

Internal distribution code:

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

**Datasheet for the decision
of 8 January 2018**

Case Number: T 0932/12 - 3.5.06

Application Number: 09738384.8

Publication Number: 2283419

IPC: G06F9/38

Language of the proceedings: EN

Title of invention:

SYSTEM FOR PROVIDING TRACE DATA IN A DATA PROCESSOR HAVING A
PIPELINED ARCHITECTURE

Applicant:

Imagination Technologies Limited

Headword:

Trace data/IMAGINATION TECHNOLOGIES

Relevant legal provisions:

EPC Art. 56

Keyword:

Inventive step - (no)

Decisions cited:

Catchword:



Beschwerdekammern
Boards of Appeal
Chambres de recours

Boards of Appeal of the
European Patent Office
Richard-Reitzner-Allee 8
85540 Haar
GERMANY
Tel. +49 (0)89 2399-0
Fax +49 (0)89 2399-4465

Case Number: T 0932/12 - 3.5.06

D E C I S I O N
of Technical Board of Appeal 3.5.06
of 8 January 2018

Appellant: Imagination Technologies Limited
(Applicant) Home Park Estate
Kings Langley Hertfordshire WD4 8LX (GB)

Representative: Slingsby Partners LLP
1 Kingsway
London WC2B 6AN (GB)

Decision under appeal: Decision of the Examining Division of the
European Patent Office posted on 8 December 2011
refusing European patent application No.
09738384.8 pursuant to Article 97(2) EPC.

Composition of the Board:

Chairman W. Sekretaruk
Members: A. Teale
S. Krischer

Summary of Facts and Submissions

I. This is an appeal against the decision, dispatched with reasons on 8 December 2011, to refuse European patent application No. 09 738 384.8 on the basis that the subject-matter of claims 1 and 3 lacked inventive step, Article 56 EPC, in view of the following document:

D1: US 6 615 370 B1.

II. A notice of appeal was received on 16 January 2012, and the appeal fee was paid on 25 January 2012.

III. A statement of grounds of appeal was received on 2 April 2012 in which the applicant requested that the application be granted on the basis of the currently pending claims.

IV. The application is thus being considered in the following form:

Description: pages 1 to 26, as originally filed.

Claims: 1 to 4, received on 26 September 2011.

Drawings: pages 1 to 4, as originally filed.

V. In an annex to a summons to oral proceedings the board gave its provisional opinion that the subject-matter of *inter alia* claim 1 lacked inventive step, Article 56 EPC, in view of D1.

VI. In a letter received on 24 November 2017 the appellant withdrew its request for oral proceedings and requested a decision based on the written proceedings. The board then cancelled the oral proceedings.

VII. Claim 1 reads as follows:

"A data processor comprising: an instruction scheduler (7) for scheduling instructions to be executed by the data processor; and a trace unit coupled to the instruction scheduler for receiving trace data relating to scheduled instructions, the trace unit having a trace output for connection with an external device, a trace output buffer (21) connected to the trace output for storing trace data prior to passing it to the trace output, characterised by a trace impact counter (20) connected to the trace output buffer (21) and to the scheduler (7), wherein, in use, the trace output buffer (21) sends buffer occupancy data to the scheduler (7), the scheduler pauses the scheduling of instructions based on the buffer occupancy data and sends an indication of the pause of scheduling to the trace impact counter (20), and the trace impact counter (20) adds the number of cycles for which the scheduling of instructions has been paused to trace data sent to the trace output buffer (21)."

Reasons for the Decision

1. The admissibility of the appeal

In view of the facts set out at I to III above, the appeal fulfils the admissibility criteria under the EPC and is consequently admissible.

2. Summary of the invention

2.1 The application relates to pipelined data processors. In general, the execution of an instruction by a processor involves the steps of "fetch instruction from memory", "decode", "execute instruction in Arithmetic

Logic Unit (ALU)" and "write result to register/memory". As each step only uses part of the processor's resources, the serial execution of instructions results in a relatively low instruction throughput. As illustrated in figure 1, a processor with a pipelined architecture achieves a higher throughput by using an "instruction scheduler" (7), connected between the instruction fetch unit (2) and a plurality of execution pipelines (8,9,10; see page 10, lines 29 to 35), to organise the execution of several instructions in parallel in separate execution pipelines, each comprising a plurality of execution stages.

