BESCHWERDEKAMMERN
DES EUROPÄISCHEN
PATENTAMTS

BOARDS OF APPEAL OF
THE EUROPEAN PATENT
OFFICE

CHAMBRES DE RECOURS
DE L'OFFICE EUROPEEN
DES BREVETS

Internal distribution code:
(A) [ ] Publication in OJ
(B) [ ] To Chairmen and Members
(C) [ ] To Chairmen
(D) [X] No distribution

Datasheet for the decision of 5 June 2008

Case Number:	T 0349/06 - 3.5.02
Application Number:	02704440.3
Publication Number:	1374418
IPC:	H03M 13/27
Language of the proceedings:	EN

Title of invention:
Random-access multi-directional CDMA2000 turbo code interleaver

Applicant:
Qualcomm Incorporated

Opponent:
-

Headword:
-

Relevant legal provisions:
EPC Art. 84

Relevant legal provisions (EPC 1973):
-

Keyword:
"Clarity and support by the description - yes (after amendments)"

Decisions cited:
-

Catchword:
-

Europäisches Patentamt | **European Patent Office** | **Office européen des brevets**

Beschwerdekammern | Boards of Appeal | Chambres de recours

Case Number: T 0349/06 - 3.5.02

DECISION

of the Technical Board of Appeal 3.5.02
of 5 June 2008

Appellant:

QUALCOMM INCORPORATED
5775 Morehouse Drive
San Diego
CA 92121-1714 (US)

Representative:

Dunlop, Hugh Christopher
R.G.C. Jenkings & Co.
26 Caxton Street
London SW1H 0RJ (GB)

Decision under appeal:

Decision of the Examining Division of the European Patent Office posted 27 September 2005 refusing European application No. 02704440.3 pursuant to Article 97(1) EPC.

Composition of the Board:

Chairman:	M. Ruggiu
Members:	M. Rognoni
	H. Preglau

Summary of Facts and Submissions

I. The appellant (applicant) appealed against the decision of the examining division refusing European application No. 02 704 440.3.

II. In the contested decision, the examining division found, *inter alia*, that the claims 1 of all requests were both unclear and not supported by the description within the meaning of Article 84 EPC.

III. Oral proceedings before the Board were held on 5 June 2008.

IV. The appellant requested that the decision under appeal be set aside and that a patent be granted on the basis of claims 1 to 7 according to the request received during the oral proceedings.

V. Claim 1 according to the appellant's request reads as follows:

"A linear congruential sequence turbo code interleaver (360) comprising:

first means (362) for receiving a binary input address comprising 5 LSBs and c MSBs and computing a first interleaved binary address *A* during a first clock cycle in response thereto;

second means (364) for receiving said input address offset by one, and computing a second interleaved binary address *A* during said first clock cycle in response thereto;

third means (374) for determining whether said first interleaved binary address A is invalid by determining whether its corresponding binary number is greater than or equal to the interleaver size N, and generating a signal in response thereto;

wherein said first means (362) and said second means (364) include means for implementing the following expression for determining the interleaved address A:

$$A = bitrev\ (row)\ .\ 2^{c} + \{(col + 1)\ .\ \ c(i)\}\ modC$$

where "row" is a binary number corresponding to the 5 LSBs of the input address, "col" is a binary number corresponding to the remaining c MSBs of the input address, C is equal to 2^{c} and c(i) is the output of a lookup-table where "i" is the current row number corresponding to the value of "row", "row" and "col" thus representing the row address and the column address of a matrix having 2^{5} rows and 2^{c} columns;

fourth means (376) responsive to said signal generated by the third means (374) for selecting the first interleaved address as an output interleaved binary address for said first clock cycle if the third means (374) determines the binary number corresponding to the first interleaved binary address to be smaller than N and thus valid, otherwise, for selecting the second interleaved binary address as the output interleaved address for said first clock cycle; and

means (380) for providing an address offset with respect to said input address, said means (380) comprising an address offset counter (384) for incrementing or decrementing the address offset by 1 in response to said third means (374) determining that said first interleaved address is invalid."

Claims 2 to 6 are dependent on claim 1.

Claim 7 reads as follows:

"A method of linear congruential sequence turbo code interleaving or deinterleaving, the method including:

receiving (362) a binary input address comprising 5 LSBs and c MSBs and computing (362) a first interleaved binary address A during a first clock cycle in response thereto;

receiving (364) said input address offset by one, and computing (364) a second interleaved binary address A during said first clock cycle in response thereto;

determining (374) whether said first interleaved binary address A is invalid by determining whether its corresponding binary number is greater than or equal to the interleaver size N, and generating a signal in response thereto;

wherein said computing (362, 364) said first and second interleaved addresses comprises implementing the expression:

$$A = bitrev\ (row)\ .\ 2^{c} + \{(col + 1)\ .\ \ c(i)\}\ modC$$

where "row" is a binary number corresponding to the 5 LSBs of the input address, "col" is a binary number corresponding to the remaining c MSBs of the input address, C is equal to 2^c and c(*i*) is the output of a lookup-table where "*i*" is the current row number corresponding to the value of "row", "row" and "col" thus representing the row address and the column address of a matrix having 2^5 rows and 2^c columns;

selecting (376) in response to said signal the first interleaved address as an output interleaved binary address for said first clock cycle if the binary number corresponding to the first interleaved binary address is determined to be smaller than N and thus valid, otherwise, selecting (376) the second binary interleaved address as the output interleaved address for said first clock cycle;

providing an address offset with respect to said input address, said address offset being incremented or decremented by one in response to an invalid address signal."

