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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
6 March 2003 (06.03.2003)

PCT

(10) International Publication Number
WO 03/019352 A1

(51) International Patent Classification⁷: G06F 7/72, H03M 13/27

(21) International Application Number: PCT/US02/27958

(22) International Filing Date: 29 August 2002 (29.08.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/316,135	29 August 2001 (29.08.2001)	US
09/981,130	17 October 2001 (17.10.2001)	US

(71) Applicant: **QUALCOMM INCORPORATED** [US/US]; 5775 Morehouse Drive, San Diego, CA 92121 (US).

(72) Inventors: **PATEL, Shimman**; 9406 Galvin Avenue, San Diego, CA 92126 (US). **KAN, Andrew**; 461 Auburn Way, #3, San Jose, CA 95129 (US). **DHAWAN, Rajat**; 9928 Kika Court, #2818, San Diego, CA 92129 (US).

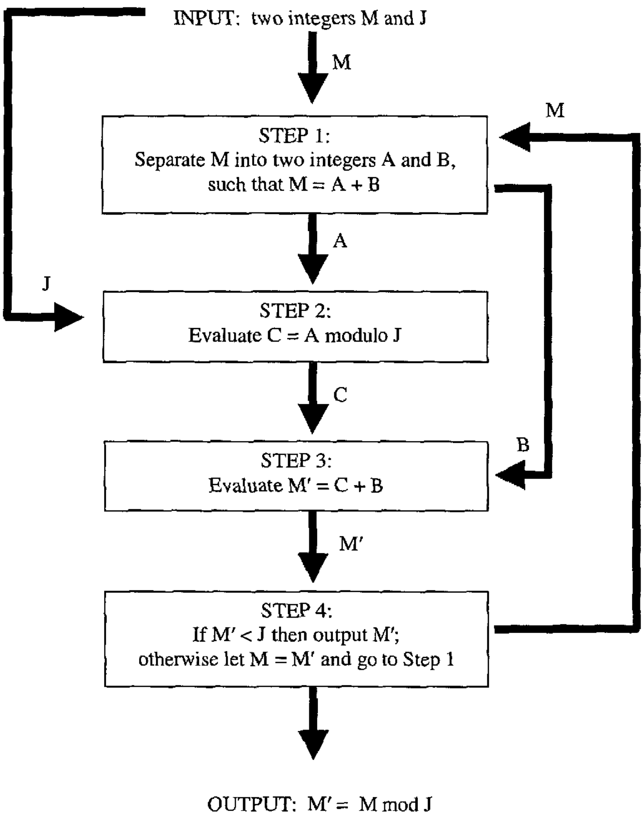
(74) Agents: **WADSWORTH, Philip R.** et al.; Qualcomm Incorporated, 5775 Morehouse Drive, San Diego, CA 92121 (US).

(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, 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, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, 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,

[Continued on next page]

(54) Title: FAST, ITERATIVE SYSTEM AND METHOD FOR EVALUATING A MODULO OPERATION WITHOUT USING DIVISION


INPUT: two integers M and J
M
STEP 1:
Separate M into two integers A and B,
such that M = A + B
M
A
J
STEP 2:
Evaluate C = A modulo J
C
B
STEP 3:
Evaluate M' = C + B
M'
STEP 4:
If M' < J then output M';
otherwise let M = M' and go to Step 1
OUTPUT: M' = M mod J

(57) Abstract: A fast, iterative technique for evaluating M modulo J, which may be easily implemented in hardware. In the illustrative embodiment, the invention includes a first circuit (10) for decomposing M into two integers A and B = M - A; a second circuit (20) for evaluating (A modulo J); a third circuit (30) for evaluating M’ = (A modulo J) + B; and, a fourth circuit (40) for determining whether to output M’ as the final answer, or to feedback M’ to said first means to evaluate M’ modulo J.

WO 03/019352 A1

ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report*

FAST, ITERATIVE SYSTEM AND METHOD FOR EVALUATING A MODULO OPERATION WITHOUT USING DIVISION

BACKGROUND OF THE INVENTION

[0001] This application claims the benefit of provisional U.S. Application Serial No. 60/316,135, entitled "FAST, ITERATIVE SYSTEM AND METHOD FOR EVALUATING A MODULO OPERATION WITHOUT USING DIVISION," filed August 29, 2001.

Field of the Invention:

[0002] The present invention relates to electronic circuits and systems. More specifically, the present invention relates to hardware implementation of arithmetic operators for use in communications systems.