2.2 Such pipelines yield performance benefits when no branches in the program logic occur. When they do occur, and the state of the branch conditions (which typically rely on other pipelines) are not yet known, execution can continue speculatively on the assumption that a branch condition either was or was not met. Should it however later transpire that the assumption was wrong, then work in progress in the pipeline has to be discarded, reducing performance. If, on the other hand, it transpires that all branch conditions were correctly assumed, then the results of the instruction are stored and the instruction is said to have been "completed".

2.3 It is known to monitor the progress of the stream of instructions passing through each pipeline using a "trace unit" which "exports" the reported "trace data" via a peripheral interface to external equipment. As the instruction execution rate of the processor can sometimes exceed the data rate of the interface used to export trace data, an output buffer is connected between them, termed the "trace output buffer" (21). The buffer sends "buffer occupancy data" (see figure 1)

(either a "buffer full" flag or an indication of the available room in the buffer; see page 17, lines 2 to 5) to the instruction scheduler (7). Hence, if the buffer fills up, or becomes "close to full capacity" (see page 16, lines 22 to 25), it can cause the instruction scheduler to cease sending instructions to the various pipelines. A "trace impact counter" (20), connected to the instruction scheduler and the trace output buffer, counts the number of instruction cycles which elapse while the scheduler is not issuing instructions; see page 16, lines 31 to 32.

3. Document D1

3.1 As illustrated in figure 1, D1 relates to a "system-on-chip" (SOC) device (101) comprising a processor (102) connected via a communication link (104) to a debug circuit (103) which buffers trace information from the processor and transmits it to an external system (106). The FIFO (first-in-first-out) buffer (see figure 2; Fifo 202) in the debug circuit can be configured to implement "Tracing with processor stalling"; see column 2, line 59, to column 3, line 32, and column 6, lines 59 to 63. In this mode the external system receives trace information at a slower rate than it is produced by the processor, at least until the buffer fills up, or is within a predefined number of storage locations of being full, when the processor is halted until space is available in the buffer again; see the flow chart in figure 4; step 409.

3.2 As shown in figure 2, trace data (221) from the processor (102) passes via the capture buffer (203) to the trace processor (205). In the trace processor, trace message generation logic extracts state information from the capture buffer and loads

corresponding trace messages into FIFO buffer 202. The trace processor also receives a pre-scaled (216) version of the processor clock (218) to provide a "time increment" time stamp for trace messages; see column 8, lines 34 to 39.

- 3.3 According to column 7, lines 12 to 21, whilst other systems filter the trace information in the external system, the system according to D1 seeks to optimise the use of the off-chip communication bandwidth by means of techniques such as on-chip trace message compression. For instance, time information is compressed to give time difference information which can later be decompressed to recover the time information; see column 4, lines 31 to 46. Thus the trace processor may determine a time difference since the last trace message was generated as a number of time increments and encode the time difference as a time stamp in a trace message; see column 4, lines 57 to 60, and column 8, lines 46 to 49.
- 3.4 Figure 7 and table 1 (see columns 12 and 13) illustrate the field structure of a trace message. Of particular interest are the "over stall" (705) and "timestamp" (708) fields. The time stamp field can specify a number of timer increments since a last trace message was generated as a byte; see column 12, lines 29 to 35. The "over stall" field comprises one bit which, when the FIFO is operated in "stall mode", is set when the processor was "stalled for some indeterminate time prior to this trace message being generated because there was no space available in the Debug Module FIFO."

4. Inventive step, Article 56 EPC

4.1 The appealed decision

4.1.1 According to the reasons for the decision, it was implicit in D1 that processor 102 (see figure 2) comprised an instruction scheduler. The debug circuit 103 (see figure 2) was regarded as a trace unit in the sense of the claims having a trace output for connection with an external device; see transmission circuit 215 and output 106. The debug circuit also comprised a trace output buffer for storing trace data prior to passing it to the trace output; see figure 2, capture buffer 203 and FIFO 202. The trace processor 205 was regarded as a trace impact counter. Column 8, lines 46 to 49, stated that trace processor 205 determined a time difference from the last trace message generated and compressed time stamp information (from reference counter 217) by encoding the time difference as time stamp information in a trace message. If the FIFO 202 was configured in "stall mode" the capture buffer (203) stalled the processor (102) until space was available in the FIFO. This was seen as disclosing the claimed sending of buffer occupancy data by the trace output buffer to the instruction scheduler. The scheduler sent an indication of the pause in scheduling to the trace output counter; see the "stalled" signal from the processor 102 to the capture buffer 203. In view of column 8, lines 46 to 49, which stated that the trace processor 205 determined a time difference from the last trace message generated and encoded the time difference as time stamp information in a trace message, the trace processor 205 as "trace impact counter" added an indication of the number of cycles for which instruction scheduling was paused to trace data sent to

