(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
27 May 2004 (27.05.2004)

PCT

(10) International Publication Number
WO 2004/044731 A2

(51) International Patent Classification^7: G06F 7/00

(21) International Application Number: PCT/EP2003/050754

(22) International Filing Date: 24 October 2003 (24.10.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
02102559.8 12 November 2002 (12.11.2002) EP

(71) Applicant *(for all designated States except US)*: **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, Armonk, NY 10504 (US).

(71) Applicant *(for LU only)*: **IBM DEUTSCHLAND GMBH** [DE/DE]; Pascalstrasse 100, 70569 Stuttgart (DE).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **BÜTTNER, Stefan** [DE/DE]; Sommerhofenstrasse 166, 71067 Sindelfingen (DE). **LEENSTRA, Jens** [DE/DE]; Brühlstrasse 12, 71149 Bondorf (DE). **MAEDING, Nicolas** [DE/DE]; Römersteinweg 8, 71088 Holzgerlingen (DE). **PILLE, Jürgen** [DE/DE]; Adlerstrasse 56, 70199 Stuttgart (DE).

(74) Agent: **KAUFFMANN, Wolfgang**; Postal Code, 70548 Stuttgart (DE).

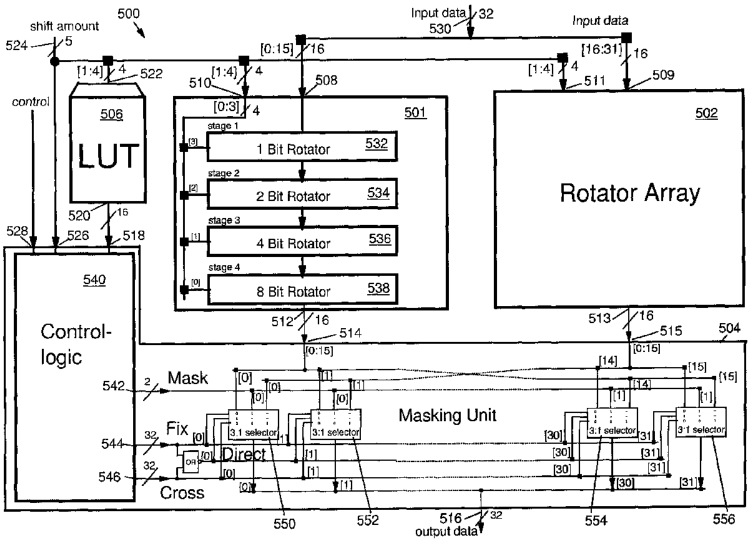
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

[Continued on next page]

(54) Title: DEVICE AND METHOD FOR PERFORMING SHIFT/ROTATE OPERATIONS



(57) **Abstract:** A method and device is provided for performing rotate operations on operands having a size of 2N bits, alternatively, for performing rotate operations on two operands each having a size of N bits to the left, whereby N is an integer. The device includes a control unit being adapted for exchanging M least significant bits of the output of a first rotate circuit with M least significant bits of the output of a second rotate circuit, when M=N is true, whenever an input having the width of 2N is to be rotated by M bits. For rotator arrays rotating N bit wide data to the right, it functions correspondingly.

DESCRIPTION

Device And Method For Performing Shift/Rotate Operations

Background of the Invention

1. Field of the Invention

The present invention generally relates to a device and a method for processing data by operating upon the order or content of the data handled. Particularly, the present invention relates to a device and method for performing shift/rotate operations on operands of any of a plurality of sizes between 2 and 2N bits and, alternatively, for performing shift/rotate operations on two operands each of any of a plurality of sizes between 1 and N bits, whereby N is an integer number.

2. Description of the Related Art

In advance microprocessor machines, means for shifting/rotating an input word and having the capability also to shift/rotate the input word in one machine cycle, or less, can be used to perform very powerful functions. Separately, in order to make maximum use of very large scale integration (VLSI) technology and its capability, the trend in processor design is towards more powerful operation codes and toward wider and wider operation code word widths.

A state-of-the-art rotator/shifter 100 is shown in Fig. 1. The rotator/shifter 100 comprises three blocks, namely, a rotator array 102, a masking unit 104 and a look-up table (LUT) 106. The rotator array is provided with a 32 bit wide input port 108 and a 5 bit wide control port 110 for specifying the shift/rotate amount. Furthermore, the rotator array has got a 32 bit wide output port 112 for transferring the shifted/rotated data to the masking unit 104. The masking unit

104 takes the data via a respective input port 114 and outputs the masked data via an output port 116. Additionally, the masking unit 104 is equipped with a 32 bit wide control port 118 for specifying the mask information. The mask information origin from the look-up-table 106, which is provided with a respective output port 120. Furthermore, the LUT 106 has got a 5 bit wide input port 122 for receiving the shift/rotate amount and a control port 124 for determining the functionality of the rotator/shifter 100 like for example if a rotate or shift needs to be performed.

Such a state of the art rotator/shifter may be configured to perform rotates and shifts to the left and right based on a shift amount with a 2-complement binary format. Different implementations of the rotator array are possible.

