BESCHWERDEKAMMERN DES EUROPÄISCHEN PATENTAMTS

BOARDS OF APPEAL OF THE EUROPEAN PATENT OFFICE

CHAMBRES DE RECOURS DE L'OFFICE EUROPÉEN 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 25 July 2014

Case Number:	T 2241/10 - 3.5.06
Application Number:	04775953.5
Publication Number:	1623307
IPC:	G06F7/38, G06F7/544
Language of the proceedings:	EN

Title of invention:
PROCESSOR REDUCTION UNIT FOR ACCUMULATION OF MULTIPLE OPERANDS WITH OR WITHOUT SATURATION

Applicant:
QUALCOMM Incorporated

Headword:
Pipelined multioperand adder/QUALCOMM

Relevant legal provisions:
EPC 1973 Art. 56, 84

Keyword:
Claims - clarity - main request (no)
Inventive step - main request (no) - auxiliary request (yes)

Decisions cited:

Catchword:

This datasheet is not part of the Decision.
It can be changed at any time and without notice.

Beschwerdekammern
Boards of Appeal
Chambres de recours

European Patent Office
D-80298 MUNICH
GERMANY
Tel. +49 (0) 89 2399-0
Fax +49 (0) 89 2399-4465

Case Number: T 2241/10 - 3.5.06

DECISION
of Technical Board of Appeal 3.5.06
of 25 July 2014

Appellant: (Applicant)
QUALCOMM Incorporated
5775 Morehouse Drive
San Diego, CA 92121 (US)

Representative:
Beresford, Keith Denis Lewis
Beresford & Co.
16 High Holborn
London
WC1V 6BX (GB)

Decision under appeal:
Decision of the Examining Division of the European Patent Office posted on 15 June 2010 refusing European patent application No. 04775953.5 pursuant to Article 97(2) EPC.

Composition of the Board:

Chairman D. Rees
Members: M. Müller
M.-B. Tardo-Dino

Summary of Facts and Submissions

I. The appeal lies against the decision of the examining division, with reasons dispatched on 15 June 2010, to refuse the application no. 04775953.5. In the decision, in particular the following documents were cited:

D1: Schulte M. J. *et al.*, "Parallel Saturating Multi-operand Adders". Proc. of the Int. Conf. on Compilers, Architectures and Synthesis for Embedded Systems, pp. 172-179, Nov. 2000, and
D6: Ungerer T. *et al.*, "A Survey of Processors with Explicit Multithreading", ACM Computing Surveys, vol. 35, no. 1, pp. 29-63, March 2003,

and it was argued that the then claim 1 violated Article 123 (2) EPC and, insofar as it did not, lacked an inventive step, Article 56 EPC 1973, over D1 in view of common knowledge as illustrated by further prior art documents, amongst which D6.

II. The applicant filed a notice of appeal on 25 August 2010 and paid the required fee. A statement setting out the grounds of appeal was received on 25 October 2010. The appellant requested that the decision be set aside and that a patent be granted based on claims 1-18 according to a main or an auxiliary request as filed with the grounds of appeal, apparently combined with the following application documents:

description, pages	
2, 4-7, 10, 11, 13-14	as published
1, 8, 9, 12	received with letter of 29 September 2008
3, 3a-3c	received with letter of 6 July 2009
drawings, sheets	
1-5	as published

III. With a summons to oral proceedings, the board informed the appellant of its preliminary opinion according to which the independent claims of both requests lacked an inventive step, Article 56 EPC 1973. Clarity objections were also raised, Article 84 EPC 1973.

IV. In response to the summons, the appellant filed amended claims 1-16 according to a main and a first auxiliary request, and new claims according to second to fifth auxiliary requests, along with arguments in favour of inventive step.

V. Oral proceedings were held as scheduled on 25 July 2014. During the oral proceedings, the appellant filed claims 1-16 of a further amended first auxiliary request to replace the first auxiliary request on file. Second to fifth auxiliary requests were maintained.