Description of the Related Art:

[0003] Interleaving of coded data for transmission (in combination with deinterleaving at the receiver) has been an effective method of transforming burst errors into statistically independent errors. Interleaving reorders the coded data sequence in an apparently random order, such that after the data is returned to its proper sequence by the deinterleaver, error bursts are spread out in time. Thus errors within one code word appear to be independent.

[0004] Previous transmission standards used a method of interleaving involving bit reversal of parts of the binary representation of bin numbers to randomize the data sequence. However, a new wireless standard, CDMA2000, requires a deinterleaver which uses a modulo operation. In particular, it requires the evaluation of M modulo J for $0 < M < 2^N$ and J = 3, 6, 12, 24, 48, and 96. The operation M modulo J returns the remainder of M divided by J. Currently, there is no hardware design which implements this modulo operation.

[0005] Hence, a need exists in the art for a fast system and method for evaluating M modulo J which can be easily implemented in hardware.

SUMMARY OF THE INVENTION

[0006] The need in the art is addressed by a technique which provides a fast, iterative method for evaluating M modulo J (M mod J) which can be easily implemented in hardware for use in such applications as deinterleavers in communications systems.

[0007] In an illustrative implementation, the invention includes the steps of: 1) decomposing M into two integers A and B = M – A; 2) evaluating C = A modulo J; 3) evaluating M′ = C + B; and 4) determining whether to output M′ as the final answer, or to feedback M′ to said first means to evaluate M′ modulo J.

[0008] The method may be easily implemented in hardware where for example in Step 1, the integer A is a power of 2 and, in Step 2, A modulo J is stored in a small look-up table for $A = 2^0, 2^1, 2^2 \ldots 2^N$.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] FIG. 1 is a block diagram of a typical wireless communications system.

[0010] FIG. 2 is a block diagram of a typical bit-reversal order deinterleaver.

[0011] FIG. 3 is a flow diagram of an iterative algorithm for evaluating M modulo J in accordance with the teachings of the present invention.

[0012] FIG. 4 is a block diagram of a hardware implementation for evaluating M modulo J in accordance with the teachings of the present invention.

DESCRIPTION OF THE INVENTION

[0013] Illustrative embodiments and exemplary applications will now be described with reference to the accompanying drawings to disclose the advantageous teachings of the present invention.

[0014] While the present invention is described herein with reference to illustrative embodiments for particular applications, it should be understood that the invention is not limited thereto. Those having ordinary skill in the art and access to the teachings provided herein will recognize additional modifications, applications, and embodiments within the scope thereof and additional fields in which the present invention would be of significant utility.

[0015] FIG. 1 is a block diagram of a typical communications system 200 using a deinterleaver. The cdma2000 transmission standard calls for an interleaver and a deinterleaver which evaluate M modulo J for $0 < M < 2^N$ and J = 3, 6, 12, 24, 48, and 96. The operation M modulo J (M mod J) returns the remainder of M divided by J.

[0016] FIG. 2 is a block diagram of a typical bit-reversal order deinterleaver 94. The deinterleaver includes a demultiplexer 102, a multiplexer 106, and a circuit 104 for evaluating $A_i = 2^m(i \bmod J) + BRO_m(\lfloor i/J \rfloor)$, where $\lfloor x \rfloor$ indicates the largest integer less than or equal to *x*, $BRO_m(y)$ indicates the bit-reversed *m*-bit value of *y* (for example, $BRO_3(6) = 3$), and *m* and *J* are given in the following table for a deinterleaver of size N:

Interleaver Size	**m**	**J**
48	4	3
96	5	3
192	6	3
384	6	6
768	6	12
1,536	6	24
3,072	6	18
6,144	7	48
12,288	7	96
144	4	9
288	5	9
576	5	18
1,152	6	18
2,304	6	36

4,608	7	36
9,216	7	72
18,432	8	72
36,864	8	144
128	7	1

[0017] The symbols input to the deinterleaver 94 are written sequentially at addresses 0 to N-1 in the demultiplexer 102. At the output of the deinterleaver, the symbols are read out in permuted order from address A_i, for i = 0 to N-1. The circuit 104 evaluates A_i, and the multiplexer 106 combine the symbols sequentially from A_0 to A_{N-1}.

[0018] In computing A_i, the circuit 104 needs to evaluate M modulo J for M = $2^m i$ for i = 0 to N-1, and m and J given by the above table. Ideally, the M modulo J operation should be implemented in hardware.

[0019] The present invention provides a fast, iterative method for evaluating M modulo J (M mod J) which can be easily implemented in hardware for use in applications such as the deinterleaver described above.