Fig. 2 shows an example of a 32-bit rotator array 200 based on logarithmic rotators with increasing rotate amount according to the prior art. The rotator array 200 is provided with a 32 bit wide input port 202 and a 5 bit wide control port 204 for specifying the shift amount. Furthermore, the rotator array 200 has got a 32 bit wide output port 206 for outputting the shifted/rotated data. Internally, the rotator array 200 comprises a 1-bit rotator 210, a 2-bit rotator 212, a 4-bit rotator 214, an 8-bit rotator 216 and a 16-bit rotator 218, each having a 32 bit wide input port, a 32 bit wide output port and a one bit wide control port, whereby the output port of one rotator is connected to the input port of the following rotator. When activated by the respective signal received via the 5 bit wide control port 204, the rotator rotates the input bit pattern, i.e., the 1-bit rotator 210 rotates by 1 bit position, the 2-bit rotator 212 rotates by 2 bit positions and so on. In case a rotator is inactive, the bit pattern present at its input port is transferred through to the output port without applying any changes. Therefore, if the 1-bit rotator

210 and the 4-bit rotator 214 are the only active rotators at a given time, a bit pattern at the input port 202 gets rotated by 5 bit positions before reaching the output port 206.

A detailed description of such a rotator array may be found in US 4,583,197 by Barbara A. Chappell et al., assigned to IBM, filed Jun. 30, 1983, issued Apr. 15, 1986, "Multi-Stage Pass Transistor Shifter/Rotator"

Beside the aforementioned implementation, it is possible to change the sequence order of the logarithmic rotators, the rotate direction (left or right) as well as using a gate array based implementation or any other rotator array.

The LUT and masking unit are needed to perform shifts. Based on the output of the LUT, the masking unit forces the outputs of the rotator array to zero or one, if algebraic right shifts are performed and the MSB is one. See for example US-Patent 4,396,994 by Sung M. Kang, assigned to Bell Telephone Laboratories, filed Dec. 31, 1980, issued Aug. 2, 1983, "Data Shifting And Rotating Apparatus."

US 5,961,575 by Mark W. Hervin, assigned to National Semiconductor Corporation, filed Feb. 26, 1996, issued Oct. 5, 1999, "Microprocessor Having Combined Shift and Rotate Circuit" describes a different approach, suitable for single word, halfword or byte shifter/rotator. The control logic is modified and additional control inputs have been introduced for the rotator array. This enables that the rotator array needed for the word rotate/shifts generates the data needed for half word and byte rotate/shift as well. Note however that in this way only 1 half word or byte instruction can be processed at a time. The upper bit for the half word and byte rotate/shift result need to be ignored.

In modern applications, such as graphic applications, e.g., video game engines, a rotator is not only required to perform the state of the art 32-bit rotate/shifts, but also a single instruction multiple data (SIMD) type of half-word rotate/shifts. In other words, e.g., two independent 16-bit rotate/shift instructions have to be performed in parallel for the bit range 0 to 15 as well as for the bit range 16 to 31.

Object of the Invention

Starting from this, the object of the present invention is to provide a device and method for performing shift/rotate operations on operands of any of a plurality of sizes between 2 and 2N bits and, alternatively, for performing shift/rotate operations on two operands.

Brief Summary of the Invention

The foregoing object is achieved by a method and a system as laid out in the independent claims. Further advantageous embodiments of the present invention are described in the sub claims and are taught in the following description.

According to the present invention a method and a device is provided for performing rotate operations on operands having a size of 2N bits and, alternatively, for performing rotate operations on two operands each having a size of N bits, whereby N is an integer number. The device according to the present invention includes an input port for receiving 2N bits numbered 1 to 2N, a first rotate circuit having an input for receiving bits 1 to N of said input port, a second rotate circuit having an input for receiving bits N+1 to 2N of said input port, whereby said first and said second rotate circuit are configured to operate each with N bits at a time. Furthermore, it includes a control unit being adapted for exchanging the M least significant bits of the output of said first rotate circuit with the M least significant bits of the output of said second rotate circuit, when M<N is true, and for exchanging the 2N-M most significant bits of the output of said first rotate circuit with the 2N-M most significant bits of the output of said second rotate circuit, when M>N is true, whenever an input having the width of 2N is to be rotated by M bits. Furthermore the control unit exchanges all N bits of said first and second rotator circuit output for the case that M=N. The first and second rotate circuits each are formed by rotator arrays capable of rotating N bit wide data to the left.

For rotator arrays that are capable of rotating N bit wide data to the right, the control unit is adapted for exchanging the M most significant bits of the output of said first rotate circuit with the M most significant bits of the output of said second rotate right circuit, when M<N is true, and for exchanging the 2N-M least significant bits of the output of said first rotate circuit with the 2N-M least significant bits of the output of said second rotate circuit, when M>N is true, whenever an input having the width of 2N is to be rotated by M bits.

In a preferred embodiment the control unit of the device according to the present invention is adapted to mask individual bit positions in order to perform a shift operation instead of a rotate operation.

Hence, the present invention presents a rotator/masking device, which may be used for SIMD (Single Instruction Multiple Data) high frequency processor that has to support half-word (16 bit) and word (32 bit) rotates/shifts. The introduction of cross signals and the integration of the last rotate logic stage into the masking unit advantageously allows the reduction of the logic delay as well as the wire delay of the rotator/shifter implementation.

With four rotator devices according to the present invention, each processing 32-bit input of a total of 128 bit input; it is possible to perform 4 word SIMD rotate operations or 8 half word SIMD rotate operations in parallel. The ability to do such operation is important in the aforementioned field of graphic applications.

Brief Description of the Several Views of the Drawings

The above, as well as additional objectives, features and advantages of the present invention, will be apparent in the following detailed written description.

