PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 : G06F 15/353	A1	(11) International Publication Number: WO 92/15065
		(43) International Publication Date: 3 September 1992 (03.09.92)

(21) International Application Number: PCT/US92/00902

(22) International Filing Date: 5 February 1992 (05.02.92)

(30) Priority data:
651,738 7 February 1991 (07.02.91) US

(71) Applicant: INTEL CORPORATION [US/US]; 3065 Bower Avenue, Santa Clara, CA 95052 (US).

(72) Inventor: SPRAGUE, David, Leroy ; 10 Jacobs Creek Road, Trenton, NJ 08628 (US).

(74) Agents: MURRAY, William, H. et al.; Schnader, Harrison, Segal & Lewis, 1600 Market Street, Suite 3600, Philadelphia, PA 19103 (US).

(81) Designated States: AT, AT (European patent), AU, BB, BE (European patent), BF (OAPI patent), BG, BJ (OAPI patent), BR, CA, CF (OAPI patent), CG (OAPI patent), CH, CH (European patent), CI (OAPI patent), CM (OAPI patent), CS, DE, DE (European patent), DK, DK (European patent), ES, ES (European patent), FI, FR (European patent), GA (OAPI patent), GB, GB (European patent), GN (OAPI patent), GR (European patent), HU, IT (European patent), JP, KP, KR, LK, LU, LU (European patent), MC (European patent), MG, ML (OAPI patent), MN, MR (OAPI patent), MW, NL, NL (European patent), NO, PL, RO, RU, SD, SE, SE (European patent), SN (OAPI patent), TD (OAPI patent), TG (OAPI patent).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: ONE-DIMENSIONAL INTERPOLATION CIRCUIT AND METHOD BASED ON MODIFICATION OF A PARALLEL MULTIPLIER

(57) Abstract

An interpolator array having a plurality of interpolator array cells is provided for receiving first and second input values to be interpolated and an interpolator weight term, to provide an interpolated output. A bit of each of the two input values to be interpolated is received by an interpolator array cell and applied to a selecting circuit within a cell of the interpolator array. Additionally, an interpolation weight bit of the interpolation weight term is applied to the selection circuit. The selecting circuit applies either the input bit of the first input value or the input bit of the second input value to an adder within the interpolator cell in accordance with the value of the interpolation weight bit. An interpolator array cell also receives a partial product input and a carry-in input and applies these additional inputs to the adder. The adder provides a partial product output and a carry-out in accordance with the applied inputs.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	**FI**	Finland	**ML**	Mali
AU	Australia	**FR**	France	**MN**	Mongolia
BB	Barbados	**GA**	Gabon	**MR**	Mauritania
BE	Belgium	**GB**	United Kingdom	**MW**	Malawi
BF	Burkina Faso	**GN**	Guinea	**NL**	Netherlands
BG	Bulgaria	**GR**	Greece	**NO**	Norway
BJ	Benin	**HU**	Hungary	**PL**	Poland
BR	Brazil	**IE**	Ireland	**RO**	Romania
CA	Canada	**IT**	Italy	**RU**	Russian Federation
CF	Central African Republic	**JP**	Japan	**SD**	Sudan
CG	Congo	**KP**	Democratic People's Republic of Korea	**SE**	Sweden
CH	Switzerland			**SN**	Senegal
CI	Côte d'Ivoire	**KR**	Republic of Korea	**SU**	Soviet Union
CM	Cameroon	**LI**	Liechtenstein	**TD**	Chad
CS	Czechoslovakia	**LK**	Sri Lanka	**TG**	Togo
DE	Germany	**LU**	Luxembourg	**US**	United States of America
DK	Denmark	**MC**	Monaco		
ES	Spain	**MG**	Madagascar		

ONE-DIMENSIONAL INTERPOLATION CIRCUIT AND METHOD BASED ON MODIFICATION OF A PARALLEL MULTIPLIER

Background of the Invention

1. Field of the Invention

This invention relates to the field of one-dimensional interpolation and in particular to one-dimensional interpolation requiring fewer mathematical operations or fewer interpolator circuit elements.

2. Background of the Invention

It is well known in the art to perform one-dimensional interpolation. One-dimensional interpolation involves the weighted summation of two values, for example, as expressed by the equation: $I = xA + (1 - x)B$. In this equation A and B are the two input values to be interpolated and x is the fractional weight term. The solution of this equation requires two add/subtract operations and two multiplications.

It is known to rearrange this equation to reduce it to the following form: $I = x (A - B) + B$. When the basic

one-dimensional interpolation equation is rearranged into this form, the solution of the interpolation requires one subtraction, one addition, and one multiplication. Thus this rearranged form requires one less multiplication. Because this rearranged form requires fewer mathematical operations, it is advantageous to design a circuit to solve the equation in this rearranged form. This advantage can be realized in the form of decreased space requirements on the semiconductor chip or in performing the interpolation more quickly using the same amount of space. However, it is desirable to further reduce the amount of space or time required to perform the interpolation.

It is also well-known in the art to provide parallel multipliers for multiplying binary numbers. A parallel multiplier is based on the observation that partial products in the multiplication process can be independently computed in parallel. For example, consider the unsigned binary integers A and B:

$$A = \sum_{i=0}^{m-1} A_i 2^i$$

$$B = \sum_{j=0}^{n-1} B_j 2^j .$$

The product is found by:

$$P_r = A_y B_r = \sum_{i=0}^{m-1} A_i 2^i \cdot \sum_{j=0}^{n-1} B_j 2^j$$

$$= \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (A_i B_j) 2^{i+j}$$

$$= \sum_{k=0}^{m+n-1} P_k 2^k$$

Thus P_k are the partial product terms called the summands. There are mn summands, which are produced in parallel by a set of mn AND gates. For four bit values A and B, the expression above may be expanded into the partial products or summands set forth in Table 1.

				A_3B_0	A_2B_0	A_1B_0	A_0B_0
			A_3B_1	A_2B_1	A_1B_1	A_0B_1	
		A_3B_2	A_2B_2	A_1B_2	A_0B_2		
	A_3B_3	A_2B_3	A_1B_3	A_0B_3			
P7	P6	P5	P4	P3	P2	P1	P0

TABLE 1

An n X n multiplier requires n(n-2) full adders, n half adders and n^2 AND gates. The worst case delay associated with such a multiplier is $(2n+1)\tau_g$, where τ_g is the worst case adder delay.

Figs. 1, 2 show parallel multiplier cell 10 for forming parallel multiplier array 30 to provide, for example, the partial products or summands of Table 1. The A_i term of value input line

16 of parallel multiplier cell 10 is propagated vertically, through cell 10. The B_j term of value input line 18 of parallel multiplier cell 10 propagates horizontally through cell 10. Incoming partial product term P_i enters at the top left of parallel multiplier cell 10 by way of partial product input line 12. Incoming carry-in values enter at the top left of parallel multiplier cell 10 by way of carry-in line 14.

A bit-wise AND of A_i on value input line 16 and B_j on value input line 18 is performed in parallel multiplier cell 10 by AND gate 22. The output of AND gate 22 and the incoming partial product term P_i of partial product input line 12 are applied to adder 20 or summer 20. The output sum value of adder 20 is passed to a diagonally adjacent parallel multiplier cell 10 in the next row and the next column of parallel multiplier array 30 by way of partial product output line 24 at the lower right of parallel multiplier cell 10. The carry-out term of adder 20 is passed to the next vertically adjacent cell below parallel multiplier cell 10 in parallel multiplier array 30 by way of carry-out line 26.

Parallel multiplier array 30 therefore is adapted to provide partial products, P0-P7, as set forth in Table 1. Thus parallel multiplier 30 is adapted to multiply two four-bit binary values. The partial product terms of the multiplication, P0-P7, are provided at parallel multiplier array 30 output lines 24. To perform the multiplication of input value A and input value B,

the four bits of value A are applied to respective value input lines 16 of parallel multiplier array 30. The four bits of value B are applied to respective value input lines 18 of multiplier array 30. As show in Fig. 3, parallel multiplier array 30 may be redrawn as a square parallel multiplier array 30. A square parallel multiplier array 30 is the most convenient for implementation. Many other configurations for performing multiplication of binary values are also known in the art.