[0020] A recursive formula for computing M modulo J can be derived from the following algebraic manipulations:

[0021] Let M be an integer from 0 to 2^N. M can be expressed as a sum of two other integers:

[0022] $M = A + B.$ (1)

[0023] For any integer J, there exists unique integers q_a, q_b, r_a, and r_b such that:

[0024] $A = q_a J + r_a$,

[0025] and

[0026] $B = q_b J + r_b$,

(2)

[0027] where $J > r_a, r_b > 0$.

[0028] Therefore,

[0029] $A + B = (q_a + q_b)J + (r_a + r_b)$ (3)

[0030] It follows that:

[0031] M modulo J = (A + B) modulo J

[0032] = $[(q_a + q_b)J + (r_a + r_b)]$ modulo J

[0033] = $(q_a + q_b)J$ modulo J + $(r_a + r_b)$ modulo J

[0034] = $(r_a + r_b)$ modulo J, (4)

[0035] since $(q_a + q_b)J$ is an integer multiple of J, and therefore has a remainder of 0 when divided by J. Thus $(q_a + q_b)J$ modulo J is equal to 0. By similar reasoning, adding a term which is an integer multiple of J will not affect the modulo J operation:

[0036] $(r_a + r_b)$ modulo J = $(r_a + q_bJ + r_b)$ modulo J

[0037] = [(A modulo J) + B] modulo J, (5)

[0038] since A modulo J = $(q_aJ + r_a)$ modulo J = r_a.

[0039] Therefore:

[0040] M modulo J = M′ modulo J, (6)

[0041] where M′ = A modulo J + B. This leads to an iterative algorithm for evaluating M modulo J.

[0042] FIG. 3 is a flow diagram of an iterative algorithm for evaluating M modulo J in accordance with the teachings of the present invention. This method includes the following steps:

[0043] decomposing M into two integers A and B = M – A;

[0044] evaluating C = A modulo J;

[0045] evaluating M′ = C + B; and,

[0046] determining whether to output M′ as the final answer, or to repeat with M = M′ to evaluate M′ modulo J.

[0047] This method can be readily implemented in hardware if in Step 1, the integer A is a power of 2, and in Step 2, A modulo J is stored in a look-up table for A = 2^0, 2^1, 2^2...2^N. Let M be an integer in binary representation, that is, $M = \sum \alpha_i 2^i$ for i = 0 to N, and α_i = 0 or 1. In step 1, A is chosen to be $\alpha_i 2^i$. Since α_i is either 0 or 1, A is either 0 or 2^i. Thus in Step 2, A modulo J is 0 for $\alpha_i = 0$, or 2^i modulo J for $\alpha_i = 1$. Then, 2^i modulo J can be evaluated through a small look-up table storing 2^i modulo J for i = 0 to N, and the values of J required (for this particular application, J = 3, 6, 12, 24, 48, or 96). The algorithm is repeated recursively, starting with i = N, and reducing i by 1 with each iteration, until a final answer is reached when M′ < J.

[0048] In one case, the algorithm does not converge. An additional step between Step 3 and Step 4 is required to insure convergence to the correct answer:

[0049] Step 3.5: if the bitwise AND between M′ and J equals J, then let M = M′ - J and return to Step 1, otherwise output M′ as the final answer.

[0050] The following is a numerical example to further illustrate this method:

[0051] EXAMPLE: Find M modulo J for M = 27, J =6.

[0052] $M = 11011_{bin} = 2^4 + 2^3 + 2^1 + 2^0$

[0053] Let $A = \alpha_4 2^4 = 10000_{bin} = 16$, and $B = M–A = 1011_{bin} = 11$

[0054] From a look-up table, find C = A modulo J = 16 modulo 6 = 4

[0055] Form $M' = C+B = 4+11 =15 = 1111_{bin}$

[0056] Step 3.5: Check if (M′&&J = J): 1111_{bin} && $110_{bin} = 110_{bin}$ = J, therefore let M′= M′-J = 15-6=9

[0057] Step 4: Check if (M′<J): 9 > 6, therefore let M = M′ = 9

[0058] and repeat

[0059] Step 1: $M = 9 = 1001_{bin}$

[0060] Let $A = 1000_{bin} = 8$, and B = M-A = 1

[0061] Step 2: From a look-up table, find C = A modulo J = 8 modulo 6 = 2

[0062] Step 3: Form $M' = C+B = 2+1 =3 = 11_{bin}$

[0063] Step 3.5: Check if (M′&&J = J): 11_{bin} && $110_{bin} = 10_{bin} \neq J$, therefore continue to Step 4

[0064] Step 4: Check if (M′<J): 3 < 6, therefore stop. The final answer is 3.

[0065] Therefore, 27 modulo 6 = 3.