The novel features of the invention are set forth in the appended claims. The invention itself, however, as well as a preferred mode of use, further objectives, and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

Fig. 1 shows a state-of-the-art rotator/shifter according to the prior art;

Fig. 2 shows a 32 bit Rotator Array based on logarithmic rotators with increasing rotate amount according to the prior art;

Fig. 3 shows a diagram schematically illustrating the operation of the device for performing rotate operations according to the present invention on the basis of a first example;

Fig. 4 shows a diagram schematically illustrating the operation of the device for performing rotate operations according to the present invention on the basis of a second example;

Fig. 5 shows a block diagram of a device for performing shift/rotate operations on operands of any of a plurality of sizes between 2 and 2N bits and, alternatively, for performing shift/rotate operations on two operands each of any of a plurality of sizes between 1 and N bits, with N equal 16, according to the present invention.

Detailed Description of the Invention

With reference to Fig. 3, there is depicted a diagram schematically illustrating the operation of the device for performing rotate operations according to the present invention on the basis of a first example. The device according to the present invention is configured to perform shift/rotate operations by M bit positions to the left on operands of any of a plurality of sizes between 2 and 2N bits and, alternatively, for performing shift/rotate operations by M bit positions to the left on two operands each of any of a plurality of sizes between 1 and N bits, whereby N and M are integer numbers.

It is acknowledged that in the case of performing rotate operations by M bit positions on two operands each of any of a plurality of sizes between 1 and N bits, two rotate circuits being capable of shifting/rotating N-bit operands operate independently from each other, like two separate N-bit rotators. This includes the extension that each rotate array receives an independent shift amount M.

However, the arrangement according to the present invention is advantageously being capable of performing rotate operations by M bit positions on operands of any of a plurality of sizes between 2 and 2N bits using the same two separate N-bit rotator and some additional circuitry, which will be described below in greater detail. In the following examples such rotate operations will be shown.

In the first example to be presented, M, denoting the number of bit positions to be shifted, is equal 2 and N is equal 16, whereby 2N equal 32 is the overall width of the data input. The input port 302 of a first 16-bit rotator (not shown) holds the bit positions 0 to 15 as illustrated by the figures "0...15" in input port 302. Correspondingly, the input port 304 of a second 16-bit rotator (not shown) holds the bit positions 16 to 31 as illustrated by the figures "16...31" in input port 304.

The output port 306 of the first 16-bit rotator holds the original bit pattern rotated by 2 positions. The most significant bit (MSB) in the output port 306 is now the original bit position 2, whereas the least significant bit (LSB) in the output port 306 is the original bit position 1, as illustrated by the figures "2...15,0,1" in output port 306. Correspondingly, the output port 308 of the second 16-bit rotator holds the original bit pattern rotated by 2 positions. The MSB in the output port 308 is now the original bit position 18, whereas the LSB in the output port 308 is the original bit position 17, as illustrated by the figures "18...31,16,17" in output port 308.

In the masking unit according to the present invention (shown in Fig. 5 and 6), the bit positions of the output ports 306 and 308 of the first and the second 16-bit rotator are manipulated in a way to form a result as of a 32-bit rotator (arrows 310, 312). In the present example, M is less than N. Therefore, the M least significant bits, i.e., the 2 least significant bits of the first rotator's output port 306 are exchanged with the 2 least significant bits of the second rotator's output 308. Now, the upper 16-bit of the masking unit's output port 314, corresponding to the first rotator, comprise the original bit positions "2...15,16,17", whereas the lower 16-bit of the masking unit's output port 316, corresponding to the second rotator, comprise the original bit positions "18...31,0,1" as illustrated in Fig. 3.

With reference to Fig. 4, there is depicted a diagram schematically illustrating the operation of the device for performing rotate operations according to the present invention on the basis of a second example. The device according to the present invention is configured to perform shift/rotate operations by M bit positions on operands of any of a plurality of sizes between 2 and 2N bits and, alternatively, for performing shift/rotate operations by M bit positions on two operands each of any of a plurality of sizes between 1 and N bits, whereby N and M are integer numbers.

As already mentioned with reference to Fig. 3, it should be noted that in the case of performing rotate operations by M bit positions on two operands each of any of a plurality of sizes between 1 and N bits, two rotate circuits being capable of shifting N-bit operands operate independently from each other, like two separate N-bit rotators.

However, the arrangement according to the present invention is advantageously being capable of performing rotate operations by M bit positions on operands of any of a plurality of sizes between 2 and 2N bits using the same two separate N-bit rotator and some additional circuitry, which will be described below in greater detail. In the following examples such rotate operations will be shown.

In the second example to be presented, M, denoting the number of bit positions to be shifted, is equal 18 and N is equal 16, whereby 2N equal 32 is the overall width of the data input. The input port 402 of a first 16-bit rotator (not shown) holds the bit positions 0 to 15 as illustrated by the figures "0...15" in input port 402. Correspondingly, the input port 404 of a second 16-bit rotator (not shown) holds the bit positions 16 to 31 as illustrated by the figures "16...31" in input port 404.

The output port 406 of the first 16-bit rotator holds the original bit pattern rotated by 18 positions. The most significant bit (MSB) in the output port 406 is now the original bit position 2, whereas the least significant bit (LSB) in the output port 406 is the original bit position 1, as illustrated by the figures "2...15,0,1" in output port 406. Correspondingly, the output port 408 of the second 16-bit rotator holds the original bit pattern rotated by 18 positions. The MSB in the output port 408 is now the original bit position 18, whereas the LSB in the output port 408 is the original bit position 17, as illustrated by the figures "18...31,16,17" in output port 408.