SUMMARY OF THE INVENTION

An interpolator array having a plurality of interpolator array cells is provided for receiving first and second input values to be interpolated and an interpolation weight term to provide an interpolator output. A bit of each of the two input values to be interpolated is received by an interpolator array cell and applied to a selection circuit within a cell of the interpolator array. Additionally, an interpolation weight bit of the interpolation weight term is applied to the selection circuit. The selection circuit applies either the bit of the first input value or the bit of the second input value to an adder within the interpolator cell in accordance with the value of the interpolation weight bit.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a schematic representation of a prior art cell which may be used to form a parallel multiplier.

Fig. 2 shows a block diagram representation of a prior art parallel multiplier array formed of a plurality of the prior art cells of Fig 1.

Fig. 3 shows a block diagram representation of the parallel multiplier array of Fig. 2 wherein the cells are arranged as a square array.

Fig. 4 shows a schematic representation of the one-dimensional interpolator cell of the present invention which forms the one-dimensional interpolation circuit of the present invention.

Fig. 5 shows a schematic representation of the one-dimensional interpolation circuit of the present invention formed of a plurality of the interpolator cells of Fig. 4 arranged as a square array.

Fig. 6 shows a flow chart representation of a prior art method of performing a multiplication of two binary values.

Fig. 7 shows a flow chart representation of the one-dimensional interpolation method of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

One-dimensional interpolation involves the weighted sum of two values A and B as expressed by the equation:

$$I = xA + (1 - x)B \qquad (1)$$

where A and B are two binary values to be interpolated and x is the fractional weight term of the interpolation. Equation (1) requires two add/subtract operations and two multiplications. The one-dimensional interpolation of Equation (1) can be reduced to:

$$I = x(A - B) + B. \qquad (2)$$

The x(A - B) term of Equation (2) provides the offset of the interpolated value between the two input values A and B. The addition of B translates the offset value to provide the interpolation result.

The one-dimensional interpolation of Equation (2) requires one less multiplication than the interpolation of Equation (1) thereby permitting the interpolation of Equation (2) to be solved more quickly or with less circuitry than the interpolation of Equation (1). However, using the system of the present invention, the solution of a one-dimensional interpolation is further speeded up compared with the interpolation of Equation (2). In the system of the present invention, Equation (2) is further algebraically manipulated before it is implemented.

Assuming that the fractional weight term of the interpolation, x, is a four bit unsigned binary number, Equation (1) can be scaled by sixteen to give:

$$16I = 16xA - 16xB + 16B \quad (3)$$

$$= 16xA + (16 - 16x)B$$

$$= yA + (16 - y)B \quad (4)$$

where:

$$y = 16x \quad (5)$$

and y is a four bit unsigned integer with values from zero to fifteen. This scaling by sixteen can be undone by selecting the proper bits of the result of the interpolation. Thus y may be used as the interpolation weight term in the method and system of the present invention provided the proper bits of the result are selected.

Note that since interpolation weight term y is a four bit positive integer,

$$16 - y = \alpha y + 1, \quad (6)$$

where αy is the one's complement of y wherein each bit of y is complemented. Substituting Equation (6) into the right hand side of Equation (4) provides:

$$16I = yA + \alpha yB + B. \quad (7)$$

In order to provide a circuit suitable for the implementation of Equation (7), the multiply operation for yA within Equation (7) is reviewed. If the four bits of weight term y are

y_3, y_2, y_1, and y_0, then yA becomes:

$$yA = y_0A + y_1(2A) + y_2(4A) + y_3(8A). \qquad (8)$$

Since each bit of weight term y can only be a zero or a one, each term in Equation (8) is either equal to zero or nA where n is a power of two.

Further to the process of providing a circuit for performing the multiplication portion of Equation (7), the remaining terms of Equation (7) are examined:

$$yA + \alpha yB = (y_0A + \alpha y_0B) + (y_1(2A) + \alpha y_1(2B)) + \ldots \qquad (9)$$

The operation performed in Equation (9) is similar to the operation performed in Equation (7) except that each term in Equation (9) is equal to either nA or nB depending on whether the corresponding bit of y is a one or a zero.

For example, if interpolation weight term y has the value five, the binary expression for y is 0101_2, and Equation (7) becomes:

$$16I = A + 2B + 4A + 8B + B = 5A + 11B, \text{ or}$$
$$I = (5/16)A + (11/16)B.$$

An implementation of an interpolator based on Equation (7) is generally more hardware efficient than prior art implementations based on either Equation (1) or Equation (2). The implementation based on Equation (7) needs one less subtraction

compared with the implementation of Equation (2) because the implementation of Equation (7) does not require the generation of the (A-B) term required by the implementation of Equation (2). The reason for the increase in efficiency provided by Equation (7) is that adders forming a multiplier in an implementation based on Equation (2) are not fully utilized when one or more bits of weight term y are equal to zero. In the case where one or more bits of y are equal to zero, the row of adders that corresponds to a zero bit of y simply passes the partial product on to the next stage. In the implementation of the present invention based on Equation (7), each stage of adders adds either an A term or a B term of the partial product. It will be understood by those skilled in the art that Equation (7) can be implemented by hardware or software.

Referring now to Fig. 4, there is shown one-dimensional interpolator cell 50. One-dimensional interpolator cell 50 or one-dimensional interpolator 50 is suitable for implementing the yA + AyB portion of the interpolation of the present invention as set forth in Equation (7). In one-dimensional interpolator cell 50, bit y_i of the interpolation weight term y is received by way of fractional weight line 18. Bit j of each of the two values to be interpolated, values A and B, are received by way of value input line 16a and value input line 16b respectively. Thus value bit A_j is received by way of value input line 16a and value bit B_j is received by way of value input line 16b. The partial prod-

uct bit P_i is received by interpolator cell 50 on incoming partial product line 12. The carry-in value is received by interpolator cell 50 by way of carry-in line 14 as previously described for prior art parallel multiplier cell 10. The outgoing partial product P_{i+1} is transmitted from interpolator cell 50 by way of partial product output line 24 and the carry-out from summer 20 is transmitted by way of carry-out line 26, also as previously described with respect to prior art parallel multiplier cell 10.

Bit j of each of the input values A and B to be interpolated is applied to an individual input line of interpolator multiplexer 52 within one-dimensional interpolator cell 50. Interpolator multiplexer 52 applies either value A_j, received by way of value input line 16a, or value bit B_j, received by way of value input line 16b, to summer 20 by way of multiplexer output signal line 53. The determination of which value input bit A_j, B_j is applied to summer 20 by interpolator multiplexer 52 is made in accordance with the value of interpolation weight bit y_i of interpolation weight term y. The value of interpolation weight bit y_i is applied to interpolator multiplexer 52 by way of interpolation weight bit line 18.

It will be understood by those skilled in the art, that one-dimensional interpolator cell 50 may be formed by modification of parallel multiplier cell 10. To perform this modification, AND gate 22 is removed and replaced with interpolator multiplexer 52, wherein the output of interpolator multiplexer 52

is applied to summer 20 in the same manner as the output of AND gate 22 within parallel multiplier cell 10. Two input lines 16a,b, rather than one, are provided in the modification of parallel multiplier cell 10 to form one-dimensional interpolator cell 50. Multiplexer 52, provided to form interpolation cell 50 by modification of multiplier cell 10, may be of a conventional type wherein two AND gates (not shown) receive the interpolation weight bit, one inverted and the other not inverted. Each of the AND gates also receives one of the input bits and the outputs of the AND gates are applied to an OR gate. It will be understood by those skilled in the art that such a multiplexer 52 behaves like AND gate 22 if one of the inputs is always zero.