[0066] FIG. 4 is a block diagram of an illustrative hardware implementation for evaluating M modulo J in accordance with the teachings of the present invention. The architecture includes a first circuit 10 for decomposing M into two integers A and B = M – A (STEP 1); a second circuit 20 for evaluating A modulo J (STEP 2); a third circuit 30 for evaluating M′ = (A modulo J) + B (STEP 3); a fourth circuit 40 for determining whether to output M′ as the final answer, or to feedback M′ to the first circuit 10 to evaluate M′ modulo J (STEP 4); and, a fifth circuit 50 for ensuring convergence (STEP 3.5).

[0067] The inputs to this circuit are two integers M and J. Initial conditions are set such that i = N, and $B_N = M - \alpha_N 2^N$.

[0068] The first circuit 10 includes a multiplexer M1 which passes $B_N = (M - \alpha_N 2^N)$ on the first iteration, and passes $B_i = (M' - \alpha_i 2^i)$ on all subsequent iterations, where i is an iteration counter starting with N and counting down. The output of the multiplexer M1 (equivalent to B in the derivations) is passed to the third circuit 30.

[0069] The second circuit 20 includes a look-up table 22 which stores 2^i modulo J for i = 0 to N. The second circuit 20 further includes a multiplexer M2 which passes 0 if ($\alpha_i = 0$), and passes C_i if ($\alpha_i = 1$). The output of M2 is therefore equivalent to A modulo J, where $A = \alpha_i 2^i$. This output is passed to the third circuit 30.

[0070] The third circuit 30 includes an adder A1 which adds the outputs of the first and second circuits and passes the result M′ = (A modulo J) + B to the fifth circuit 50.

[0071] The fifth circuit 50 includes a multiplexer M3 which passes J if the bitwise AND of M′ and J equals J, otherwise it passes 0. The output of M3 is subtracted from M′ by an adder A2, and the result is passed to the fourth circuit 40.

[0072] The fourth circuit 40 includes a multiplexer M4 which passes M′ as the final output if (M′ < J); otherwise i is set to i-1, and M′ is fed back to the first circuit 10. The feedback loop is repeated until the condition M′ < J is met. Then M′ is output as the final solution to M modulo J.

[0073] Hence, the new hardware implementation of FIG. 3 evaluates the operation M modulo J.

[0074] Thus, the present invention has been described herein with reference to a particular embodiment for a particular application. Those having ordinary skill in the art and access to the present teachings will recognize additional modifications, applications and embodiments within the scope thereof. For example, those skilled in the art will appreciate that for the algorithm can be used in applications other than a deinterleaver in a communications system. Further, the invention can be used in any digital signal processing (DSP) application requiring the operation M modulo J.

[0075] It is therefore intended by the appended claims to cover any and all such applications, modifications and embodiments within the scope of the present invention.

[0076] Accordingly,

WHAT IS CLAIMED IS:

1. A system for evaluating M modulo J, where J is an integer and M is an integer expressed in binary form ($M = \sum_{i=0}^{N} \alpha_i 2^i$), where α_i is 0 or 1, and $N+1$ is the number of digits in a binary word) comprising:

a first circuit for decomposing M into two integers A and B = M – A;

a second circuit for evaluating (A modulo J);

a third circuit for evaluating M′ = (A modulo J) + B; and

a fourth circuit for outputting M′ or feeding M′ back to the first means to evaluate M′ modulo J.

2. The system of Claim 1, wherein the first circuit includes a multiplexer M1 which passes $B_N = (M - \alpha_N 2^N)$ to the second circuit on a first iteration, and passes $B_i = (M' - \alpha_i 2^i)$ on all subsequent iterations, where i is an iteration counter starting with N and counting down.

3. The system of Claim 1, wherein the second circuit includes a look-up table that stores $C_i = 2^i$ modulo J for i = 0 to N.

4. The system of Claim 3, wherein the second circuit further includes a multiplexer M2 that passes 0 to the third circuit when ($\alpha_i = 0$), and passes C_i when ($\alpha_i = 1$).

5. The system of Claim 1, wherein the third circuit includes an adder A1 whose inputs are B_i and $(\alpha_i C_i)$ and which passes its output $M' = B_i + (\alpha_i C_i)$ to the fourth circuit.

6. The system of Claim 1, wherein the fourth circuit includes a multiplexer M4 that passes M' as a final output if $(M' < J)$; otherwise i is set to i-1, and M' is fed back to the first circuit.

7. The system of Claim 1, wherein the circuit further includes fifth circuit for ensuring convergence.

8. The system of Claim 7, wherein the fifth circuit includes a multiplexer M3 that passes J when the bitwise AND of M' and J equals J, otherwise it passes 0.