VI. Claim 1 according to the the main request reads as follows:

"A multithreaded processor comprising a plurality of arithmetic units (104-1 to 104-m) and an accumulator unit, the processor comprising:

a reduction unit (102, 102') for coupling between m arithmetic units included in the plurality of arithmetic units and the accumulator unit, the reduction unit being configured to receive input operands (P[1] to P[m]) from the m arithmetic units and a first accumulator value (P[0]) from the accumulator unit, the input operands associated with a thread of the multithreaded processor, the reduction unit further comprising m inputs for receiving the input operands, m adders and an m stage pipeline, the m stage pipeline reducing the worst case delay of the reduction unit;

wherein the reduction unit is operative to sum the input operands and the first accumulator value, and to generate a second accumulator value for delivery to the accumulator unit; and

wherein the reduction unit is controllable to support saturation and wrap-around arithmetic;

characterised in that the reduction unit includes one or more pipeline registers (204), such that each of the m inputs is coupled to a corresponding adder by means of N-1 pipeline registers, where N is a stage number greater than or equal to 1; and

wherein multiplications and reductions for one dot product are configured to execute concurrently with operations from other threads, and wherein the number of cycles between execution of instructions from a given thread is greater than or equal to the number of pipeline stages (m) in the reduction unit plus cycles needed to write and read from the accumulator unit."

Claim 1 of the first auxiliary request reads as follows. The differences over claim 1 of the main request are highlighted by the board:

"A multithreaded processor comprising a plurality of arithmetic unity (104-1 to 104-m) and an accumulator unit, the processor comprising:

a reduction unit (102, 102') for coupling between m arithmetic units included in the plurality of arithmetic units and the accumulator unit, the reduction unit being configured to receive input operands (P[1] to P[m]) from the m arithmetic units and a first accumulator value (P[0]) from a source location of the accumulator unit specified in an instruction, the input operands associated with a thread of the multithreaded processor, the reduction unit further comprising m inputs for receiving the input operands, m adders and an m stage pipeline~~, the m stage pipeline reducing the worst case delay of the reduction unit~~;

wherein the reduction unit is operative to sum the input operands and the first accumulator value, and to generate a second accumulator value for delivery to a destination location of the accumulator unit specified in an instruction; and

wherein the reduction unit is controllable to support saturation and wrap-around arithmetic;

characterised in that the reduction unit includes one or more pipeline registers (204), such that each of the m inputs is coupled to a corresponding adder by means of N-1 pipeline registers, where N is a stage number greater than or equal to 1; ~~and~~

wherein multiplications of a dot product are performed by the arithmetic units; and

wherein multiplications and reductions for one dot product are configured to execute concurrently with operations from other threads;~~, and wherein the number of cycles between execution of instructions from a given thread is greater than or equal to the number of pipeline stages (m) in the reduction unit plus cycles needed to write and read from the accumulator unit.~~

wherein the accumulator unit is configured to store accumulator values for dot products of different threads."

The wording of the claims of the further auxiliary requests is not relevant for the present decision.

VII. At the end of the oral proceedings the chairman announced the decision of the board.

Reasons for the Decision

The invention

1. The application is concerned with circuitry for speeding up the evaluation of dot products, *i.e.*, given two k-element vectors X[1]...X[k] and Y[1]...Y[k], the calculation of X[1]*Y[1]+...+X[k]*Y[k].

1.1 At the center of the proposed circuitry is a so-called "reduction unit" to calculate the sum of m+1 input operands: m of these, P[1] to P[m], are the primary operands, while the additional one, P[0], is meant to be connected to an accumulator unit in which the intermediate result of a prior summation has been stored. To calculate the dot product of two k-element vectors, k multiplication results X[i]*Y[i] must be added. This addition is carried out in groups of size m, the intermediate result being stored in the accumulator unit. In total, essentially k/m such iterations are needed.

1.2 In an architecture for the calculation of the dot product (as depicted in the original application in fig. 1), m multiplications in each iteration can be carried out in parallel and in a pipelined sequence with the summation of m+1 operands, each of which requiring m binary additions.