It will also be understood by those skilled in the art, that other parallel multiplier cells (not shown), differing from parallel multiplier cell 10, may be modified in a similar manner to provide a one-dimensional interpolator cell (not shown) such as one-dimensional interpolator cell 50 provided that the required interpolator partial product terms are thereby produced. For example, the iterative cellular array multipliers taught by Kai Hwang in "Computer Arithmetic", ISBN 0-471-03496-7, may be modified to provide one-dimensional interpolator array 70 of the present invention. It will be understood that a multiplier cell having fewer gates than another will provide an interpolator cell having fewer gates.

Referring now to Fig. 5, there is shown one-dimensional interpolator array 70 of the present invention. One-dimensional interpolator array 70 is formed of a plurality of one-dimensional interpolator cells 50. To perform the one-dimensional interpolation of Equation (7), each bit of the interpolation weight term y is applied to a respective row of horizontally adjacent interpolator cells 50 within interpolator array 70. Each row of interpolator cells 50 of interpolator array 70 is provided with a respective interpolator weight bit line 18 for receiving an interpolator weight bit y_i of interpolator weight term y and applying interweight bit y_i to each horizontally adjacent interpolator cell 50 in the row.

Each bit of the input value A to be interpolated within one-dimensional interpolator array 70 is applied to a respective column of vertically adjacent interpolator cells 50 within one-dimensional interpolator array 70 by way of a respective value input line 16a. A bit of input value A applied to a column of interpolator cell 50 by a single value input line 16a is applied to a respective interpolator multiplexer 52 of each interpolator cell 50 in the column by value input line 16a. Each bit of the input value B to be interpolated within one-dimensional interpolator array 70 is applied to a respective column of vertically adjacent interpolator cells 50 within interpolator array 70 by way of a respective value line 16b. Thus each value line 16b applies its input value bit to the multiplexer 52 of each cell 50 in its respective column. Thus, the bit of input value B applied

to a column of interpolator cells 50 by a single value input line 16b is applied to a respective interpolator multiplexer 52 of each interpolator cell 50 in the column by value input line 16b in the same manner as value input line 16a.

The output partial product terms P0-P7 of the resulting interpolation value are provided by one-dimensional interpolator array 70 at partial product output lines 24. For four bit values A and B, the partial product terms P0-P7 of one-dimensional interpolator array 70 are set forth in Table 2.

			$[A_3B_3]y_0$	$[A_2B_2]y_0$	$[A_1B_1]y_0$	$[A_0B_0]y_0$	
		$[A_3B_3]y_1$	$[A_2B_2]y_1$	$[A_1B_1]y_1$	$[A_0B_0]y_1$		
	$[A_3B_3]y_2$	$[A_2B_2]y_2$	$[A_1B_1]y_2$	$[A_0B_0]y_2$			
$[A_3B_3]y_3$	$[A_2B_2]y_3$	$[A_1B_1]y_3$	$[A_0B_0]y_3$				
P7	P6	P5	P4	P3	P2	P1	P0

TABLE 2

The partial product terms within one-dimensional interpolator array 70 are transmitted between diagonally adjacent interpolator cells 50 by way of partial product lines 24 which apply the outgoing cell partial products to incoming cell partial product lines 12 of diagonally adjacent interpolator cells 50. Because the interpolation weight bit y_i of each row of interpolator array 70 controls the multiplexing of bits A_j and B_j by means of

multiplexer 52 within each row of interpolator array 50, the interpolation weight term y determines the partial product terms provided on the diagonal paths of one-dimensional interpolator array 70 formed by partial product lines 12, 24.

The first row of one-dimensional interpolator array 70 may be formed of parallel multiplier cells 10 rather than one-dimensional interpolator cells 50. Conventional parallel multiplier cells 10 having full adders 22 are sufficient for the first row of one-dimensional interpolator array 70 because no carry-in is required in the first row. The bits of input value B may be therefore be applied to carry-in lines 14 of the first row of interpolator array 70 to perform the +B operation set forth in Equation (7). In order to simplify the drawing of one-dimensional interpolator array 70, carry-in lines 14 and carry-out lines 26 are not shown. It will be understood by those skilled in the art that carry-in lines 14 and carry-out lines 26 of vertically adjacent interpolation cells 50 are coupled in each column of interpolation array 70.

Referring now to Fig. 6, there is shown a flow chart representation of prior art multiplication algorithm 100 for a performing a multiplication of two binary values. Prior art multiplication algorithm 100 is adapted, in particular, to multiply two four-bit binary values, x and A. However, it will be understood by those skilled in the art that multiplication algorithm 100 may be adapted to multiply binary values of varying lengths.

Execution of multiplication algorithm 100 proceeds from start 102 to block 104 where a counter is initialized to the value four where four is the length of binary values y and A. In block 106 a shift register is initialized to the value of y, where y is defined within multiplication algorithm 100 as one of the values to be multiplied. In block 108 register REG is initialized to the other input binary value, binary value A. The accumulator is then initialized to zero as shown in block 110.

At decision 112 a determination is made whether the least significant bit of the shift register containing the binary value y is zero. If it is determined at decision 112 that the least significant bit of the shift register is zero, register REG is shifted to the right by multiplying REG by two as shown in block 114. The counter is then decremented at block 120 and a determination is made at decision 122 whether the counter has been decremented to zero. If the counter has been decremented to zero as determined at decision 122, execution of multiplication algorithm 100 proceeds to terminal 126 and multiplication algorithm 100 is complete. Thus, if the least significant bit of the shift register containing the binary value y is zero, as determined by decision 112, register REG is shifted to the right and no additional value is accumulated in the accumulator.

If the least significant bit of the shift register containing the binary value y is one, as determined by decision 112, the contents of REG are accumulated by adding REG to the

accumulator as shown in block 116. Register REG is shifted to the right in block 118 by multiplying REG by two as previously described with respect to block 114. The counter is then decremented as shown in block 120 and tested in decision 124. If the counter is not zero, the binary value y, contained in shift register SHFT, is shifted one bit to the right and execution of algorithm 100 returns to block 112 where the least significant bit of the shifted binary value y is again tested.

Thus, whether the least significant bit of the shift register containing y is zero or one, register REG is shifted one bit to the right during each pass through multiplication algorithm 100. However, only if the least significant bit of the shift register is one does multiplication algorithm 100 require adding the contents register REG to the accumulator. In this manner a conventional multiplication of the binary values y and A is performed by multiplication algorithm 100.

Referring now to Fig. 7, there is shown a flow chart representation of interpolation algorithm 140 for performing the interpolation method of the present invention. Interpolation algorithm 140 of the present invention is adapted to perform an interpolation as set forth in Equation (7) by means of software operations corresponding to the logic of one-dimensional interpolator array 70.

Execution of interpolation algorithm 140 begins at start terminal 142 and proceeds to block 144 where a counter is initialized to four. In block 146, a shift register, SHFT, is initialized to the binary value y. The binary value y is the interpolation weight term. Equation (5) sets forth the relationship between interpolation weight term y and the fractional weight term x of the interpolation of value A and value B defined in Equation (1). In block 148, REG1 is initialized to the interpolation input value A and in block 150 REG2 is initialized to the interpolation input value B. In block 152, the accumulator is initialized to B in order to perform the +B operation of Equation (7) thereby translating the interpolation offset value to provide the interpolation result.

At decision 154 a determination is made whether the least significant bit of the fractional weight term in the shift register SHFT is zero. If the state of the least significant bit of the fractional weight term is zero, as determined in decision 154, then the contents of REG2, input value A, are accumulated by adding REG2 to the accumulator as shown in block 156. REG2 is then shifted to the right as shown in block 158. If the least significant bit of the fractional weight term is one, as determined by decision 154, REG1 rather than REG2 is accumulated by adding REG1 to the accumulator as shown in block 160. In this case, REG1 rather than REG2 is shifted to the right as shown in block 162.

