

Internal distribution code:

- (A) [] Publication in OJ
(B) [] To Chairmen and Members
(C) [X] To Chairmen

D E C I S I O N
of 13 June 1995

Case Number: T 0877/93 - 3.5.1

Application Number: 87112494.7

Publication Number: 0257650

IPC: G06F 9/30

Language of the proceedings: EN

Title of invention:
Microprocessor

Applicant:
HITACHI, LTD., et al

Opponent:
-

Headword:
Microprocessor/HITACHI

Relevant legal provisions:
EPC Art. 56

Keyword:
"Inventive step (yes) "

Decisions cited:
-

Catchword:
-



Case Number: T 0877/93 - 3.5.1

D E C I S I O N
of the Technical Board of Appeal 3.5.1
of 13 June 1995

Appellant: HITACHI, LTD.
6, Kanda Surugadai 4-chome
Chiyoda-Ku
Tokyo 101 (JP)

Representative: Strehl Schübel-Hopf Groening & Partner
Maximilianstrasse 54
D-80538 München (DE)

Decision under appeal: Decision of the Examining Division of the European
Patent Office dated 7 May 1993 refusing European
patent application No. 87 112 494.7 pursuant to
Article 97(1) EPC.

Composition of the Board:

Chairman: P. K. J. van den Berg
Members: R. Randes
G. Davies

Summary of Facts and Submissions

I. The appeal contests the Examining Division's decision of 7 May 1993 to refuse the European patent application No. 87 112 494.7 (publication No. EP-A-0 257 650), filed on 27 August 1987 and claiming priority of 27 August 1986.

II. The reason given for the refusal was that the subject-matter of independent Claims 1 and 7 filed on 21 December 1992 lacked an inventive step, having regard to the prior art document

D1: US-A-4 293 907.

III. An appeal was lodged against the decision on 28 June 1993. The appeal fee was paid on the same day.

On 17 September 1993, the Appellant filed a Statement of Grounds, together with a new set of claims. In response to a communication pursuant to Article 11(2) of the Rules of Procedure of the Boards of Appeal, which cited D1 and

D2: IBM Technical Disclosure Bulletin Vol. 24, No. 7B, December 1981, pages 3737 - 3738; M.M.BHANSALI et al.: "Decimal instruction set for IBM Series/1 processors",

the Appellant filed, on 24 April 1995, a further new set of Claims 1 to 7, as well as amendments to the description.

In the oral proceedings held on 13 June 1995, the Appellant further amended the claims and description in response to comments from the Board.

IV. The independent claims of the current request read:

"1. A microprocessor comprising

instruction memory means (3, 7, 8) for storing in sequential order instructions of a predetermined program to be performed,

control means (2) coupled to said instruction memory means for controlling the microprocessor in accordance with said stored instructions, and

instruction executing means (6) coupled to said control means for performing operations in accordance with said stored instructions, said instruction executing means including information holding means (R5) and an arithmetic-and-logic-unit (ALU),

characterised in that

said instruction executing means (6) is coupled to data memory means to load data specifying a kind of operation including an ALU operation to be performed into said information holding means (R5) in response to a first instruction, said data being stored and changeable in said data memory means, and

that said arithmetic-and-logic unit (ALU) is coupled to said information holding means to perform directly, without decoding, said included ALU operation specified exclusively by said data stored in said information holding means (R5) in response to a second instruction.

7. A method for controlling a process to indicate a kind of operation to be performed in a microprocessor, which comprises

instruction memory means (3, 7, 8) for storing in sequential order instructions of a predetermined program to be performed,

control means (2) coupled to said instruction memory means for controlling the microprocessor in accordance with said stored instructions, and

instruction executing means (6) coupled to said control means for performing operations in accordance with said stored instructions, said instruction executing means including information holding means (R5) and an arithmetic-and-logic-unit (ALU), characterised by the steps of loading data specifying a kind of operation including an ALU operation to be performed from data memory means into said information holding means (R5) in response to a first instruction, said data being stored and changeable in data memory means, and causing said arithmetic-and-logic-unit (ALU) to perform directly, without decoding, said included ALU operation specified exclusively by said data stored in said information holding means (R5) in response to a second instruction."

V. The Appellant argued essentially as follows:

In the invention, the register R5 held the complete "kind of operation", i.e. the arithmetical or logical operation to be carried out. In contrast, the only element stored in the OER register in D1 which could be said to influence the kind of operation was a mere optional modifier specifying the length of the operands to be operated on. The system proposed in the application was more flexible, in that the complete ALU operation could be specified dynamically, by changing the contents of R5. Further, in D1, the extra length-determining part from the OER had to be combined with the kind of ALU instruction specified in the instruction register to determine the precise operation of the ALU, so that a further decoding step of the combined instruction was required. In the microprocessor of the application, since the complete ALU operation was specified in the register, that operation could be applied directly to the ALU without another decoding

step. This was very important for speed of operation, especially where the ALU operation was to be carried out many times on a sequence of data. The feature of applying the ALU operation directly, without decoding, from the register to the ALU resulted from the feature of storing the whole ALU operation in the register. Therefore these features should be considered in combination when determining the question of inventive step.

In D1, the only reference to the storage of values for the OER register indicated that these values were stored in ROM, i.e. a fixed memory. It was clearly specified in the patent that the values were stored in a changeable memory. This was a point at least of novelty over D1, even if (as was conceded in the oral proceedings) the substitution of RAM for ROM in D1 might not in itself be inventive. The invention in the current case lay in a combination of factors which worked together.

