$ are represented as blocks [B], the values of $a_0, b_0,.....$ etc. are represented as $[(+/-)1 \text{ or } 0]$, the said Block [B] is a combinational-sequential block consisting of serial adders (SA) and serial subtractors (SS) elements, the connection of elements (SA/SS) to S1, S2.....Sn lines and interconnection of the elements (SA, SS) depend on the value of coefficients, the SA/SS elements are arranged in matrix form SA0_0 to SA0_n in bit position 0 and SA1_1 to SA1_n in bit position 1 and similarly SAM_1 to SAM_n in bit position m, the presence of one of these elements is defined by coefficient value, the output of each block [B] is connected to [T] elements through line b_1, b_2,....b_m, the number of T elements depends on the size of maximum coefficient and is share-able for all the coefficient in the coefficient architecture [A], the output of element [T] is connected to one of the inputs of combinational logic of block [B] of next bit position (i.e connected to input of element (SA or SS) of block [B] depending upon the sign value+/-), Lines t_1, t_2,.....t_m are used to mark the interconnections from cluster [C] to [B], in the said structure [A], all the elements in block [B] are clustered together as block [D] and all the unit delay elements $\{T[1], T[2].....T[m]\}$ are clustered together in





block [C], thereby separating the combinational-sequential and sequential logic, while the sequential elements [T] of block [C] are common for all the coefficients and are share-able and positioned at end position of each block [B], the Block [D] has combinational-sequential element block [B] which are essentially SA, SS, In the structure the hardware within block [B] are share- able across various [B] blocks and also within block [D]. The final output is taken from the output of the elements of last bit position.

2. A device as claimed in claim 1 wherein the area minimal realization of digital filters based on coefficient architecture[A], when operated in bit serial fashion, the structure [A] provides hardware minimization for finite impulse response(FIR) filter, infinite impulse response filter(IIR) and for other applications related to combinational logic consisting of delay element(T), multiplier(M), adder(SA) and subtractor (SS).

3. A device as claimed in claim 1 wherein further optimization technique in cluster [D] is done by using common adders (SA) and common subtractor (SS) in Block [B] and using this shared outputs.

4. A device as claimed in claim1 wherein further optimization technique in cluster [D] is done by using subtractor (SS) instead of adders, when the coefficient value is closer to power of two.

5. A device as claimed in claim 1 wherein further optimization technique in cluster [D] is done by minimizing the use of subtractor by taking common subtraction operator and using adder instead.

This Sheet Has Been Replaced





6. A device as claimed in anyone of the preceding claims wherein when used in implementation 2 of FIR/IIR filter and similar structure of filters, results in quite area efficient realization of filter, this is due to the fact that storage area of delay elements of filter in implementation2 is smaller in size as compared to implementation 1 which is present due to inherent property of the structure of implementation 2, An additional saving in area in filter coefficient realization design is achieved by using the claimed structure of coefficient architecture [A] of "Figure 13"

This Sheet Has Been Replaced

