# EXHIBIT J

EXPEDITED PROCEDURE - AF
PATENT
RM-1

5

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
## BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

| | |
|---|---|
| In re application of ) | |
| ) | |
| Blomgren et al. ) | Examiner: V. Vu |
| ) | |
| Serial No. 08/179,926 ) | Group Art Unit: 2315 |
| ) | |
| Filed: 1/11/94 ) | |
| ) | |
| For: Dual-Instruction-Set Architecture CPU ) | |
| with Hidden Software Emulation Mode ) | |
| ) | |
| ) | |

10

### REPLY BRIEF ON APPEAL UNDER 37 C.F.R. § 1.192

15 Hon. Commissioner of Patents and Trademarks
Box AF
Washington, DC 20231

Sir:

20

This is in reply to the Examiner's Answer of 11/9/95.

Several new points of argument were raised in the Examiner's Answer, and two new

references were cited as background skill level.

'PORTNOVA IS NOT A SIMULATOR'

In the response section of the Examiner's answer, page 6, last 2 paragraphs[1], Examiner asserts as a new point of argument[2] that "Portanova is NOT a simulator":

> "Appellants allege that Portanova does not teach the claimed invention because Portanova is a simulator and does not execute CISC instructions.
> The examiner disagrees. The fact that Portanova's system can execute CISC instructions, whether by software emulation or hardware emulation, and produce real results clearly shows that Portanova is NOT a simulator."

10   However, Portanova states "The designer then writes the CISC instruction **emulator** using RISC instructions, as described in the example above." (col. 30, lines 55-57, emphasis added) Portanova over and over again uses various forms of the word "emulate". Webster's New World Compact School and Office Dictionary (1989) defines "emulate" as "to imitate" while "simulate" is defined as "to look or act like",

15   "to feign". Appellant finds very little difference in meaning between "emulator" and "simulator". Portanova himself calls his own system an "emulator". It is absurd to state that Portanova is an "emulator" but not a "simulator".

The fact that "real results" are produced by a simulator, emulator, or computer is

20   irrelevant. What "real results" does Portanova's system, or any other computer system produce ? Binary Numbers. A person could write a BASIC computer program and produce the same binary numbers that Portanova's emulator, or DEC's VAX produces. Producing 'real results' does not mean that Portanova is NOT a simulator or emulator.

25   The fact that Portanova is an emulator drives to the heart of this appeal: the complete, total absence of prior art showing both RISC and CISC hardware execution. Some computers, such as the 'prior art' CISC systems cited in Portanova, have hardware which executes ONLY CISC instructions. Other computers execute ONLY RISC instructions. Portanova is an example of a RISC computer that can 'emulate' CISC

30   instructions by first translating them to RISC instructions. Portanova is a RISC

---

[1] Locations are approximate since no line numbers were provided on the Examiner's Answer

computer. <u>Portanova</u> ONLY executes RISC instructions. Some of these RISC instructions may have been *translated* from CISC instructions, but they are still only RISC instructions. <u>PORTANOVA</u> EXECUTES ONLY RISC INSTRUCTIONS.

5    The Examiner uses the term "hardware emulation' in a way not recognized by artisans. Hardware emulation refers to IC design tools such as hardware accelerators, made by IKOS and Quickturn, for simulating complex logic on a hardware accelerator which is normally simulated by software. Indeed, "hardware emulation" is a contradiction: emulation is something which is done by software, not by hardware. When hardware

10   exists, there is no emulation. Thus "hardware" is the antithesis of "emulation". Combining these two words together as the Examiner does is improper and deceptive as the term "hardware emulation' does not appear in the cited prior art.

Examiner believes that since <u>Portanova</u> discusses Prior-Art CISC computers, <u>Portanova</u>

15   is suggesting that his 'preferred' embodiments[3] be modified to directly execute in hardware CISC instructions on his RISC processor. However <u>Portanova</u> emulates CISC instructions as clearly disclosed in his preferred embodiments. Of course, all of <u>Portanova</u>'s discussion about writing software emulation routines of RISC instructions is then irrelevant if the Examiner's modification is made. Why hire a bunch of software

20   programmers to write emulation code when you can just execute those CISC instructions ?

Another new point of argument[4] was raised in the first paragraph of page 7 of the Examiner's answer:

25       "Appellants also allege, in the identification of points of disagreement, that appellants fail to find
         teaching in Portanova which *suggests hardware execution of CISC instructions.*"

This refers to page 8, lines 19-24 of the appeal brief:

---

[2] Examiner has not before asserted that Portanova is not a Simulator. The argument that real results somehow define a simulator is also new.
[3] The argument that Portanova's *preferred embodiments* do not contain the suggestion appears new.
[4] This partial quotation from the Appeal Brief is used for a new argument.

Examiner believes that Portanova *suggests* hardware execution of CISC instructions in an otherwise RISC processor. This suggestion appears somewhere in col 29-30. Appellant has been unable to find this suggestion and has requested that the Examiner specifically point out what he is relying on.

5    The appellant's brief has been partially quoted in a misleading way. Portanova does

suggest hardware execution of CISC instructions in a CISC processor. Portanova does

not suggest hardware execution of CISC instructions in an otherwise RISC processor.