Thus, the accumulation and shifting operations of blocks 156, 158 are performed on the contents of REG2, input value B, when the interpolation weight bit is zero, and the same accumulation and shifting operations, as set forth in blocks 160, 162, are performed upon the contents of REG1, input value A, when the interpolation weight bit is one. Therefore, decision 154 and blocks 156-162 perform the operations set forth for interpolator multiplexer 52 and summer 20 of one-dimensional interpolation cell 50. In one-dimensional interpolator cell 50 either input bit A_j on input line 16a, or input value B_j on input line 16b, is applied to summer 20 in accordance with the interpolation weight bit y_1 which controls interpolator multiplexer 52.

In block 164 of interpolation algorithm 140, the counter is decremented and a determination is made at decision 166 whether the counter is decremented to zero. If the counter is decremented to zero, as determined in decision 166, execution proceeds to terminal 170 and algorithm 140 is complete. If the counter is not decremented to zero, the interpolation weight term y is shifted to the right in block 168. After the shifting of block 168, execution of interpolation algorithm 140 proceeds to decision 154 where a determination is again made whether the least significant bit of the interpolation weight term is zero or one. It will be understood by those skilled in the art that the interpolation method of interpolation algorithm 140 may be performed with a left shift rather than a right shift at block 168

provided that the most significant bit of SHFT is tested in decision 154 rather than the least significant bit.

It will be understood that various changes in the details, materials and arrangement of the parts which have been described and illustrated in order to explain the nature of this invention may be made by those skilled in the art without departing from the principle and scope of the invention as expressed in the following claims.

CLAIMS:

1. An interpolator having means for receiving a first input value having first input value bits, a second input value having second input value bits, and an interpolation weight term having interpolation weight bits, to interpolate said first and second input values and provide an interpolation output value, said interpolator being formed of a plurality of interpolator cells having summing means for providing a summing means output value in accordance with summing means input values applied to said summing means, comprising:

selection means for receiving a first input value bit, a second input value bit, and an interpolation weight bit to provide a selected output signal representative of a selected one of said first and second input value bits in accordance with said interpolation weight bit; and,

means for applying said selected output signal to said summing means to provide a summing means output value representative of said interpolation output value.

2. The interpolator of claim 1, wherein said summing means further comprises:

means for receiving a first incoming interpolator cell partial product term; and,

means for providing a second outgoing interpolator cell partial product term in accordance with summing means input values including said first incoming interpolator cell partial product term.

3. The interpolator of claim 1, wherein said summing means further comprises:

means for receiving a first incoming interpolator cell carry-in term; and,

means for providing a second outgoing interpolator cell carry-out term in accordance with summing means input values including said first incoming interpolator cell carry-in term.

4. The interpolator of claim 1, wherein a plurality of said interpolator cells is arranged in a one-dimensional interpolator array having array rows and array columns, an array row having a plurality of horizontally adjacent interpolator cells and an array column having a plurality of vertically adjacent interpolator cells, and an interpolator cell having diagonally adjacent interpolator cells wherein a diagonally adjacent interpolator cell is disposed in a differing array row and in a differing array column.

5. The interpolator of claim 4 further comprising:

means for applying each of said first input value bits of said first input value to a respective plurality of

vertically adjacent interpolator cells forming a respective array column;

means for applying each of said second input value bits of said second input value to a respective plurality of vertically adjacent interpolator cells forming a respective array column; and,

means for applying each interpolation weight bit of said interpolation weight term to a respective plurality of horizontally adjacent interpolator cells forming a respective array row.

6. The one-dimensional interpolator of claim 5 further comprising:

means for receiving a first incoming interpolator cell partial product term; and,

means for providing a second outgoing interpolator cell partial product term in accordance with summing means input values including said first incoming interpolator cell partial product term.

7. The interpolator of claim 6, further comprising means for applying said second outgoing interpolator cell partial product term of a first interpolator cell to said means for receiving said first incoming interpolator cell partial product term of a second diagonally adjacent interpolator cell.

8. The one-dimensional interpolator of claim 5, wherein said summing means further comprises:

means for receiving a first incoming interpolator cell carry-in term; and,

means for providing a second outgoing interpolator cell carry-out term in accordance with summing means input values including said first incoming interpolator cell carry-in term.

9. The interpolator of claim 8, further comprising means for applying said second outgoing interpolator cell carry-out term of a first interpolator cell to said means for receiving a first incoming interpolator cell carry-in term of a second vertically adjacent interpolator cell.

10. An interpolation method for receiving a first input value having first input value bits, a second input value having second input value bits, and an interpolation weight term having interpolation weight bits, to interpolate said first and second input values and provide an interpolator output value in a interpolator formed of a plurality of interpolator cells, each cell having summing means for providing a summing means output value in accordance with summing means input values applied to said summing means, comprising the steps of:

(a) applying said first input valve bit and said second input value bit and said interpolation weight bit to selecting means;

(b) providing a selecting means output signal representative of a selected one of said first and second input value bits in accordance with said interpolation weight bit; and,

(c) applying said selecting means output signal to said summing means.

11. The interpolation method of claim 10, comprising the further steps of:

(d) applying a first incoming interpolator cell partial product term to said summing means; and,

(e) providing a second outgoing interpolator cell partial product term in accordance with summing means input values including said first incoming interpolator cell partial product term.

12. The interpolation method of claim 11, comprising the further steps of:

(f) applying a first incoming interpolator cell carry-in term to said summing means; and,

(g) providing a second outgoing interpolator cell carry-out term in accordance with summing means input values including said first incoming interpolator cell carry-in term.

13. The interpolation method of claim 10, comprising the further steps of arranging a plurality of said interpolator cells in an array having array rows and array columns, an array row having a plurality of horizontally adjacent interpolator cells and an array column having a plurality of vertically adjacent interpolator cells, and an interpolator cell having diagonally adjacent interpolator cells wherein a diagonally adjacent interpolator cell is disposed in a differing row and in a differing column.

14. The interpolation method of claim 13, comprising the further steps of:

(h) applying each of said first input value bits of said first input value to a respective plurality of vertically adjacent interpolator cells forming an array column;

(i) applying each of said second input value bits of said second input value to a respective plurality of vertically adjacent interpolator cells forming an array column; and,

(j) applying each interpolation weight bit of said interpolation weight term to a respective plurality of horizontally adjacent interpolator cells forming an array row.

15. The interpolation method of claim 14, comprising the further steps of:

(k) receiving a first incoming interpolator cell partial product term by receiving means of said interpolator cell;

(l) applying said first incoming interpolator cell partial product term to said summing means; and,

(m) providing a second outgoing interpolator cell partial product term in accordance with summing means input values including said first incoming interpolator cell partial product terms.

16. The interpolation method of claim 15, comprising the further step of applying said second outgoing interpolation cell partial product term of a first interpolator cell to said means for receiving said first incoming interpolator partial product term of a second diagonally adjacent interpolator cell.

17. The interpolation method of claim 14, comprising the further steps of:

(n) receiving a first incoming interpolator cell carry-in term by receiving means of said interpolator cell;

(o) applying said first incoming interpolator cell carry-in term to said summing means; and,

(p) providing a second outgoing interpolator cell carry-out term in accordance with summing means input values including said first incoming interpolator cell carry-in term.

18. The interpolation method of claim 17, comprising the further step of applying said second outgoing interpolator cell carry-out term of a first interpolator cell to said means for receiving said first incoming interpolator cell carry-in term of a second vertically adjacent interpolator cell.