It can be seen that the result of the 2-bit-rotate of the first example (cf. Fig. 3) and the 18-bit-rotate of the present (second) example are identical up to this point. However, in the masking unit according to the present invention (shown in Fig. 5 and 6), the bit positions of the output ports 406 and 408 of the first and the second 16-bit rotator are manipulated in a way to form a result as of a 32-bit rotator (arrows 410, 412), in which it is different whether to rotate by 2 bit positions or by 18 bit positions.

In the present example, M is greater than or equal N. Therefore, the 2N-M, i.e., 32-18=14 most significant bits of the first rotator's output port 406 are exchanged with the 14 most significant bits of the second rotator's output port 408. Now, the upper 16-bit of the masking unit's output port 414, corresponding to the first rotator, comprise the original bit positions "18...31,0,1", whereas the lower 16-bit of the masking unit's output port 416, corresponding to the second rotator, comprise the original bit positions "2...15,16,17" as illustrated in Fig. 4.

With reference now to Fig. 5, there is depicted a block diagram of a device 500 for performing shift/rotate operations on operands of any of a plurality of sizes between 2 and 2N bits and, alternatively, for performing shift/rotate operations on two operands each of any of a plurality of sizes between 1 and N bits according to the present invention. For the sake of clarity, in the implementation as presented in Fig. 5 and described in the following, N is equal 16.

The device 500 comprises four blocks, namely, a first 16-bit rotator array 501, a second 16-bit rotator array 502, a masking unit 504 and a look-up table (LUT) 506. Both rotator arrays are each provided with a 16 bit wide input port 508, 509 and 5 bit wide control ports 510, 511 for specifying the shift amount. Furthermore, both rotator arrays have got each a 16 bit wide output port 512, 513 for transferring the rotated data to the masking unit 504.

The masking unit 504 takes the data via two respective 16-bit input ports 514, 515 and outputs the masked data via one single 32-bit output port 516. Additionally, the masking unit 504 is equipped with a 16-bit masking port 518 for specifying the mask information. The mask information origin from the look-up-table 506, which is provided with a respective output port 520. Furthermore, the LUT 506 has got a 4 bit wide shift amount input port 522 for inputting the 4 least significant bits of the 5-bit shift amount control signal on shift amount data bus 524. In addition, all 5 bits of the shift amount control signal are inputted into the masking unit 504 via a respective input port 526, whereas further control signals, e.g., for specifying whether to rotate or to shift in two times 16-bit mode or in 32-bit mode, are inputted via another input port 528.

A 32-bit input port 530 is provided for receiving input data to be rotated/shifted. The data gets split into two times 16-bit to be fed into the first and second rotator array via the respective input ports 508 and 509.

Each of the two 16-bit rotator arrays are may be formed by state of the art rotators/shifters configured to perform rotates and shifts to the left and right based on a shift amount with a 2-complement binary format. Different implementations of the rotator array may be possible.

The first rotator array 501 shows the internal configuration. It is based on logarithmic rotators with increasing rotate amount according to the prior art. The first rotator array 501 comprises a 1-bit rotator 532, a 2-bit rotator 534, a 4-bit rotator 536 and an 8-bit rotator 538, each having an input port, output port and a control port, whereby the output port of one rotator is connected to the input port of the following rotator and the control port of each rotator is connected to its corresponding signal of the control port 510. When activated by the respective signal received via the 4 bit wide control port 510, the rotator rotates the input bit pattern, i.e., the 1-bit rotator 532 rotates by 1 bit position, the 2-bit rotator 534 rotates by 2 bit positions and so on. In case a rotator is inactive, the bit pattern present at its input port is transferred through to the output port without applying any changes. Therefore, if the 1-bit rotator 532 and the 4-bit rotator 536 are the only active rotators at a given time, a bit pattern at the input port 508 gets rotated by 5 bit positions before reaching the output port 512.

According to the present invention, the two rotator arrays 501 and 502 each generate the results as needed for the 2 half word SIMD 16-bit rotates, whereby word denotes 32 bit wide data and half word denotes 16 bit wide data. The task of the masking unit 504 is, first of all, to perform the masking as needed for shifts. In order to support the 32-bit word rotates, the masking unit 504 is able to exchange bit[i] with bit[i+16] with i ={0..15} for generating the correct rotate/shift word result from the results of the two 16-bit rotate arrays.

The bits that have to be exchanged are calculated by a control logic 540 based on the shift amount, the operation code representing the operation to perform and the LUT 506.

The control logic generates the three vectors to control the operation of the masking unit 504, namely, a 2-bit mask vector 542 (Mask[0..1]), a 32-bit fix vector 544 (Fix[0..31]) and a 32-bit cross vector 546 (Cross[0..31]). A fourth 32-bit direct vector (Direct[0..31]) is generated by applying the logical NOR operation on the cross vector 546 and the fix vector 544.

As to the mask vector 542, Mask[0] determines for the upper half word how in the case of shifts the bits that are shifted-in need to be set. Mask[1] does the same for the lower half word. The signals of the cross vector 546 determine whether a particular one of the bits has to be exchanged or not. The signals of the fix vector determine whether or not the mask input is selected or not.

In the embodiment shown in Fig. 5, a 1-value in the cross vector, i.e., Cross[i]=1, is prioritized over a 1-value in the fix vector, i.e., Fix[i]=1. In other words, the data is exchanged for bit i instead of selecting the mask input as the result.