capture buffer 203 and FIFO 202 as the claimed "trace output buffer". When the FIFO 202 was configured in trace hold mode (see column 7, lines 56 to 66) the processor was stalled when the FIFO was full or nearly full. This change of system state created a trace message, referred to in the decision as "TM1". The next trace message generated by the trace processor 205 was referred to in the decision as "TM2". As explained in column 12, lines 16 to 19, and table 1 in column 13, in "stall mode" the "overstall" bit of a trace message, in this case TM2, was set if the processor 102 was previously stalled due to a lack of space in the FIFO 202. TM2 could also contain a time stamp field indicating a number of timer increments since the last trace message was generated and thus an indication of the number of cycles that the processor was stalled while waiting for space to appear in buffer FIFO 202.

4.1.2 The subject-matter of claim 1 differed from the disclosure of D1 in that D1 did not specify when the second trace message TM2 occurred. It could occur when the processor resumed execution, meaning that the time stamp would indicate the exact number of cycles during the pause, or it could occur later. The skilled person would have recognised that generating TM2 later led to inaccuracies regarding the analysis of trace data and would thus have chosen, as a matter of usual design, to produce TM2 as soon as the processor resumed execution, thus arriving at the subject-matter of claim 1 in an obvious manner.

4.2 The grounds of appeal

The appellant has argued that the decision was based on an incorrect test for inventive step and has disputed

the analysis and possible modifications of the prior art set out in the decision.

4.3 The board's view on inventive step

4.3.1 The board accepts the appellant's argument that it is neither implicit nor obvious from D1 that a stall would generate a trace message, since, as the appellant says, the purpose of trace messages is to reveal the status of the processor rather than that of the FIFO buffer 202. Moreover it would be counterproductive to signal a buffer being full by adding a further message to the same buffer. The decision cites column 8, lines 12 to 27, as evidence that the occurrence of particular system states results in the creation of a trace message, but this passage does not mention processor stalls and merely states that the trace processor 205 extracts state information details from the capture buffer 203 and loads corresponding trace messages into FIFO 202. The decision also cites column 7, lines 63 to 66, relating to how the debug circuit reacts to the FIFO buffer filling up, but this passage also does not mention a processor stall triggering a trace message. Hence the board finds that the scenario set out in the decision (see points 13.4 and 13.5) of trace messages TM1 and TM2 relating, respectively, to a processor stall and a subsequent event once instruction execution has resumed, is not disclosed in D1.

4.3.2 The board takes the view that, in terms of claim 1, D1 discloses a data processor (101) comprising an instruction scheduler (implicit in processor 102) for scheduling instructions to be executed by the data processor, a trace unit (103) coupled to the instruction scheduler for receiving trace data relating to scheduled instructions, the trace unit having a

trace output for connection with an external device (106), and a trace output buffer (202) connected to the trace output for storing trace data prior to passing it to the trace output. The processor also comprises a trace impact counter (implicit in column 7, lines 63 to 66) connected to the trace output buffer (202) and to the scheduler (102), wherein, in use, the trace output buffer (202) sends buffer occupancy data to the scheduler (102), the scheduler pauses the scheduling of instructions based on the buffer occupancy data and sends an indication (see figure 2; "Stalled" 220) of the pause in scheduling to the trace impact counter.

- 4.3.3 Hence the subject-matter of claim 1 differs from the disclosure of D1 in that the trace impact counter adds the number of cycles for which the scheduling of instructions has been paused to trace data sent to the trace output buffer.
- 4.3.4 The appellant has argued that the technical problem solved by the invention is to provide more useful information in the trace data. Given the breadth of the term "useful" and the lack of an indication of a technical use of the information, the board is not persuaded that a technical problem is solved or that the difference features in claims 1, set out above, have a technical effect and thus can contribute to inventive step.
- 4.3.5 The board consequently finds that the subject-matter of claim 1 does not involve an inventive step, Article 56 EPC, in view of D1.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar:

The Chairman:



B. Atienza Vivancos

W. Sekretaruk

Decision electronically authenticated