VI. The appellant has essentially argued that the language of claim 1 specified in a clear and complete manner an interleaver as described in the application as originally filed with reference to Figure 5. In particular, the equation recited in the claim defined how to take a linear address and map it into an interleaved address A. As specified in claim 1, the LSBs and the MSBs of the linear address were shifted to the MSBs and LSB of the interleaved address, respectively. This represented the filling by rows and reading by columns that was described in the original application with respect to a notional matrix of the input data sequence. By definition, only the last row of the interleaver matrix could have invalid addresses when the interleaver size N was smaller than the number of elements of the interleaver matrix. As the matrix was read out by columns rather than rows, no consecutive invalid addresses could be generated. In other words, when the first means required by claim 1 generated an invalid address, the second means, which processed the following input address, would by definition generate a valid address.

Claim 7 related to a method comprising steps corresponding to the functions performed by the means of claim 1.

Hence, the claims now on file fulfilled the requirements of Article 84 EPC.

Reasons for the Decision

1. The appeal is admissible.

2.1 The present invention relates to the interleaver 360 shown in Figure 5 of the present application and to a corresponding method of linear interleaving or deinterleaving.

2.2 According to the description (application as published, page 24, lines 32 to 35) "*the interleaver 360 maps a linear address sequence into a permuted address sequence. The permuted addresses are generated in a manner similar to a bit reversal block interleaver except that the addresses in a given row of the block* *are further permuted using a linear congruential sequence (LCS)*."

The mapping of a linear address sequence into a permuted address sequence is explained with reference to a notional "matrix" having R rows and C columns (application as published, page 24, last line to page 25, line 20). The relevant steps can be summarized as follows:

- the matrix is filled with a linear sequence of addresses starting with the top row, filling from left to right (*ibid.* page 25, lines 1 and 2);

- the addresses within each row are shuffled according to a row - specific linear congruential sequence (LCS) (*ibid.* page 25, lines 6 to 7*)*;

- the rows of the matrix are shuffled according to a bit reversal rule applied to the row index (*ibid.* page 25, lines 7 to 9);

- the addresses are read out of the matrix by column starting with the left-most column reading from top to bottom and proceeding to the right (*ibid.* page 25, lines 9 to 11).

The LCS applied to each row i takes the following form:

$$x_i (n) = c(i) * (n+1)$$

when the initial condition on each row, *i.e.* the initial permutation of the first element in row *i,* is given by $x_i(0) = c(i)$.

Thus, in the above expression, x_i represents a permuted column address of a symbol in row *i* having an initial column address n, whereas c(*i*) is an odd constant modulo C, whose value depends on the row *i*.

2.4 As explained by the appellant, the step of reading out the interleaver matrix by columns ensures that there can never be two consecutive invalid addresses in the permuted address sequence.

An invalid address relates to an element of the interleaver matrix which does not correspond to any of the addresses of the data array. The notional interleaver matrix is filled with the linear sequence of addresses starting with the top row and from left to right. If the number of addresses N (the "interleaver size") of the data array is comprised between 2^{k-1} and 2^k, some of the last addresses in the last row of the interleaver matrix 2^k will not correspond to addresses of the linear address sequence to be interleaved and thus will not be valid. In other words, the input linear address sequence having N elements does not fill up an interleaver matrix 2^k if $N < 2^k$.

According to the rules for mapping the input linear address sequence summed up above and exemplified with reference to a notional interleaver matrix referred to in the description, all the initial addresses in a row may be moved to a different row but are never shifted to separate rows. Thus, as the permuted addresses of the interleaver matrix are read out by columns and all invalid addresses are necessarily in the same row, the permuted address sequence cannot have two consecutive invalid addresses.

3.1 Figure 5 shows a block diagram of an interleaver according to the present invention. The rules for mapping a sequence of N elements (0, 1, ..., K-1) into a permuted sequence summarized above are implemented by a first computation unit 362 as follows:

- the sequence of consecutive linear addresses is expressed by a binary number of m + n digits and constitutes the input address of the interleaver;

- m bits, representing the "row address" of a notional "matrix", are sent to a bit reversal block 366 and to a lookup table 368;

- the bit reversal block 366 outputs m bits corresponding to a bit-reversed row address;

- the lookup table 368 outputs an n - bit number which corresponds to the constant c(*i*) ;

- the n bits corresponding to a "column address" are sent to the adder 365, which outputs a column address incremented by 1;

- the n - bit column address incremented by 1 is multiplied by the n - bit constant c(*i*) to generate an n - bit <u>permuted</u> column address;

- the "interleaved address", the linear address input is mapped into, is obtained by combining the n – bit address and the m - bit address so that the corresponding n - bit binary number becomes a "row address" and the m - bit binary number a "column address".