Since the direct vector Direct[0..31] is the NOR of Cross and Fix, it is the default case, if Fix[i] and Cross[i] are both 0. In this case the result of the rotator array is directly routed to the output.

It should be noted that the LUT has an output vector of only 16 bit, instead of the 32 bit. The expansion to 32 bit is done as part of the control logic 540, which is needed for the word rotate/shifts.

The device according to the present invention has the advantage that there is no additional delay in the data path since, the two half-word rotators have less logical stages as a word rotator array, because the two half-word rotator arrays need no word ability extension and the word extension is integrated into the masking unit without additional delay. It should be noted that the masking unit is needed to perform the shift operations, anyway. Two homogenous rotator arrays build out of 2:1 multiplexers having an equal delay for all output signals may form the two half-word rotator arrays. In addition, the amount of long wires is advantageously reduced, because all connectivity needed is within the 16-bit range.

The outputs of 32 3:1-selectors are forming the 32-bit data outputted by the output port 516. For the sake of clarity only a first 3:1-selector 550 for generating output bit position

[0], a second 3:1-selector 552 for generating output bit position [1], a third 3:1-selector 554 for generating output bit position [30] and a fourth 3:1-selector 556 for generating output bit position [31] are shown.

Each 3:1-selector is equipped with three input ports. In the embodiment of the present invention as illustrated in Fig. 5, the 16 3:1-selectors generating the output bit positions 0 to 15 of the masking unit have the left input port connected to the respective 16 bit positions [0:15] of the first rotator array's output, the middle input port of each 16 3:1-selectors is connected to the bit position [0] of the mask vector 542, and the right input port is connected to the respective 16 bit positions [0:15] of the second rotator array's output.

Correspondingly, the 16 3:1-selectors generating the output bit positions 16 to 31 of the masking unit have the left input port connected to the respective 16 bit positions [0:15] of the second rotator array's output, the middle input port of each 16 3:1-selectors is connected to the bit position [1] of the mask vector 542, and the right input port is connected to the respective 16 bit positions [0:15] of the first rotator array's output.

In conclusion of this, the left input port of a 3:1-selector is selected, whenever the corresponding bit of the direct vector is a 1-value and the corresponding bits of the cross and fix vector are both a 0-value. The middle input port of a 3:1-selector is selected, whenever the corresponding bit of the fix is a 1-value and the corresponding bit of the cross vector is a 0-value. The right input port of a 3:1-selector is selected, whenever the corresponding bit of the cross vector is a 1-value.

The 3:1-selectors are implemented by using AND/OR gates, a gate array based implementation or any other technique for realizing the 3:1 selector functionality.

CLAIMS

1. A device for performing rotate operations on operands having a size of 2N bits and, alternatively, for performing rotate operations on two operands each having a size of N bits, whereby N is an integer number, the device comprising:

an input port for receiving 2N bits numbered 1 to 2N,

a first left rotating circuit having an input for receiving bits 1 to N of said input port,

a second left rotating circuit having an input for receiving bits N+1 to 2N of said input port, whereby said first and said second rotate circuit are configured to operate each with N bits at a time, and

a control unit being adapted for exchanging

- the M least significant bits of the output of said first rotate circuit with the M least significant bits of the output of said second rotate circuit, when M<N,
- the 2N-M most significant bits of the output of said first rotate circuit with the 2N-M most significant bits of the output of said second rotate circuit, when M>N,
- the N bits of said first rotate circuit with the N bits of said second rotate circuit, when M=N,

whenever an input having the width of 2N is to be rotated by M bits.

2. A device for performing rotate operations on operands having a size of 2N bits and, alternatively, for performing rotate operations on two operands each having a size of N bits, whereby N is an integer number, the device comprising:

an input port for receiving 2N bits numbered 1 to 2N,

a first right rotating circuit having an input for receiving bits 1 to N of said input port,

a second right rotating circuit having an input for receiving bits N+1 to 2N of said input port, whereby said first and said second rotate circuit are configured to operate each with N bits at a time, and

a control unit being adapted for exchanging

- the M most significant bits of the output of said first rotate circuit with the M most significant bits of the output of said second rotate circuit, when M<N,
- the 2N-M least significant bits of the output of said first rotate circuit with the 2N-M least significant bits of the output of said second rotate circuit, when M>N,
- the N bits of said first rotate circuit with the N bits of said second rotate circuit, when M=N,

whenever an input having the width of 2N is to be rotated by M bits.

3. The device according to claim 1 or 2, wherein the control unit is adapted to mask individual bit positions in order to perform a shift operation instead of a rotate operation.

4. The device according to claim 1,2 or 3, wherein the first and second rotate circuits are each formed by rotator arrays capable of rotating N bit wide data.

5. The device according to one of the preceding claims, wherein said control unit comprises 2N selectors forming the 2N-bit data to be outputted.

6. The device according to claim 5, wherein the N selectors generating the output bit positions 1 to N of the masking unit have

a first input port connected to the respective N bit positions of the first rotate circuit's output, and

a second input port is connected to the respective N bit positions of said second rotate circuit's output.

7. The device according to claim 5 or 6, wherein the N selectors generating the output bit positions N+1 to 2N of the masking unit have

the first input port connected to the respective N bit positions of the second rotate circuit's output, and

the second input port is connected to the respective N bit positions of said first rotate circuit's output.

8. The device according to one of the claims 5 to 7, wherein the N selectors generating the output bit positions 1 to N of the masking unit have

a third input port connected to the bit position 1 of a mask vector.