19. An interpolator for receiving first and second input values and an interpolation weight value, each of said values having a current shifted value, said current shifted interpolation weight value having a current test bit, to interpolate said first and second input values and provide an interpolation output value, said interpolator having means for accumulating each of said values and means for shifting each of said values, comprising;

means for determining the state of said current test bit of said current shifted interpolation weight value to provide an interpolation weight test signal in response to said determination;

accumulating means for accumulating said current shifted values of said first and second input values in accordance with said interpolation weight test signal;

input value shifting means for shifting said first and second current shifted input values in accordance with said interpolation weight test signal to provide new first and second current shifted input values; and,

weight value shifting means for shifting said interpolation weight value to provide a new current shifted interpolation weight value having a new current test bit.

20. The interpolator of claim 19, wherein said accumulating means comprises means for accumulating only a selected one of said first and second current shifted input values in accordance with said interpolation weight test signal.

21. The interpolator of claim 20, wherein said input value shifting means comprises means for shifting only a selected one of said first and second current shifted input values in accordance with said interpolation weight test signal.

22. The interpolator of claim 21, wherein said current test bit comprises the least significant bit of said current shifted interpolation weight value.

23. The interpolator of claim 22, wherein input value shifting means comprises shifting only a selected one of said first and second current shifted input values to the right.

24. The interpolator of claim 21, wherein said current test bit comprises the most significant bit of said current shifted interpolation weight value and said weight value shifting means comprises means for shifting said selected one of said current shifted input values to the left.

25. The interpolator of claim 19, further comprising means for accumulating one of said first and second input values.

26. The interpolation of claim 19, wherever said means for further accumulating comprises means for initializing said means for accumulating to contain one of said first and second values.

27. An interpolation method for receiving first and second input values and an interpolation weight value, each of said values having a current shifted value, said current shifted interpolation weight value having a current test bit, to interpolate said first and second input values and provide an interpolation output value in a system having means for accumulating and shifting each of said values, comprising the steps of:

(a) determining the state of said current test bit of said current shifted interpolation weight value and providing an interpolation weight test signal in response to said determination;

(b) activating said accumulating means to accumulate said current shifted values of said first and second input values in accordance with said interpolation weight test signal;

(c) activating said shifting means to shift said first and second current shifted input values in accordance with said interpolation weight test signal for providing new first and second current shifted input values;

(d) activating said shifting means to shift said interpolation weight value for providing a new current shifted interpolation weight value having a new current test bit; and,

(e) repeating steps (a)-(d) until said interpolation is complete.

28. The interpolation method of claim 27, wherein step (b) comprises accumulating only a selected one of said first and second current shifted input values in accordance with said interpolation weight test signal.

29. The interpolation method of claim 28, wherein step (c) comprises shifting only a selected one of said first and second current shifted input values in accordance with said interpolation weight test signal.

30. The interpolation method of claim 29, wherein said current test bit comprises the least significant bit of said current shifted interpolation weight value.

31. The interpolation method of claim 30, wherein step (c) comprises shifting a selected one of said first and second current shifted input values to the right.

32. The interpolation method of claim 29, wherein said current test bit comprises the most significant bit of said current shifted interpolation weight value and step (d) comprises shifting a selected one of said current shifted input values to the left.

33. The interpolation method of claim 27, comprising the further step of further accumulating one of said first and second input values.

34. The interpolation method of claim 33, wherein the step of further accumulating comprises initializing said means for accumulating to contain one of said first and second input values.

35. A method for forming an interpolator array by modification of a parallel multiplier cell of parallel multiplier array, said parallel multiplier array being formed of a plurality of multiplier cells, each multiplier cell having logical AND means for receiving first and second input bits of respective first and second input values and summing means for receiving the output of said logical AND means, said formed interpolator array being adapted to receive said first and second input values and an

interpolation weight term having interpolation bits to provide an interpolator output, comprising the steps of:

(a) deactivating said logical AND means;

(b) providing selecting means having first and second input means for receiving said first and second input bits and selection output means for providing a selecting means output signal;

(c) controlling said selecting means to provide a selected one of said first and second input bits as said selecting means output signal in accordance with one of said interpolation weight bits; and,

(d) applying said selecting means output signal to said summing means to provide said modified multiplier cell for forming an interpolator cell.

36. The method for forming an interpolator array of claim 35, wherein said selecting means comprises multiplexer means.

37. The method for forming an interpolator array of claim 35, comprising the further steps of;

(e) applying a first incoming interpolator cell partial product term to said summing means; and,

(f) providing a second outgoing interpolator cell partial product term in accordance with summing means input values including said first incoming interpolator cell partial product term.

38. The method for forming an interpolator array of claim 37, wherein a partial product term of said parallel multiplier cell and said partial product term of said modified multiplier term differ from each other according to the relationship $[A_i B_i] Y_j - A_i B_j$ where A and B are said first and second input values and y is said interpolation weight term.

39. The method for forming an interpolation array of claim 35, comprising the further steps of:

(g) applying a first incoming interpolator cell carry-in term to said summing means; and,

(h) providing a second outgoing interpolator cell carry-out term in accordance with summing means input values including said first incoming interpolator cell carry-in term.

40. The method for forming an interpolator array of claim 35, comprising the further step of arranging a plurality of said modified multiplier cells in an array having array rows and array columns, an array row having a plurality of horizontally adjacent modified multiplier cells and an array column having a plurality of vertically adjacent modified multiplier cells, and

modified multiplier cell having diagonally adjacent modified multiplier cells, wherein a diagonally adjacent modified multiplier cell is disposed in a differing row and in a differing column.

41. The method for forming an interpolator array of claim 40, comprising the further steps of:

(i) applying each of said first input value bits of said first input value to a respective plurality of vertically adjacent modified multiplier cells forming an array column;

(j) applying each of said second input value bits of said second input value to a respective plurality of vertically adjacent interpolator cells forming an array column; and,

(k) applying each interpolation weight bit of said interpolation weight term to a respective plurality of horizontally adjacent modified multiplier cells forming an array row.

42. The method for forming an interpolator array of claim 35, wherein steps (a) and (b) comprise the steps of:

(l) providing first and second logical AND means;

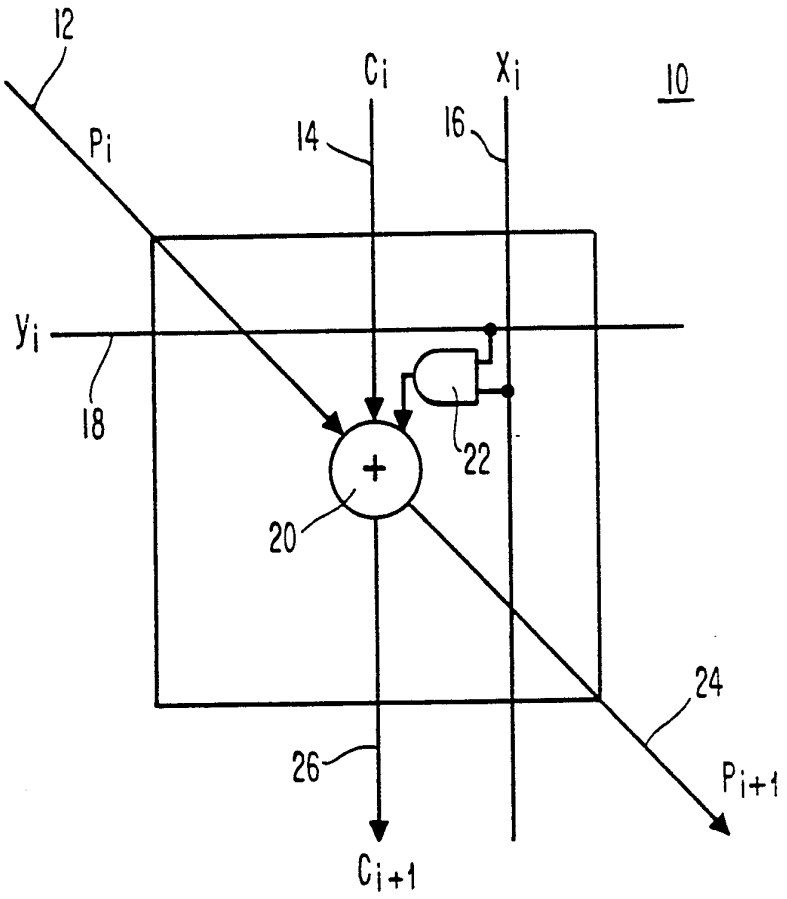
(m) applying one of said interpolation weight bits, inverted to an input of one of said first and second logical AND means and uninverted to the other of said first and second logical AND means.