Certainly old prior-art CISC computer suggest and indeed are examples of hardware

10   execution of CISC instructions. But these old CISC computers do not execute those

CISC instructions in an otherwise RISC processor !

Indeed, Examiner has never pointed out a specific line in Portanova which suggests that

Portanova's software emulator be replaced with CISC hardware execution in

15   Portanova's RISC computer. Portanova does suggest that his software emulator could

replace these prior-art CISC architectures. Indeed, Portanova claims that he can

emulate in software on his RISC computer any number of CISC architectures, such as

Z8000, 68000, VAX, and System/370[5].

The design method disclosed herein applies to any number of CISC instruction sets including
20   MIL-STD-1750, VAX, NEBULA, etc. The approach is to first build a single-level control
(hardwired) using RISC design philosophy. In so doing, the designer attempts to maximize
execution of the RISC (hardwired) instruction set. (col 30, lines 48-54 emphasis added)

Of course, Portanova does not have 4 CISC instruction decoders as well as his RISC

25   instruction decoder. Portanova does not even have 1 CISC instruction decoder. But

Portanova can write 4 different CISC emulation routines and run any of these 4

software emulators on his single RISC hardware. Since the 4 CISC emulators translate

all CISC instructions into RISC instructions, only one RISC decoder is needed. Adding

4 CISC hardware decoders is not 'speculation' any more than adding 1 CISC hardware

30   decoder is 'speculation'. Examiner is correct that it is untrue that Portanova teaches 4

CISC decoders on his RISC computer. It is also untrue that Portanova teaches 1 CISC

decoder on his RISC computer.

---

[5] The argument that it is speculation that 4 CISC decoders are needed is new.

'THIS IS NOT THE ONLY WAY'

Another new point[6] is raised at the bottom of page 8 and top of page 9 of the
Examiner's Answer.

5
> Portanova specifically chooses the design that employs software emulation by using native RISC
> instructions to execute CISC instructions as set forth in his third embodiment to meet his design
> goal. However, this is NOT the ONLY way. Portanova clearly suggests other alternatives design
> approaches to implement CISC architecture known in the prior art (see figures 9-13).

Portanova discloses 2 ways to process CISC instructions:

10    1.    Use a Prior-Art CISC computer (col 29-30, Figures 9-13).

      2.    Use his RISC computer with a software program that emulates CISC
instructions by replacing them with RISC instructions.


Portanova does NOT disclose a third way:

15    3.    Add CISC hardware to a RISC computer and execute either CISC or RISC
instructions.


Appellant's specification discloses this third approach. Examiner has used improper
hindsight to choose the third way from appellant's disclosure over the other two ways
20    fairly disclosed by the prior art.


Examiner has brought out further points of argument to dispute the clear statements in
Portanova that teach away. First it is now asserted that appellants do not properly
construe[7] Portanova (Examiner's answer, page 9).

25
> Appellants appear to suggest that since the proposed modification contradicts Portanova's third
> embodiment, such modification could not be made.

Such contradictions are evidence that the reference teaches away from the proposed
modification.

30

---

[6] The argument that Portanova's teaching is not the only way is new.
[7] The argument that appellants improperly construe Portanova and do not consider every word and figure
is new.

When a proposed modification destroys the intended purpose of the reference, that modification is in error and cannot be made. The intended purpose of faster design time is destroyed by adding CISC hardware to Portanova's RISC computer. The software routines to emulate CISC instructions decrease design time compared to hardware CISC

5   execution because it is faster to write software than it is to design hardware. Thus a modification to add CISC hardware increases design time and destroys the purpose of Portanova. The modification cannot be fairly made.

It is also insinuated that appellants considered only the preferred embodiments and not

10  "every single word and figure of the reference." Appellants have asked the Examiner to point out what 'word' contains the suggestion to use CISC hardware execution on a RISC processor. The Examiner has not found such words in the reference.

It is finally asserted that appellants have not considered the level of skill in the art. To

15  bolster this argument, two new references were cited[8]. The newly-cited IBM disclosure again shows CISC instructions being 'decoded and translated' to RISC instructions. The RISC instructions are then decoded and executed. The new Iwata reference shows microcode programs selected based on 'architecture modes'. However, there is still NO TEACHING WHATSOEVER in cited prior art of CISC hardware execution on a RISC

20  computer. The level of skill in the art for combining RISC and CISC hardware is ZERO. This is an entirely new area.

Why can Portanova emulate any of the Prior-Art CISC architectures ?
Why can Portanova cut his Design Time ?

25

The answer is that Portanova emulates in software CISC instructions. Portanova never executes in hardware CISC instructions on a RISC processor. That is what Portanova fairly teaches.
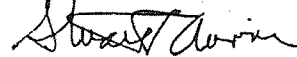
---

[8] These references are new.

For the foregoing reasons, Appellant submits that the rejection of claims 1-20 is in error and should be reversed on appeal.

5

Stuart T. Auvinen
429 26th Avenue
Santa Cruz, CA 95062

(408) 476-5506
(408) 477-0703 Fax

Respectfully Submitted,

Stuart T. Auvinen
Agent for Appellant
Reg. No. 36,435