9. The device according to one of the claims 5 to 8, wherein the N selectors generating the output bit positions N+1 to 2N of the masking unit have

the third input port connected to the bit position 2 of a mask vector.

10. The device according to one of the claims 5 to 9, wherein the selectors are formed by 2N 3:1-selectors.

11. The device according to one of the preceding claims, where said first and second rotate circuit are operated independent from each other and receive independent shift amounts M.

12. A method for performing left rotate operations on operands having a size of 2N bits and, alternatively, for performing rotate operations on two operands each having a size of N bits, whereby N is an integer number, the method comprising the steps of:

receiving 2N bits numbered 1 to 2N,

rotating independently at each case bit positions 1 to N and N+1 to 2N, respectively, exchanging the M least significant bits of the rotated bit positions 1 to N with the M least significant bits of the bit positions N+1 to 2N, when M<N and the input is to be rotated by M bits,

exchanging the 2N-M most significant bits of the rotated bit positions 1 to N with the 2N-M most significant bits of the rotated bit positions N+1 to 2N, when M>N and the input is to be rotated by M bits,

exchanging the rotated bit positions 1 to N with the rotated bit positions N+1 to 2N, when M=N and the input is to be rotated by M bits.

13. A method for performing right rotate operations on operands having a size of 2N bits and, alternatively, for performing rotate operations on two operands each having a size of N bits, whereby N is an integer number, the method comprising the steps of:

receiving 2N bits numbered 1 to 2N,

rotating independently at each case bit positions 1 to N and N+1 to 2N, respectively,

exchanging the M most significant bits of the rotated bit positions 1 to N with the M most significant bits of the bit positions N+1 to 2N, when M<N and the input is to be rotated by M bits,

exchanging the 2N-M least significant bits of the rotated bit positions 1 to N with the 2N-M least significant bits of the rotated bit positions N+1 to 2N, when M>N and the input is to be rotated by M bits,

exchanging the rotated bit positions 1 to N with the rotated bit positions N+1 to 2N, when M=N and the input is to be rotated by M bits.