43. The method for forming an interpolator array of claim 42, comprising the further steps of:

(n) applying one of said first and second input bits to each of said first and second logical AND means; and,

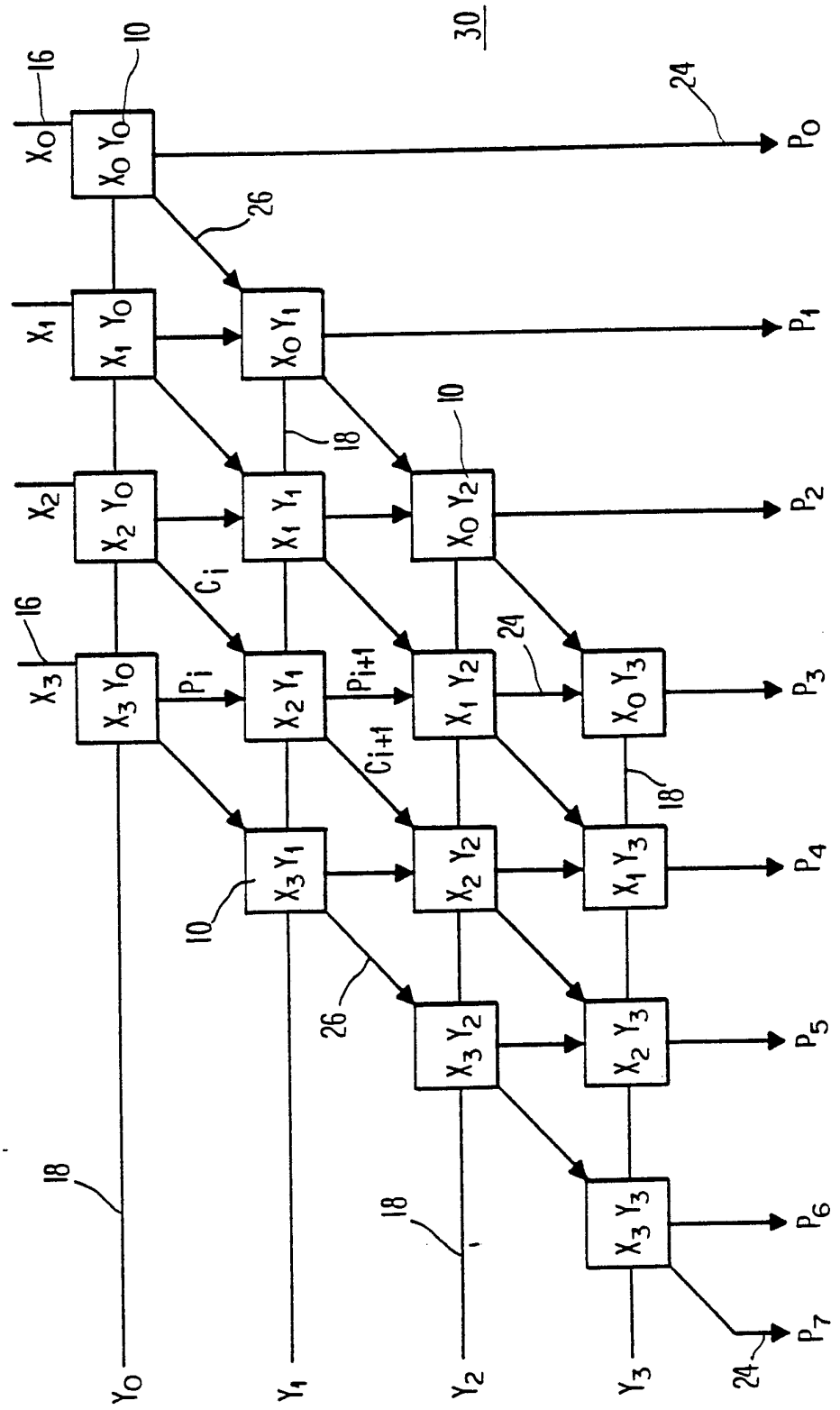
(o) applying the output of each of said first and second logical AND means to the input of logical OR means.

44. The method for forming an interpolator array of claim 43, comprising the step of applying the output of said logical OR means to said summing means.