The last step of the above procedure ensures that, after the row and column permutations have been carried out, the elements of the notional matrix which is filled with the input data sequence are read out by columns and not by rows.

The interleaver of Figure 5 comprises also a second computation unit 364, which calculates the "next" interleaver address, *i.e.* the current interleaver address incremented or decremented by one, and a threshold detector 374 which outputs a "Bad Addr" signal and, by means of this signal, controls an interleaver select multiplexer 376. In response to the "Bad Addr" signal, the multiplexer 376 selects the output of the first computation unit 362 or the output of the second computation unit 364 as the output of the interleaver.

3.2 As pointed out in the description (published application, page 26, lines 13 to 16), the computation units 362 and 364 are designed to implement the equation [3] specified on page 25, line 25 and recited in claims 1 and 7.

According to this equation, an address of the linear input sequence defined by c MSBs and 5 LSBs is mapped into an interleaved address A expressed in terms of a "row" address and a "column" address of an element of the notional matrix of the input data.

As explained on page 25 of the description (lines 21 to 34), the 5 LSBs of the input linear address become the "row" address of A, while the remaining c MSBs represent its "column" address. The fact that the 5 LSBs of the input address are used as "row" address of the notional matrix of the initial data array implies that the matrix is filled by rows but read out by columns. Thus, successive linear addresses, which differ by one, are mapped into permuted addresses of the notional interleaver matrix which identify successive rows, as their "row" addresses, *i.e.* the 5 LSBs of successive linear addresses, differ by one.

In other words, the definitions of the 5 LSBs as "row" addresses and of the remaining c MSB as column addresses is equivalent to taking as successive elements of the permuted data sequence elements of the input matrix located in successive row addresses.

4.1 The interleaver specified in claim 1 of the appellant's request comprises the following means shown in the block diagram of Figure 5:

- first means 362 ("first computation unit 362") for receiving a binary input address comprising 5 LSBs and c MSBs and computing a first interleaved binary address A ("the current interleaved address") during a first clock cycle in response thereto;

- second means 364 ("second computation unit 364") for receiving said input address offset by one and computing a second interleaved binary address A ("the next interleaved address") during said first clock cycle in response thereto;

- third means 374 ("threshold detector 374") for determining whether said first interleaved binary address A is invalid by determining whether its corresponding binary number is greater than or equal to the interleaver size N, and generating a signal ("Bad Addr") in response thereto (see application as published, page 26, lines 23 to 25);

- wherein said first means 362 and said second means 364 include means for implementing the expression [3] specified on page 25, lines 22 - 30 of the application;

- fourth means 376 ("multiplexer") responsive to said signal generated by the third means 374 for selecting the first interleaved address as an output interleaved binary address for said first clock cycle if the third means 374 determines the binary number corresponding to the first interleaved binary address to be smaller than N and thus valid, otherwise, for selecting the second interleaved binary address as the output interleaved address for said first clock cycle; and

- means 380 ("address offset circuit") for providing an address offset with respect to said input address, said means 380 comprising an address offset counter 384 for incrementing or decrementing the address offset by one in response to said third means 374 determining that said first interleaved address is invalid (see published application, page 26, line 30 to page 27, line 3 and page 27, lines 19 to 28).

4.2 In summary, claim 1 relates to an interleaver as shown in Figure 5 of the present application and which maps a linear address sequence into a permuted address sequence according to the algorithm [3] specified on page 25, line 25, whereby the implementation of this algorithm corresponds to permuting the data elements of a notional interleaver matrix and reading them out as specified in the description on pages 13 and 14.

4.3 Claim 2 dependent on claim 1 and corresponding to claim 8 of the application as originally filed specifies that the interleaver of claim 1 further includes fifth means which is equivalent to the second multiplexer 390 shown in Figure 5.

Dependent claims 3 to 6 are equivalent to claims 4 to 7 of the application as originally filed.

In the contested decision, no objections under Article 84 EPC were raised against the dependent claims.

5. Claim 7 refers to a method for interleaving and de-interleaving a turbo code which defines in terms of method steps the functions implemented by the means recited in claim 1.

6. In the result the Board considers that the wording of the claims of the appellant's request is clear and concise and that the claims are supported by the description of the application as originally filed. Thus, the appellant's request now satisfies the requirements of Article 84 EPC.

7. In the examination proceedings and in particular in the contested decision, the examining division considered only the requirements of Article 84 EPC and did not examine the claimed invention as to novelty and inventive step.

Under these circumstances, the Board considers it appropriate to make use of its power under Article 111(1) EPC and to remit the case to the department of first instance for further prosecution on the basis of the amended claims 1 to 7.

Order

For the above reasons it is decided that:

1. The decision under appeal is set aside.

2. The case is remitted to the first instance for further prosecution.

The Registrar: The Chairman:

U. Bultmann M. Ruggiu