9. The system of Claim 8, wherein the output of the multiplexer M3 is subtracted from M' by an adder A2 and the result is passed to the fourth circuit.

10. A deinterleaver comprising:

a demultiplexer;

a multiplexer; and

a circuit for connecting the outputs of the demultiplexer to the inputs of the multiplexer, wherein the circuit includes a system for evaluating M modulo J comprising:

a first circuit for decomposing M into two integers A and B = M – A;

a second circuit for evaluating (A modulo J);

a third circuit for evaluating M′ = (A modulo J) + B; and

a fourth circuit for outputting M′ or feeding M′ back to the first circuit to evaluate M′ modulo J.

11. A method for evaluating M modulo J including the steps of:

decomposing M into two integers A and B = M – A;

evaluating (A modulo J);

evaluating M′ = (A modulo J) + B; and,

determining whether to output M′ as the final answer, or to feedback M′ to the decomposing step to evaluate M′ modulo J.

12. The method of Claim 11, wherein the decomposing involves passing $B_N = (M - \alpha_N 2^N)$ to the evaluating (A modulo J) step on a first iteration, and passing $B_i = (M' - \alpha_i 2^i)$ on all subsequent iterations, where i is an iteration counter starting with N and counting down.

13. The method of Claim 11, wherein evaluating (A modulo J) involves using a look-up table that stores $C_i = 2^i$ modulo J for i = 0 to N.

14. The method of Claim 13, wherein the evaluating (A modulo J) further

includes passing 0 to the evaluating $M' = (A \text{ modulo } J) + B$ step when $(\alpha_i = 0)$, and passes C_i when $(\alpha_i = 1)$.

15. The method of Claim 11, wherein the evaluating $M' = (A \text{ modulo } J) + B$ step involves using an adder A1 whose inputs are B_i and $(\alpha_i C_i)$ and which passes its output $M' = B_i + (\alpha_i C_i)$ to the determining step.

16. The method of Claim 11, wherein the determining involves passing M' as a final output when $(M' < J)$; otherwise i is set to i-1, and M' is fed back to the decomposing step.

17. The method of Claim 12, further comprising ensuring convergence has occurred.

18. The method of Claim 17, wherein ensuring convergence includes passes J when the bitwise AND of M' and J equals J, otherwise passing 0.