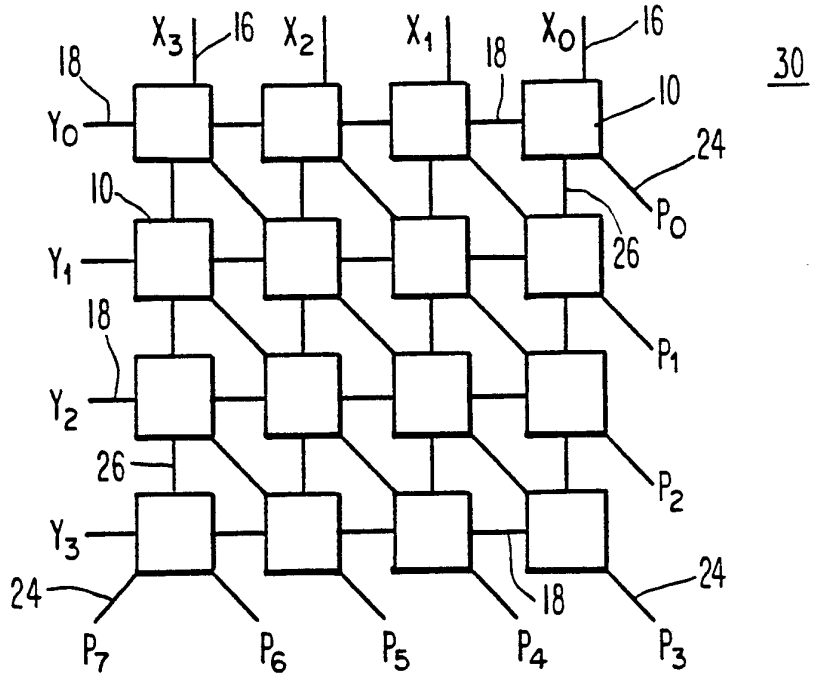
PRIOR ART

Fig. 1



PRIOR ART

Fig. 2



PRIOR ART

Fig. 3

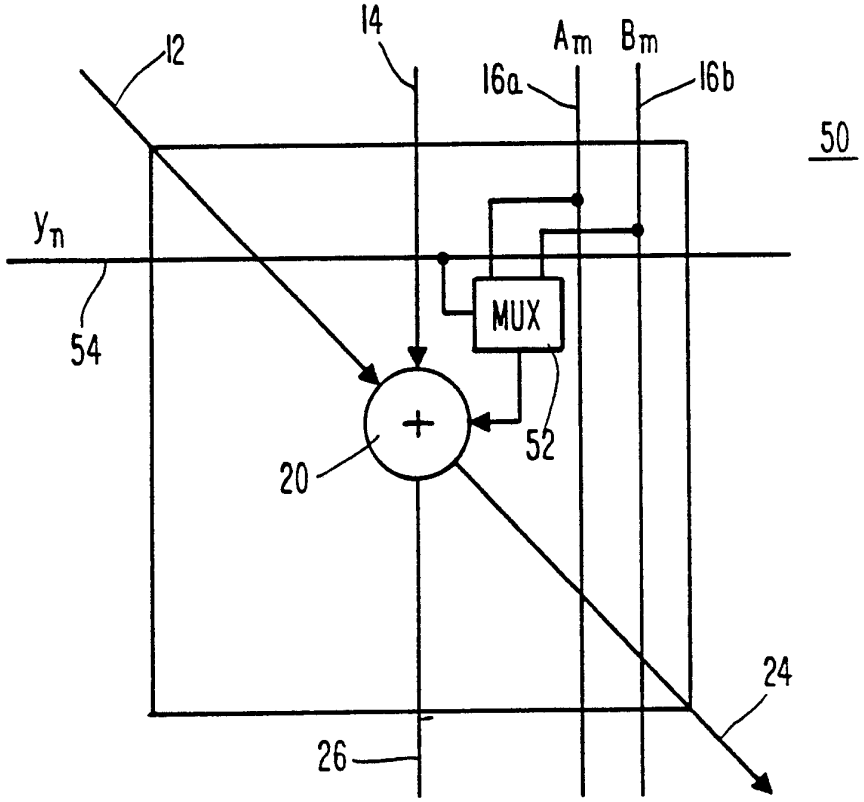


Fig. 4

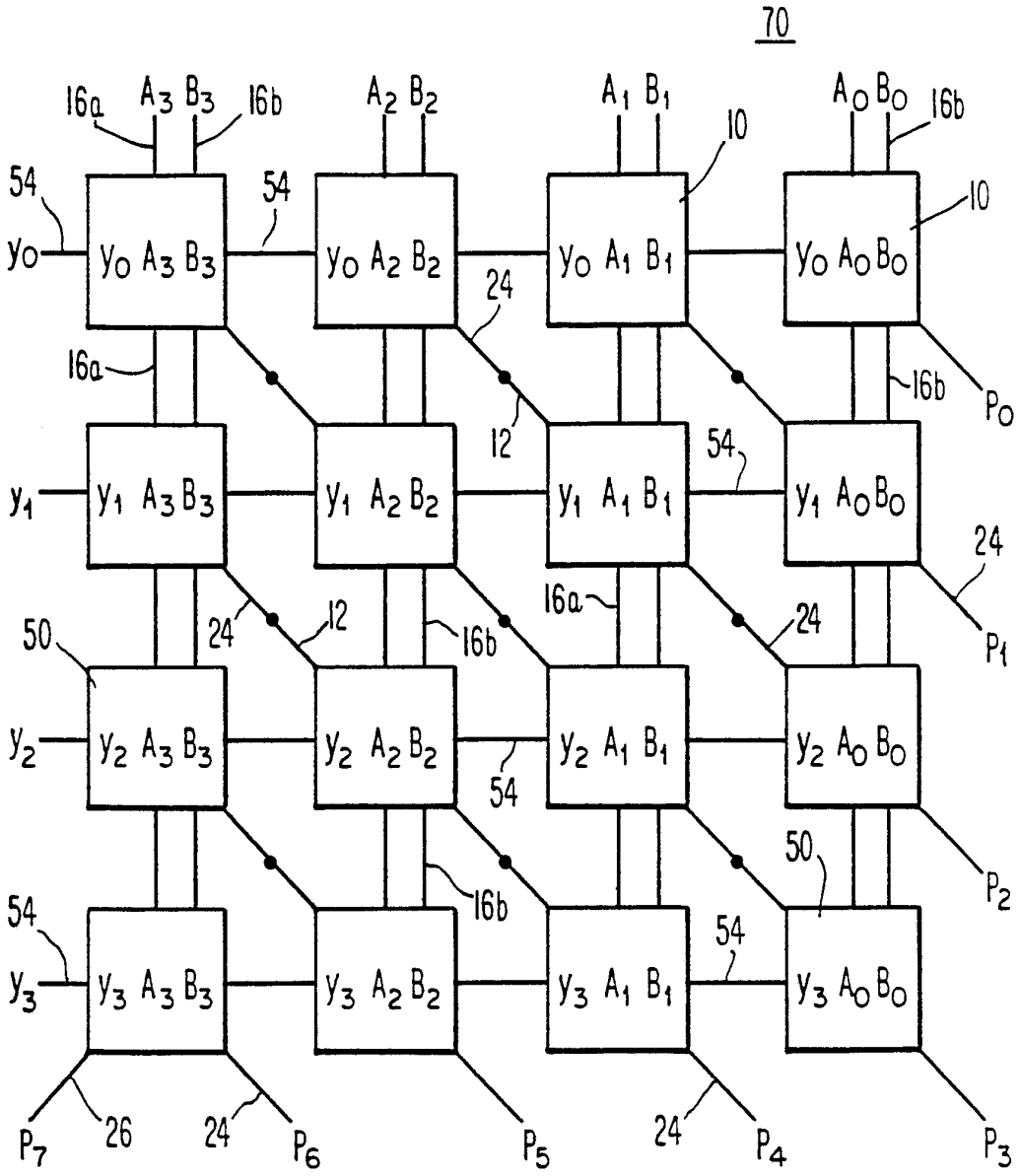
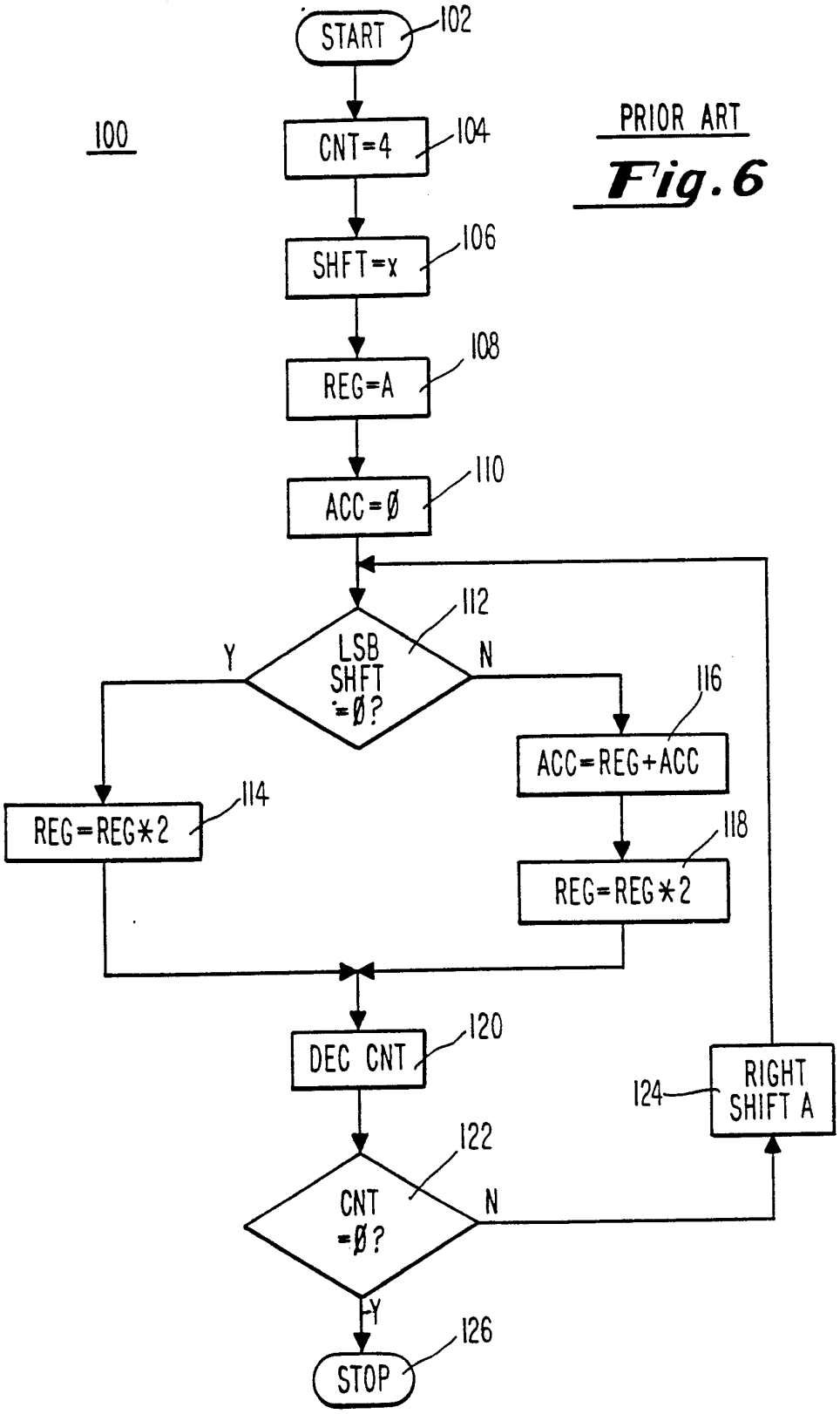