1.3 In order to speed up this summation, the invention proposes to set up the "reduction unit" itself as a pipeline (depicted in fig. 2). This pipeline uses registers, referred to as "pipeline registers", to delay the operands according to their respective position in the sum: Specifically, the two operands P[0] and P[1] for the first addition are not delayed but the operands

P[i], for any i between 2 and m, are delayed by i-1 delay registers (and thus i-1 cycles).

1.4 On its own, this pipeline does not speed up any summation performed by the reduction unit but rather, as the application notes, increases its latency, *i.e.* the required number of cycles for each summation "by roughly a factor of m" (see p. 8, lines 10-13). However, a multithreaded processor can exploit the pipeline by interleaving the calculation of multiple dot products. If the pipeline has a length of m, then at least m threads are required to "hid[e]" the mentioned "increase in cycle count" (see p. 8, lines 25-31).

1.5 To deal with multiple threads, the accumulator provides a suitable number of separate accumulator registers (see fig. 4, no 106'). In the instructions controlling the reduction unit, the accumulator source and destination registers are separately specified (see p. 12, lines 7-14 w.r.t. fig. 5).

1.6 The reduction unit can be controlled to support "saturation arithmetic" or "wrap-around arithmetic", two well-known alternative ways of handling overflow of the additions (see p. 1, line 21 - p. 2, line 8). Saturation arithmetic is required by the GSM standard for the calculation of dot products (see p. 2, last par.), but wrap-around arithmetic may be required elsewhere.

Clarity, Article 84 EPC

2. Claim 1 of the main request specifies that "the m stage pipeline reduc[es] the worst case delay of the reduction unit". The expression "worst case delay" is unclear. It does not appear to be a term of the art. The board would be inclined to suppose that it indicated

the time between the arrival of the (first) input at a unit and the time of production of the result. However firstly this would appear to be the same as the "latency", which is identified separately, and secondly if it were the latency, the statement would be incorrect. The latency of a pipelined serial reduction unit is precisely the same as that of a serial reduction unit where all the inputs are presented at the same time, and indeed m times greater than that of an ideal parallel unit executing its function in one cycle, as noted in the description (see above, 1.4). The appellant could not give any further explanation in the oral proceedings. Thus the board simply does not know what is being claimed, and judges that the skilled person would be in the same situation. Therefore the claim is unclear, Article 84 EPC 1973.

3. Claim 1 of the main request refers, on the one hand, to unspecified "arithmetic units" coupled to the reduction unit and, on the other hand, to "multiplications and reductions for ... dot product[s]". The claim language leaves unspecified the relation between the arithmetic units and the multiplications by failing to specify that, as is apparently intended, the multiplications are to be performed by the "arithmetic units", Article 84 EPC 1973.

4. Claim 1 of the main request specifies a minimal "number of cycles between execution of instructions from a given thread". As it stands, this language does not specify a feature of the claimed multithreaded processor but a feature of its use and it is unclear whether or in what way this feature actually limits the claimed matter, Article 84 EPC 1973.

5. Claim 1 of the auxiliary request does not suffer from these clarity problems: It is explicitly claimed that the multiplications are performed by the arithmetic units and both the claimed reduction of a "worst case delay" and the minimal number of cycles between instructions of the same thread are deleted.

5.1 The board has no occasion to raise other clarity problems of its own motion and is thus satisfied that the claims of to the first auxiliary request are clear.

Article 123 (2) EPC

6. The decision under appeal (reasons 4) objected under Article 123 (2) EPC against then claim 1 requiring that multiplications and reductions execute "concurrently using different threads" whereas the description (page 8, line 27) rather disclosed an execution "concurrently with operations from other threads". Since claim 1 according to both present requests now uses the wording from the description, this objection is now moot.

7. The amendments made to claim 1 of the first auxiliary request conform with the requirements of Article 123 (2) EPC:

7.1 The fact that the arithmetic units perform multiplications is disclosed, *inter alia*, on p. 6, 20-23, of the original application.