FIG. 1

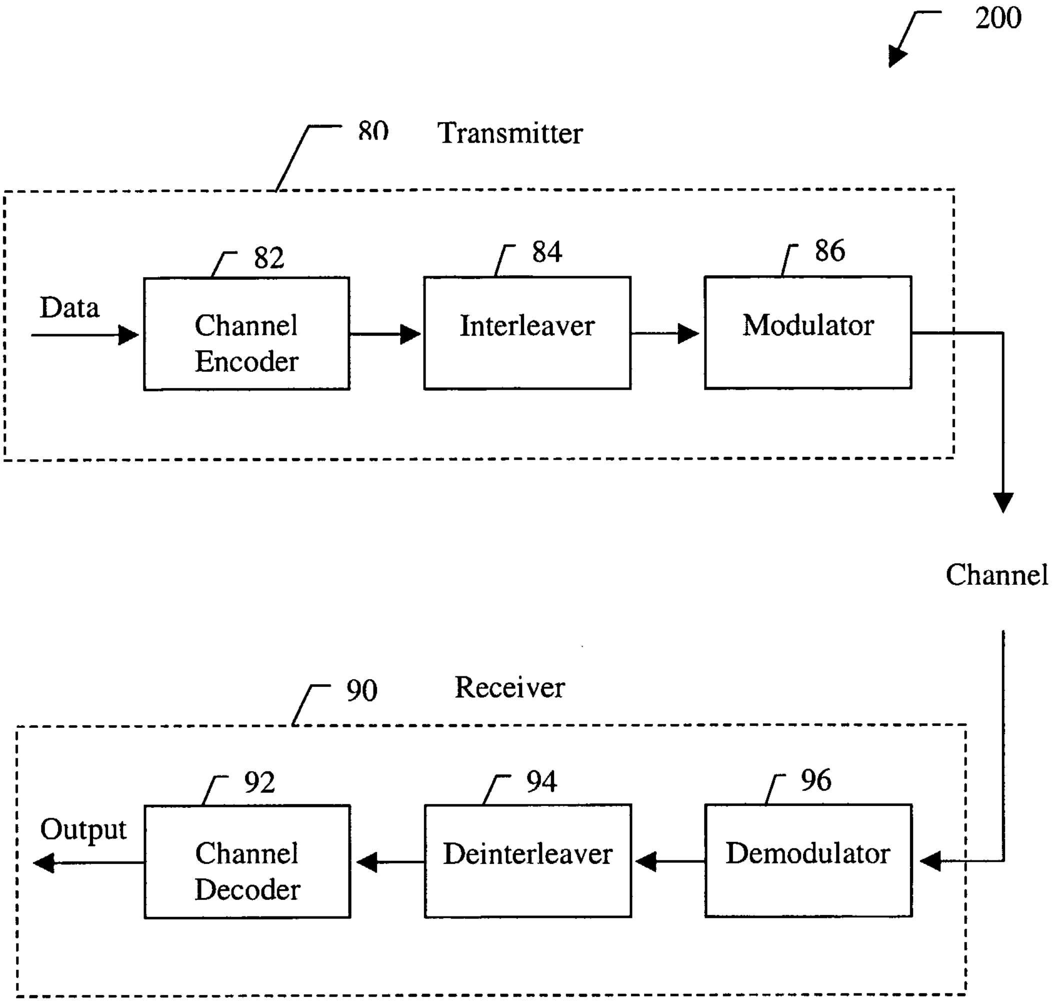

200
80
Transmitter
82
84
86
Data
Channel Encoder
Interleaver
Modulator
Channel
90
Receiver
92
94
96
Output
Channel Decoder
Deinterleaver
Demodulator

FIG. 2

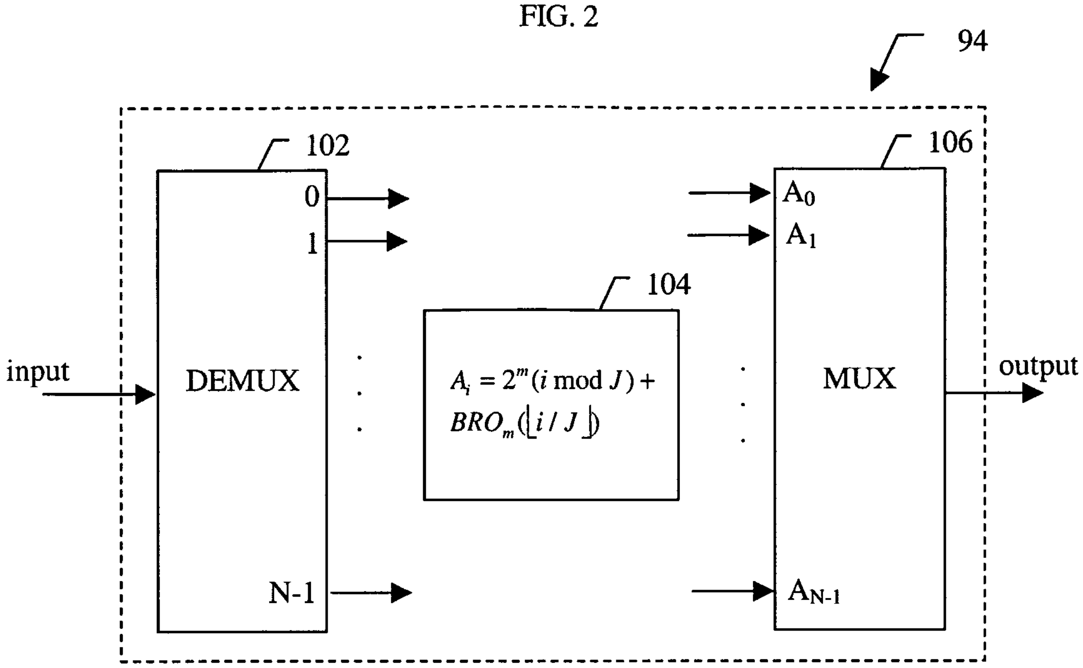

94
102
106
104
input
DEMUX
0
1
N-1
$A_i = 2^m(i \bmod J) + BRO_m(\lfloor i/J \rfloor)$
MUX
A_0
A_1
A_{N-1}
output