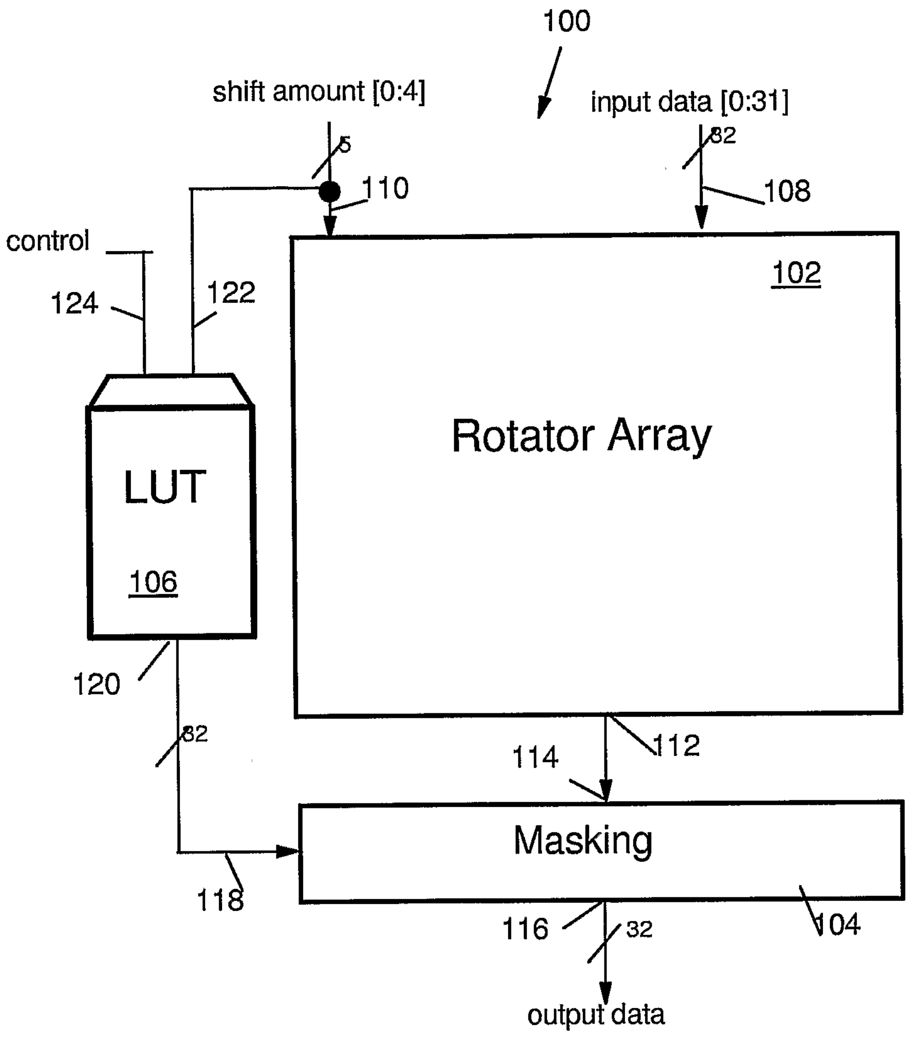


FIG. 1 Prior Art

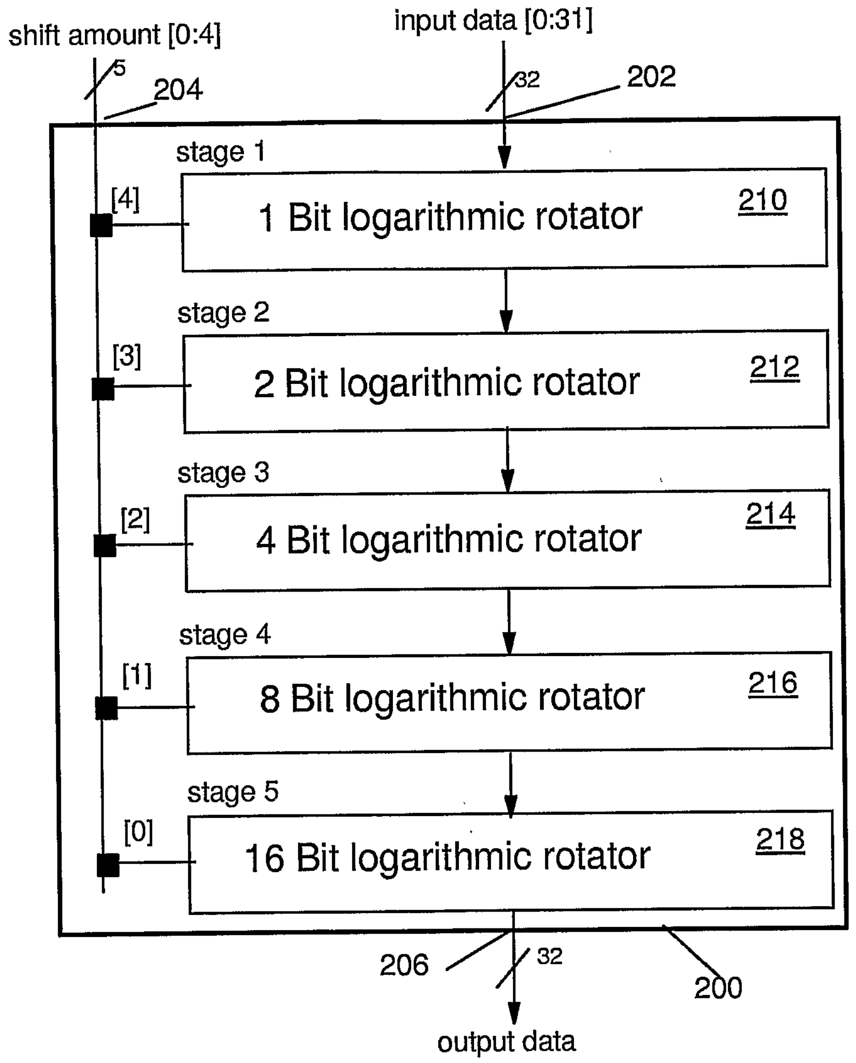


FIG. 2 Prior Art

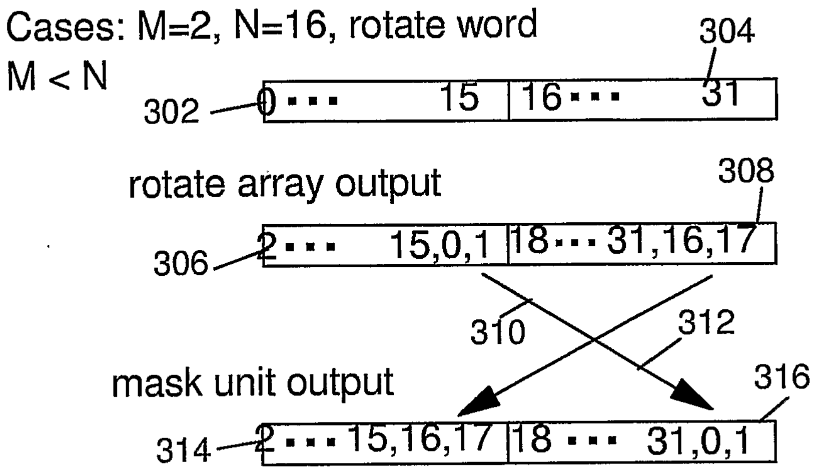

Cases: M=2, N=16, rotate word
M < N
304
302
0 ▪▪▪ 15
16 ▪▪▪ 31
rotate array output
308
306
2 ▪▪▪ 15,0,1
18 ▪▪▪ 31,16,17
310
312
mask unit output
316
314
2 ▪▪▪ 15,16,17
18 ▪▪▪ 31,0,1

FIG. 3

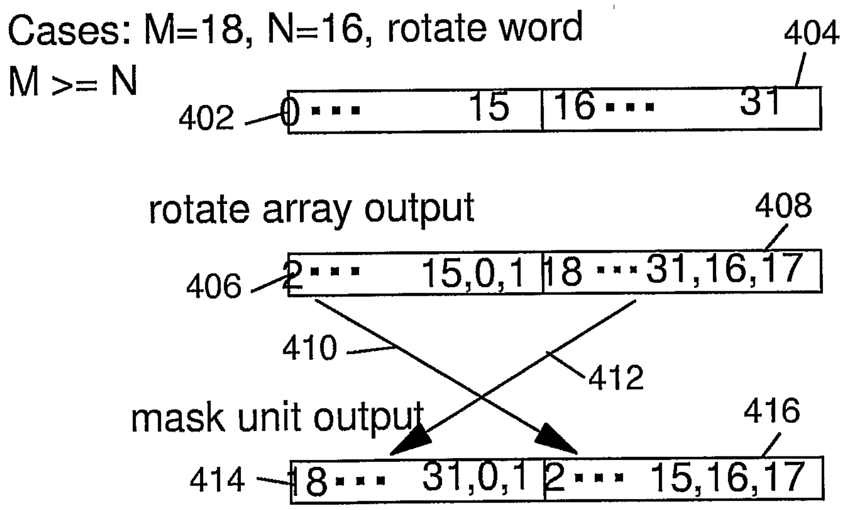

Cases: M=18, N=16, rotate word
M >= N
404
402
0 ▪▪▪ 15
16 ▪▪▪ 31
rotate array output
408
406
2 ▪▪▪ 15,0,1
18 ▪▪▪ 31,16,17
410
412
mask unit output
416
414
18 ▪▪▪ 31,0,1
2 ▪▪▪ 15,16,17