7.2 That multiplications are performed during the calculation of dot products is disclosed throughout the application. The deletion from claim 1 of the effect that the reduced worst case delay is reduced does not constitute added matter because the effect was not originally claimed, because it did not limit claim 1 since and to the extent to which it merely expressed what was meant to be achieved by the features of the claimed multithreaded processor, and because the claimed effect is, on the board's best guess at an interpretation, in conflict with the explicit disclosure of the original application (p. 8, lines 10-13).

7.3 The deletion of the minimal number of cycles between instructions from the same thread does not add matter because this constraint was not originally claimed either. Moreover, the board notes that this temporal spacing is required by the data dependencies between two partial m-element additions of the same k-element addition on a given thread, and thus is an essentially implicit consequence of the claimed use of the pipeline for the calculation of dot products (see also point 14.3 below).

7.4 The fact that accumulator values are received from and delivered to locations in an accumulator unit which are specified in an instruction was originally disclosed in figure 4 (no. 106'), figure 5 (fields ACCS and ACCD), and the description on page 12, lines 7-14).

8. The board is thus satisfied that the claims of the first auxiliary request conform with Article 123 (2) EPC.

The prior art

9. D1 was co-authored by some of the present inventors and relates to the same general problem as the present invention. It relates in general to what is called a MAC (multiply-accumulate) unit which may, in particular, be used to calculate the dot products for a GSM speech coder (see secs. 1 and 2).

9.1 The general architecture of such a MAC-unit most pertinent for the present invention is depicted in figure 2. A number of arithmetic units ("saturating MACs") are provided to calculate, for instance, the pointwise multiplication of a number of vector elements (p. 173, left col., equations (1) and (2)). The results of these calculations are sent, via a set of "pipeline registers", to an "(m+1)-input saturating multioperand adder" SMA, *i.e.* a "reduction unit". The result of the SMA is fed back into one of the pipeline registers, acting as an "accumulator", and thus made available for a subsequent operation. A p-element dot product can thus be calculated in essentially p/m cycles (p. 173, right col., 3rd par.). The SMA is also equipped to perform saturating or wrap-around arithmetic depending on an operation control signal OP_{sma} (see p. 173, left col., lines 2-5 from the bottom).

9.2 D1 is specifically concerned with speeding up individual multioperand additions by calculating, in parallel, several partial additions and combing them selectively depending on the occurring overflows. One design of such a parallel SMA is depicted in figure 10.

9.3 For assessing the speed-up achieved by the parallel SMAs, a design for a serial SMA is also presented (see figure 11). It is disclosed that the parallel SMA is indeed significantly faster than the serial one but also significantly larger (p. 178, left col. 3rd par. and table 3).

9.4 It is also remarked that "[a]lthough serial and parallel SMAs can be pipelined to reduce their worst case delay, this increases their latency and decreases their throughput for saturating dot products". By way of example, it is disclosed that a 2-stage pipeline would cause the calculation of a p-element dot product to require essentially 2*(p/m) cycles (p. 178, left col. last par.).

Starting point for inventive step assessment

10. The board considers the serial SMA according to D1 as a suitable starting point for the assessment of inventive step of the present invention and the remark that a "serial SMA can be pipelined" as a motivation for the skilled person to consider what such a pipelined serial SMA could look like.

10.1 The appellant challenges this position by arguing as follows:

1) "[I]t cannot reasonably be regarded as obvious for the skilled person [in order to] arrive at a solution to a technical problem to select as a starting point a design explicitly identified as disadvantageous" (letter of 25 June 2014, p. 8, 6th par.).

2) "[T]he mere reference to the possibility of pipelining the SMA [in] D1 ... cannot be regarded as a clear disclosure of the desirability of pipelining the internal operations of the SMA" (see p. 7, 2nd par.).

3) The objective technical problem to be considered is "how to provide a processor having a more efficient and economical reduction unit to provide the desired dual functionality of handling both saturating and wrap-around arithmetic" (p. 7, 3rd and 4th par.).