FIG. 3

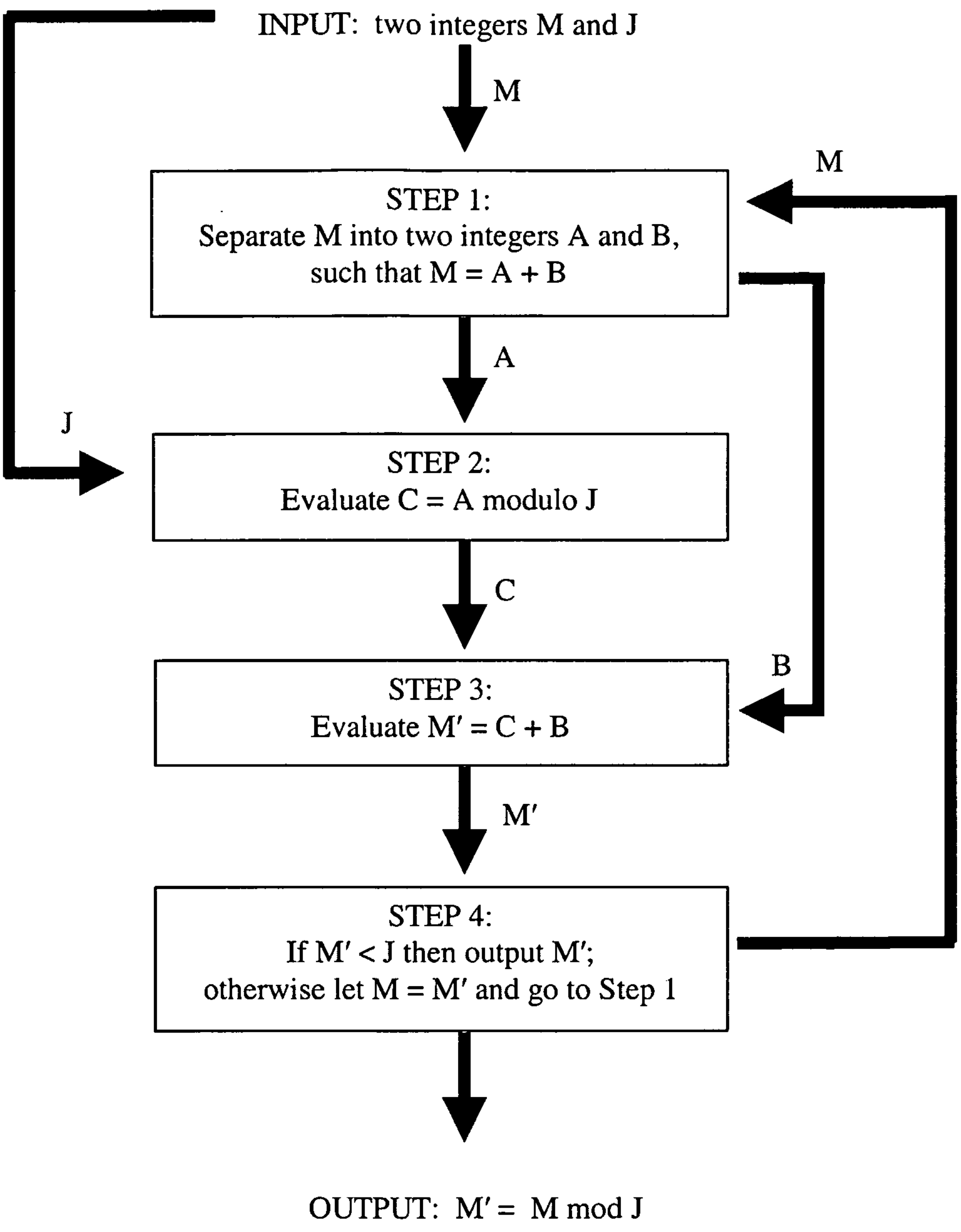
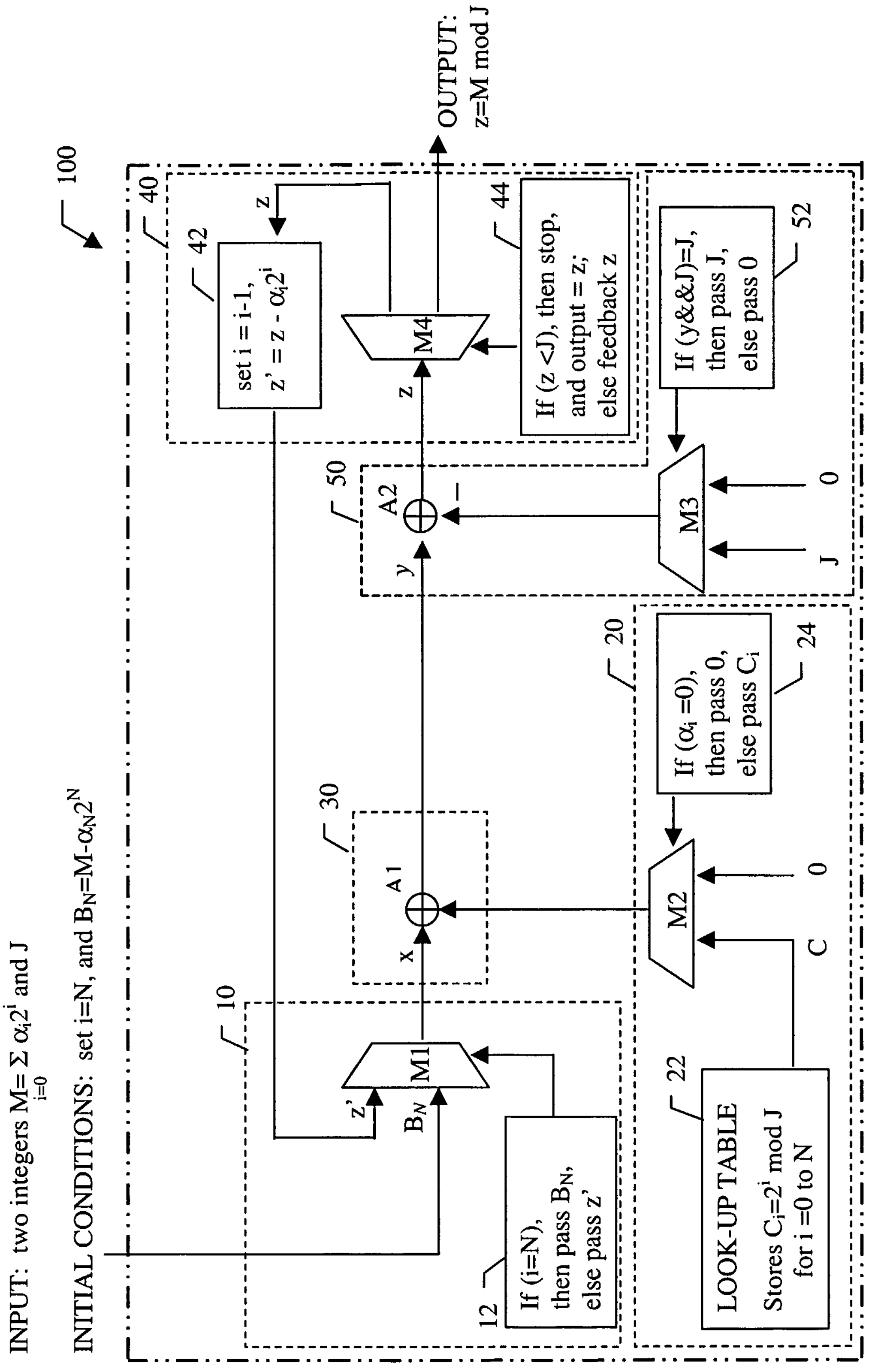