In response to the citation, by the Board, of D2, it was remarked that this document, as well as not showing the storage of the kind of operation separate from the instruction register, also showed an ALU operation specification which was clearly coded.

VI. The Appellant therefore requested that the decision of the Examining Division be set aside and that a patent be granted on the basis of

Claims 1 to 7 submitted in the oral proceedings;

description pages 5 to 10, 13, 16 to 24, 26 to 31, 33, 34 and 36 as originally filed; pages 1 to 4, 11, 12, 14, 15 and 32 filed 24 April 1995; and pages 2a, 25 and 35 submitted at the oral proceedings;

drawing sheets 1 to 7 as originally filed.

VII. At the end of the oral proceedings the Board announced its decision to grant the Appellant's request.

Reasons for the Decision

1. The appeal is admissible.
2. The Board is satisfied that the amendments submitted meet the requirements of Articles 123(2) and 84 EPC, in that they do not go beyond the content of the application as filed, and in that the claims are sufficiently clear and are supported by the description.
3. In no prior art document available to the Board is the operation of the arithmetic and logic unit (ALU) in the execution of an instruction determined exclusively by a value stored in an information holding means by a previous instruction. Hence the claimed subject-matter is novel.
- 4.1 D1 is the only document available to the Board in which the nature of the processing carried out in the execution of an instruction is significantly affected by a value previously taken from memory and stored in a register. It appears therefore to be the only appropriate starting point for consideration of inventive step.

This document discloses instruction memory means as defined in the current claims, including ROM 402 (Figure 4) and an instruction register IR. It further shows control means ("controller" 408) and instruction executing means, coupled to the control means, including

information holding means ("op-code extension register", OER 421), and an arithmetic-and-logic unit ALU (410, 411, etc.). Thus all the features of the pre-characterising part of the current independent claims are disclosed. In addition, the instruction executing means (the elements controlled by controller 408) are coupled via the data bus to data memory means ("RAM", etc., Figure 4), as well as to the instruction memory. One function of the instruction executing means is to load data specifying elements of the behaviour of following instructions into the information holding means OER. In particular, a bit may be loaded into the OER to specify the length of the operands which the ALU uses in subsequent arithmetic and logical operations (column 9 lines 4 to 8). Finally, the ALU of D1 is clearly coupled to the information holding means OER (via controller 408) to influence operations carried out in the ALU in response to following instructions insofar as they specify ALU operations.

4.2 D1 explicitly only mentions storing values to be loaded into the OER in ROM (column 11 lines 39 to 42), but it would appear to be self-evident to the skilled person, in consideration of basic principles of computer design, that these values might also be stored to and loaded from RAM, i.e. in a "changeable" form, as specified in the current claims.

Thus there are two significant differences between the disclosure of D1 and the invention as now claimed. Firstly, in the application the complete ALU operation specifier is stored in the information holding means. Secondly, this ALU operation specifier is applied directly from the information holding means to the ALU, without further decoding.

4.3 The problem which is said to be overcome by the system disclosed in D1 is to keep instruction lengths short, the major reason for this being to save memory and processing time (D1, column 1, lines 48 to 62). It is crucial to the solution adopted in D1 that the instruction modifier held in the OER will normally apply to every ALU instruction within a long sequence of code, so that "LOAD OER" will not appear too frequently within a program. There is no hint in D1 that the whole ALU operation specifier should be taken from the OER. This would in fact run counter to the goal of D1, in that a LOAD OER instruction would have to be executed before every change of ALU operation. Considering the instruction set envisaged in D1 (see D1 Table 1), such a change would normally be very frequent, as each instruction executes at most one ALU operation. Thus loading the entire ALU operation to be executed into the OER register would actually defeat the object of D1, since it would mean that very many LOAD OER instructions would have to be included in the code, to the detriment of both memory occupied and processing speed. Hence not only is there no indication in D1 that the entire ALU operation specifier should be loaded into the information holding means, but indeed the designer of a system addressing the problem of D1 could be expected to reject this idea if it arose.

The current application starts from the consideration of a more complex instruction set than that contemplated in D1, one in which the execution of certain instructions, such as an inter-bit field instruction, may cause the same ALU operation to be executed many times in succession (see page 3 lines 12 to 20 and Figure 7 of the application as published). The object of the invention is to improve the flexibility of use of such instructions in that the same program sequence (as e.g. in Table 3, page 6) can be run with a different ALU

operation each time it is executed, the particular ALU operation depending on program-determined data and thus changeable by, for instance, user input. To this end, the ALU operation is not encoded in the particular instruction (the inter-bit field operation BVMAP) itself, but rather loaded into a register from a memory location. As long as the application of the invention is confined to such multiple-ALU-operation instructions, the negative consequence of loading the entire ALU operation specifier into a register which would be seen to result if this were done in the context of D1 does not apply. Thus because of the different instruction sets contemplated, a step which would degrade the system of D1 overcomes a problem in the context of the application.

Thus the invention as now claimed is not obvious in the light of D1 alone.

- 4.4 After the amendments made to the claims, the Board accepts that D2 is not of relevance, since there is no teaching which the skilled person would derive from D2 and which would lead him to adapt the system of D1 to store the whole ALU operation specifier in the OER. Neither does it seem that any other prior art document available to the Board would lead the skilled person to take this step. Hence it is to be concluded that the subject-matter of the current independent claims satisfies the requirements of Articles 52(1) and 56 EPC for an inventive step.

Order

For these reasons it is decided that:

1. The decision under appeal is set aside.
2. The case is remitted to the first instance with the order to grant a patent on the basis of the Appellant's request.

The Registrar:

The Chairman:

M. Kiehl

P. K. J. van den Berg