10.2 The board disagrees.

1) While D1 discloses that the serial SMA is inferior to the parallel SMA in terms of speed, it is advantageous in terms of size and thus cost. The board considers that this is sufficient motivation for the skilled person to consider whether alternative ways of speeding up the calculation of dot products exist based on the simpler, smaller and cheaper serial SMA. Moreover, the board is of the opinion that the remark in D1 that pipelining the SMA increases its latency by a factor corresponding to the number of pipeline stages would not discourage the skilled person from considering a pipelined serial SMA: In the board's view, the skilled person would be aware that pipelining commonly involves a trade-off between increased latency and the advantage of interleaved, "pipelined" computations.

2) The reference in D1 to the pipelining of an SMA clearly refers to the internal workings of the SMA, *i.e.* the saturating multioperand adder, rather than, as the appellant has argued, the structure of the SMAC unit (see fig. 2) as a whole which itself shows a 2-stage pipeline structure.

3) The objective technical problem is derived from the difference between the prior art and the claimed invention. Since the claimed feature that the reduction unit - the SMA - be "controllable to support saturation and wrap-around arithmetic" is already known from D1, the objective technical problem cannot relate to how to achieve the effect of this feature. Moreover, in the board's understanding the provided choice between saturating

and wrap-around arithmetic does not affect in a non-trivial manner the design of a pipelined SMA, nor did the appellant argue that and why such an interaction existed.

Inventive step, Article 56 EPC 1973

Main request

11. Notwithstanding these clarity problems, the board deems it appropriate in this case to give an inventive step assessment for claim 1 of the main request as well.

12. As argued above, the board considers the serial SMA according to D1 as the most suitable starting point for an assessment of inventive step.

13. The invention according to claim 1 of the main request differs from the serial SMA according to D1 by the following three features.

i) The reduction unit contains an m stage pipeline which the serial SMA does not.

ii) The arithmetic units and the reduction unit are comprised in a multithreaded processor which is not mentioned in D1.

iii) The number of cycles between execution of instructions from a given thread is limited from below as claimed.

The appellant's argument that there is an additional difference, namely that "[t]he reduction unit is controllable to support saturation and wrap-around arithmetic" (see appellant's letter of 25 June 2014, p. 5, 4th par. from the bottom, and p. 6, point iv) cannot be followed in view of the operation control signal OP_{sma} known from D1 (p. 173, left col. lines 2-5 from the bottom and figs. 1 and 2).

14. D1 mentions that "serial ... SMAs can be pipelined" (p. 178, left col., last par.). All further details about this pipeline are left open except for the suggestion that a pipelined SMA has a latency increased by a factor corresponding to the number of pipeline stages.

14.1 The board considers this as an explicit prompt for the skilled person to consider how a serial SMA could be pipelined, with a view to compensating its increased latency. Furthermore, the board considers this to be the objective technical problem solved by the subject-matter of claim 1 in view of D1 due to the differences i)-iii).

14.2 *Re. difference i)* In the board's view, the skilled person would realize that the serial SMA depicted in figure 11 shows an almost pipelined setup already, the only obstacle being that all four input lines P1 to P4, in particular P3 and P4, are blocked until the third addition has completed. In order to turn this serial SMA into a pipeline it would be sufficient to free the input lines P1-P4 for new input data at every cycle. Since, however, the arguments P3 and P4 are not consumed before one respectively two more cycles, the skilled person would find it obvious to introduce pipeline "delay" registers as claimed to produce a pipelined serial SMA.

14.3 *Re. difference iii)* The skilled person would further note that the use of the accumulator/feedback operand as the first argument to the SMA (see D1, fig. 2) constrains the way in which this pipeline can be filled.

Specifically, two partial sums of m elements in a k-element addition cannot be fed into the pipeline in subsequent cycles because the first argument in the second sum is the outcome of the first sum which is only available after at least m cycles. Without further changing the SMA the pipeline just constructed would thus have to remain idle during precisely the number of instructions claimed (namely "the number of pipeline stages (m) in the reduction unit plus cycles need to write to and read from the accumulator unit") if it were to operate only on one summation task. This is an issue of data dependencies, well known in pipeline architectures.