INPUT: two integers M and J
M
M
STEP 1:
Separate M into two integers A and B,
such that M = A + B
A
J
STEP 2:
Evaluate C = A modulo J
C
B
STEP 3:
Evaluate M′ = C + B
M′
STEP 4:
If M′ < J then output M′;
otherwise let M = M′ and go to Step 1
OUTPUT: M′ = M mod J

FIG. 4


INPUT: two integers $M=\sum_{i=0}^{N} \alpha_i 2^i$ and J
INITIAL CONDITIONS: set i=N, and $B_N=M-\alpha_N 2^N$
OUTPUT: z=M mod J
100
10
12
If (i=N), then pass B_N, else pass z'
M1
z'
B_N
x
30
A1
20
22
LOOK-UP TABLE
Stores $C_i=2^i$ mod J for i =0 to N
24
If (α_i =0), then pass 0, else pass C_i
M2
C
0
50
A2
y
–
M3
J
0
52
If (y&&J)=J, then pass J, else pass 0
40
42
set i = i-1, $z' = z - \alpha_i 2^i$
z
M4
44
If (z <J), then stop, and output = z; else feedback z

INTERNATIONAL SEARCH REPORT

Inte ional Application No
PCT/US 02/27958

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F7/72 H03M13/27

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F H03M

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, PAJ, WPI Data, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	PARHAMI B: "Analysis of tabular methods for modular reduction" SIGNALS, SYSTEMS AND COMPUTERS, 1994. 1994 CONFERENCE RECORD OF THE TWENTY-EIGHTH ASILOMAR CONFERENCE ON PACIFIC GROVE, CA, USA 31 OCT.-2 NOV. 1994, LOS ALAMITOS, CA, USA,IEEE COMPUT. SOC, US, 31 October 1994 (1994-10-31), pages 526-530, XP010148556 ISBN: 0-8186-6405-3 page 526, column 1, line 1 -page 530, column 2, last line; figures 1-5 -----	1,10,11

□ Further documents are listed in the continuation of box C.

□ Patent family members are listed in annex.

° Special categories of cited documents :

"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 when the document is taken alone

"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

Date of the actual completion of the international search 4 November 2002	Date of mailing of the international search report 11/11/2002
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Authorized officer Van Staveren, M