FIG. 4

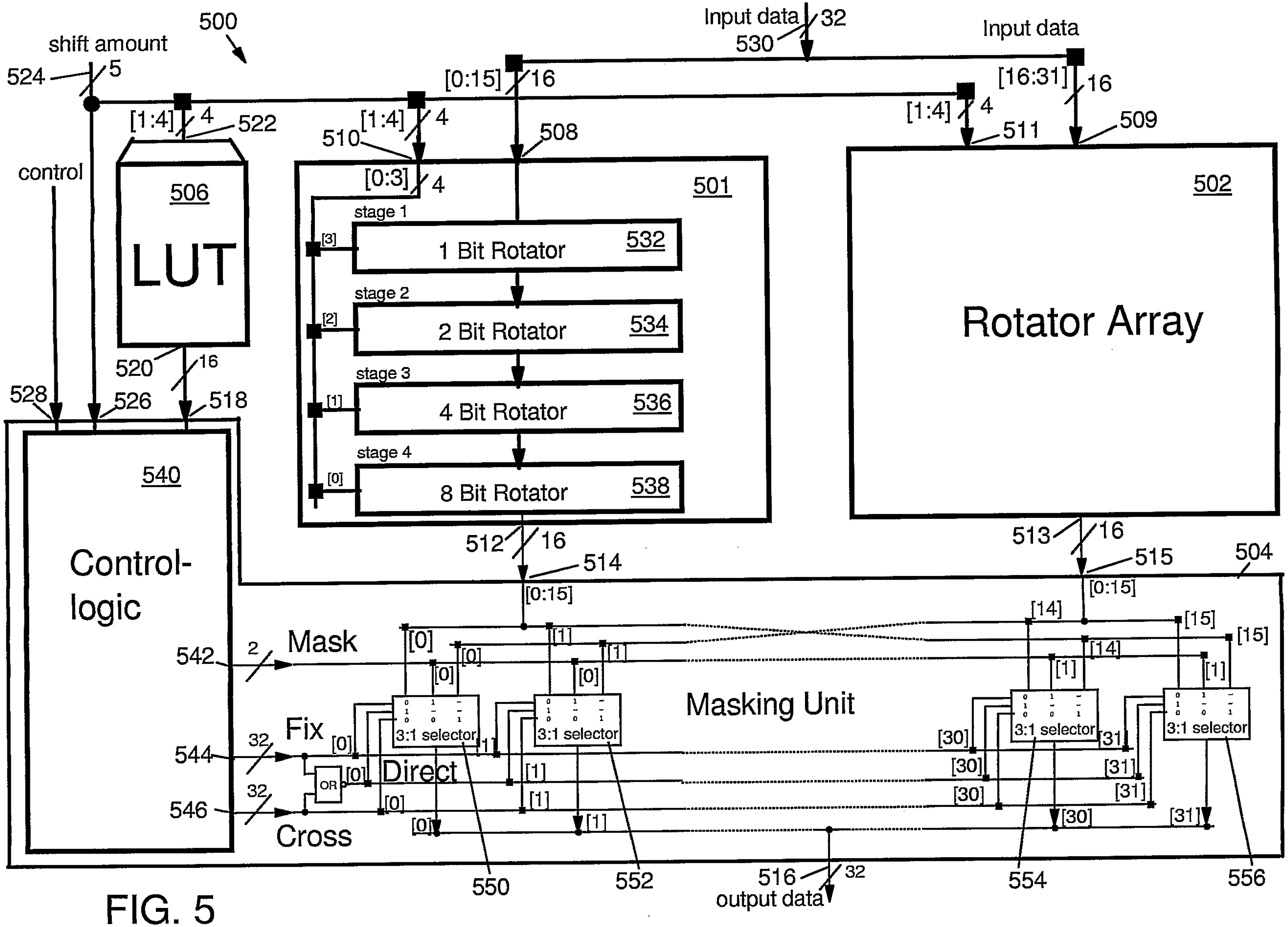

500
shift amount
524
5
control
[1:4]
4
522
506
LUT
520
16
528
526
518
540
Control-
logic
Input data
530
32
[0:15]
16
[1:4]
4
510
508
[0:3]
4
501
stage 1
1 Bit Rotator
532
stage 2
2 Bit Rotator
534
stage 3
4 Bit Rotator
536
stage 4
8 Bit Rotator
538
[3]
[2]
[1]
[0]
512
16
514
Input data
[16:31]
16
[1:4]
4
511
509
502
Rotator Array
513
16
515
504
[0:15]
[0:15]
542
2
Mask
544
32
Fix
546
32
Cross
OR
Direct
3:1 selector
3:1 selector
3:1 selector
3:1 selector
Masking Unit
[0]
[1]
[14]
[15]
[30]
[31]
550
552
554
556
516
32
output data

FIG. 5