70
16a
A3 B3
16b
A2 B2
A1 B1
A0 B0
16b
10
10
54
54
yo
yo A3 B3
yo A2 B2
yo A1 B1
yo Ao Bo
24
16a
16b
Po
12
y1
y1 A3 B3
y1 A2 B2
y1 A1 B1
54
y1 Ao Bo
24
16a
24
24
12
16b
P1
50
y2
y2 A3 B3
y2 A2 B2
y2 A1 B1
y2 Ao Bo
54
50
16b
P2
54
54
y3
y3 A3 B3
y3 A2 B2
y3 A1 B1
y3 Ao Bo
P7
26
24
P6
P5
24
P4
P3

Fig. 5



PRIOR ART

Fig. 6

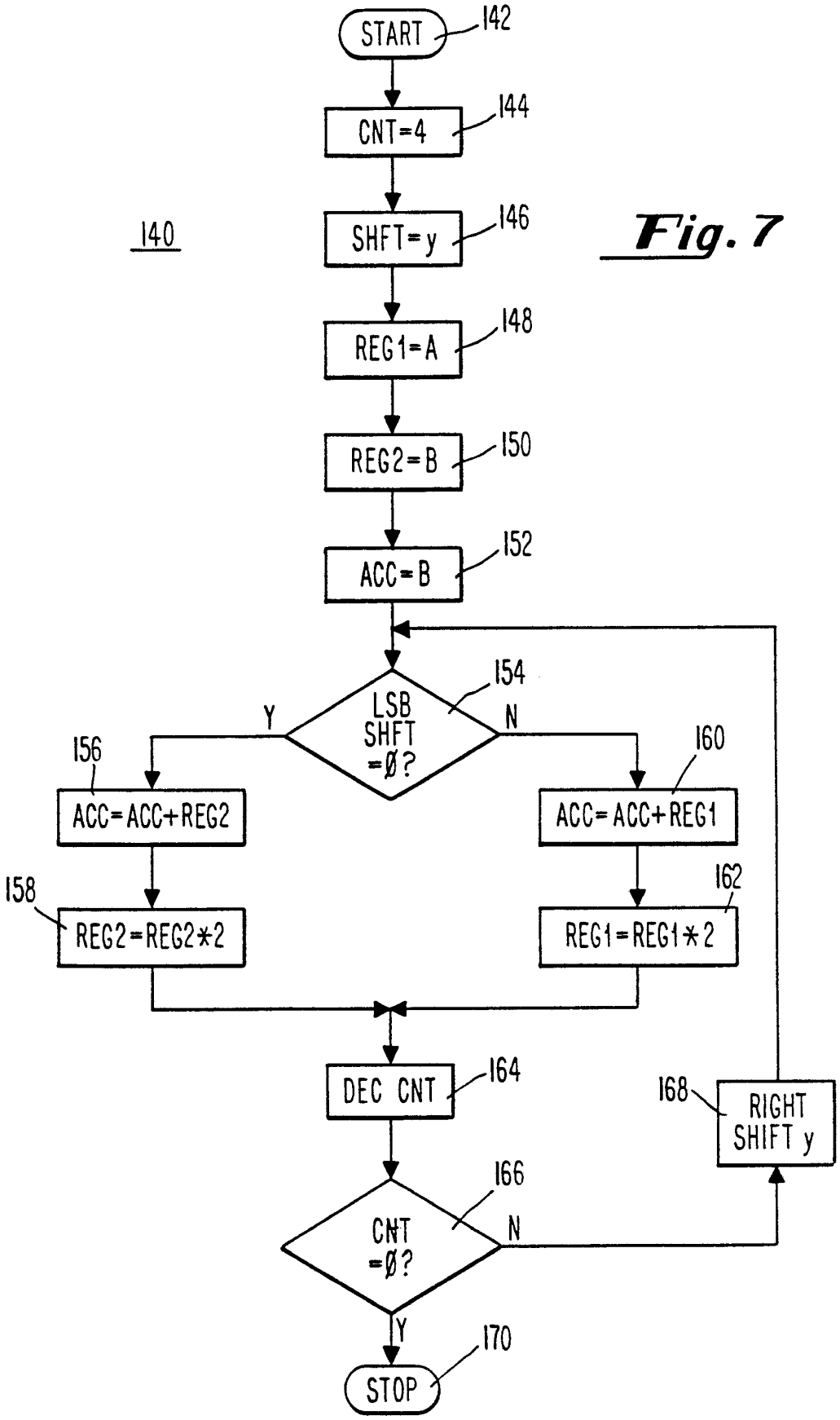

START
142
CNT=4
144
140
SHFT=y
146
REG1=A
148
REG2=B
150
ACC=B
152
LSB SHFT =Ø?
154
Y
N
156
ACC=ACC+REG2
160
ACC=ACC+REG1
158
REG2=REG2*2
162
REG1=REG1*2
DEC CNT
164
168
RIGHT SHIFT y
CNT =Ø?
166
N
Y
170
STOP

Fig. 7

INTERNATIONAL SEARCH REPORT

International Application No PCT/US 92/00902

I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all)[6]

According to International Patent Classification (IPC) or to both National Classification and IPC

Int.Cl. 5 G06F15/353

II. FIELDS SEARCHED

Minimum Documentation Searched[7]

Classification System	Classification Symbols
Int.Cl. 5	G06F

Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched[8]

III. DOCUMENTS CONSIDERED TO BE RELEVANT[9]

Category °	Citation of Document,[11] with indication, where appropriate, of the relevant passages[12]	Relevant to Claim No.[13]
P,X P,Y	WO,A,9 118 355 (ANALOG DEVICES, INC., US) 28 November 1991 see page 2, line 10 - line 25	1-3, 10-12 4-8, 13-18, 35-44
P,A	see page 4, line 1 - page 5, line 5 see page 6, line 12 - page 8, line 11 see page 8, line 17 - page 10, line 5 see page 13, line 20 - line 24	19-34
Y	IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, IEEE COMPUTER SOCIETY PRESS, NEW YORK, US vol. 36, no. 11, November 1989, pages 1488 - 1493; RAJAT ROY ET AL: 'AN EFFICIENT TWO'S COMPLEMENT SYSTOLIC MULTIPLIER FOR REAL-TIME DIGITAL SIGNAL PROCESSING' see page 1488, right column, line 18 - line 34	4-8, 13-18, 35-44

° Special categories of cited documents :[10]

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

IV. CERTIFICATION

Date of the Actual Completion of the International Search	Date of Mailing of this International Search Report
25 JUNE 1992	13. 07. 92
International Searching Authority	**Signature of Authorized Officer**
EUROPEAN PATENT OFFICE	BARBA M. M. Barba

Form PCT/ISA/210 (second sheet) (January 1985)

ANNEX TO THE INTERNATIONAL SEARCH REPORT ON INTERNATIONAL PATENT APPLICATION NO. US 9200902 SA 58182

This annex lists the patent family members relating to the patent documents cited in the above-mentioned international search report. The members are as contained in the European Patent Office EDP file on The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information. 25/06/92

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO-A-9118355	28-11-91	None	