14.4 *Re. difference ii)* The skilled person would thus naturally be led to filling the pipeline with other, independent computational tasks. The board is of the opinion that this is sufficient motivation for the skilled person to consider multiple threads, given that it is common knowledge that pipelines may be used efficiently in combination with multi-threading (see D6, sec. 3), and to interleave the calculation of dot products from different threads.

14.5 The board thus comes to the conclusion that the skilled person starting from D1 would arrive at the claimed invention in an obvious way from the mere suggestion to produce a pipelined version of the serial SMA according to D1. The board also disagrees with the appellant's allegation that the above construction "relies upon the skilled person producing a series of useless and ineffective intermediate modifications of the prior art" (letter of 25 June 2014, p. 10, 6th par.) since each of the above steps is clearly motivated.

14.6 The subject matter of claim 1 according to the main request thus lacks an inventive step over D1 in view of common knowledge on pipelines and multi-threading, Article 56 EPC 1973.

First auxiliary request

15. Apart from the above-mentioned clarifications, claim 1 of the first auxiliary request differs from claim 1 of the main request by requiring the accumulator unit

a) to be "configured to store accumulator values for dot products of different threads" and

b) to comprise a number of locations which can be specified as "source" and "destination locations" in an instruction controlling the reduction unit.

15.1 Based on the board's above finding that calculations of independent sums on different threads are to be interleaved with each other it is a matter of mathematical necessity that different accumulator/feedback values must be made available to the different calculations and thus, in fact, to the different threads.

15.2 D1 itself however contains no hint to the skilled person as to how this requirement should be put into practice, due to the fact that D1 does not mention multi-threading at all. Also D6 does not disclose or suggest any preferred structure of the required accumulator unit, let alone the one specifically claimed.

15.3 The appellant argued in oral proceedings that an alternative to the claimed accumulator structure would be a queue of accumulator registers into which, at one end, a new partial sum would be entered and from which, at the other end, any required partial sum would be retrieved. If the SMA pipeline had a latency of, say, n cycles and, therefore, two subsequent partial summation instructions from the same thread had to be separated by n cycles, it would be natural to provide a queue of n accumulator registers and switch between up to n threads. The appellant argued that the sequence of pipeline registers employed in the pipelined SMA would prompt the skilled person to consider, by analogy, a similar sequence of accumulator registers to handle the accumulator values from different threads. In contrast, the claimed block of independently accessible accumulator registers would not be obvious for the skilled person.

15.4 The board accepts that the solution proposed by the appellant would solve the problem of how to handle the accumulator values from different threads. At least this establishes that the claimed structure of the accumulator unit is not the only possible one. The board also notes that the accumulator queue and the claimed accumulator unit have different advantages and disadvantages: The former allows for simpler instructions since the ends of the queue need not be specified in the instructions but can be left implicit, while the latter makes it simpler to handle a variable number of threads. The board also considers that *a priori* it must be assumed that still further solutions for the handling of accumulator values from different threads exist.

15.5 Given that neither D1 nor D6 disclose or suggest the claimed structure of the accumulator unit and that the claimed structure is neither the only one nor necessarily the one which would occur to the skilled person, the board comes to the conclusion that this structure cannot be considered obvious in view of D1 and D6.

15.6 Therefore, the subject-matter of claim 1 according to the first auxiliary request establishes an inventive step of claim 1 over the prior art to hand, Article 56 EPC 1973.

Further requests

16. The claims according to the first auxiliary request being allowable, the further auxiliary requests need not be considered.

Order

For these reasons it is decided that:

1. The decision under appeal is set aside.

2. The case is remitted to the department of first instance with the order to grant a patent on the basis of claims 1-16 of the first auxiliary request as filed during the oral proceedings, description and drawings to be adapted if needed.

The Registrar: The Chairman:

Beschwerdekammern des Europäischen Patentamts ✦ Boards of Appeal of the European Patent Office ✦ Chambres de recours de l'Office européen des brevets

B. Atienza Vivancos D. Rees

Decision electronically authenticated