# EXHIBIT D

#5/a
9/8
9-21-9


I hereby certify that this paper is being deposited with the United States Postal Service with sufficient postage as First Class Mail under 37 CFR 1.8 on the date indicated below and is addressed to: "Commissioner of Patents and Trademarks, Washington, DC 20231."

Signed _____
Stuart T. Auvinen, Reg. No. 36,435

Date Signed and Mailed 9/20/94

PATENT
RM-1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

| | |
|---|---|
| In re application of | |
| Blomgren | 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 | |

AMENDMENT UNDER 37 C.F.R. § 1.111

Hon. Commissioner of Patents and Trademarks
Washington, DC 20231

Sir:

In response to the office action mailed 6/27/94, please amend the above-identified application as follows:

**In the specification:**

On page 2, line 8, before "BACKGROUND OF THE INVENTION" please insert

--BACKGROUND OF THE INVENTION -- RELATED APPLICATIONS

This application is related to co-pending application for a "Pipeline with Temporal Re-Arrangement of Functional Units for Dual-Instruction-Set CPU", filed 1/11/94, U.S.

Serial No. 08/180,023, now Patent No. 5,542,059. This application is further related to copending application for "Emulating Operating System Calls in an Alternate Instruction Set Using a Modified Code Segment Descriptor", filed 7/20/94, U.S. Serial No. 08/277,905, now Patent No. 5,481,684. This application is further related to copending application for "Shared Register Architecture for a Dual-Instruction-Set CPU" ", filed 7/20/94, U.S. Serial No. 08/277,962, now Patent No. 5,481,693. These related applications have a common inventor and are assigned to the same assignee as the present application. --

**In the claims:**

Please amend claims 1-20 as follows:

1. (amended) A central processing unit (CPU) for processing instructions from two separate instruction sets, [the] said CPU comprising:

first instruction decode means for decoding instructions from a first instruction set, [the] said first instruction set having a first encoding of instructions;

second instruction decode means for decoding instructions from a second instruction set, [the] said second instruction set having a second encoding of instructions, [the] said first encoding of instructions independent from [the] said second encoding of instructions;

select means, coupled to [the] said first instruction decode means and [the] said second instruction decode means, for selecting [the] said decoded instruction from either [the] said first instruction decode means or from [the] said second instruction decode means; and

execute means for executing decoded instructions selected by [the] said select means,

whereby instructions from both [the] said first instruction set and [the] said second instruction set are executed by [the] said CPU.

2. (amended) The CPU of claim 1 further comprising:

an instruction fetch buffer, containing instructions to be decoded, coupled to [the] said first instruction decode means and [the] said second instruction decode means; and

instruction pointer means, coupled to [the] said instruction fetch buffer, for indicating [the] an address of [the] a next instruction to be decoded.

3. (amended) The CPU of claim 1 further comprising:

mode register means, coupled to [the] said select means, for indicating [the] an instruction set to be decoded and executed.

4. (amended) The CPU of claim 3 further comprising:

mode control means, coupled to [the] said mode register means, for changing [the] said instruction set to be decoded.

5. (amended) The CPU of claim 4 wherein

the second instruction decode means decodes only a portion of [the] said second instruction set, and [the] said second instruction decode means indicating to [the] said mode control means when an instruction is not in [the] said decoded portion of [the] said second instruction set;

the mode control means changing [the] said instruction set to be decoded to [the] said first instruction set when an indication is received that an instruction is not in [the] said decoded portion of [the] said second instruction set.

6. (amended) The CPU of claim 5 further comprising

a translation-lookaside buffer (TLB) coupled to [the] said execute means, [the] said TLB having address translation entries for translating a virtual address from [the] said execute [unit] means to a physical address for accessing a main memory, [the] said TLB providing an indication to [the] said mode control means to change [the] said instruction set to be decoded to [the] said first instruction set when no translation is

found in [the] said TLB corresponding to [the] said virtual address from [the] said execute [unit] means.

7. (amended) The CPU of claim 6 wherein a handler routine comprised of instructions from [the] said first instruction set is fetched from main memory and executed when mode control is signaled by [the] said TLB or by [the] said second instruction decode means.

8. (amended) The CPU of claim 7 wherein [the] said execute unit provides an indication to [the] said mode control means when an exception occurs in [the] said execute unit, [the] said mode control means changing [the] said instruction set to be decoded to [the] said first instruction set when [the] said indication is received.

9. (amended) The CPU of claim 6 wherein all references to main memory generated by instructions in [the] said second instruction set are translated by [the] said TLB.

10. (amended) The CPU of claim 6 wherein [the] said address translation entries in [the] said TLB are loaded only by instructions decoded by [the] said first instruction decode means.

11. (amended) The CPU of claim 10 wherein [the] said first instruction decode means decodes instructions from [the] said first instruction set and extended instructions added to [the] said first instruction set, and wherein [the] said address translation entries in [the] said TLB are modified only by [the] said extended instructions.

12. (amended) The CPU of claim 11 wherein [the] said first instruction decode means is selected to decode instructions immediately following a reset of [the] said CPU.

13. (amended) The CPU of claim 11 wherein [the] said extended instructions are decoded by [the] said first instruction decode means only when [the] said mode control means is signaled to change [the] said instruction set to be decoded or immediately following a reset.

14. (amended) A method for processing instructions from two separate instruction sets on a central processing unit (CPU), [the] said method comprising:

decoding instructions from a first instruction set with a first instruction decoder, [the] said first instruction set having a first encoding of instructions;

decoding instructions from a second instruction set with a second instruction decoder, [the] said second instruction set having a second encoding of instructions, [the] said first encoding of instructions independent from [the] said second encoding of instructions;

selecting [the] said decoded instruction from either [the] said first instruction decoder or from [the] said second instruction decoder; and

executing [the] said decoded instruction that was selected,

whereby instructions from both [the] said first instruction set and [the] said second instruction set are executed by [the] said CPU.

15. (amended) A method for processing instructions from a complex instruction set computer (CISC) instruction set on a reduced instruction set computer (RISC) Central Processing Unit (CPU), [the] said method comprising:

attempting to decode an instruction with a CISC instruction decode unit that does not decode all instructions in [the] said CISC instruction set;

directly executing [the] said instruction in an execute unit if [the] said CISC instruction decode unit is able to decode [the] said instruction;

entering an emulation mode if [the] said CISC instruction decode unit is not able to fully decode [the] said instruction, indicating that [the] said execute unit cannot directly execute [the] said instruction;

disabling [the] said CISC instruction decode unit and enabling a RISC instruction decode unit when entering emulation mode;

loading an instruction pointer with an address of a software emulation routine for emulating [the] said undecodable instruction, [the] said routine comprising instructions from a separate RISC instruction set;

decoding RISC instructions with [the] said RISC instruction decode unit as [the] said software routine is executed;

executing [the] said RISC instructions in [the] said execute unit; and

exiting emulation mode, disabling [the] said RISC instruction decode unit and enabling [the] said CISC instruction decode unit when [the] said end of [the] said software emulation routine is reached,

whereby all instructions from [the] said CISC instruction set are executed, either directly by [the] said execute unit or by emulation with a software emulation routine comprised of RISC instructions.

16. (amended) The method of claim 15 wherein

the software emulation routine is comprised of RISC instructions and extended instructions, [the] said extended instructions using undefined opcodes in [the] said RISC instruction set;

the method further comprising decoding and executing extended instructions while [the] said software emulation routine is being executed.

17. (amended) The method of claim 16 further comprising:

translating memory references generated by [the] said CISC instructions that are directly executed, [the] said translation of memory references controlled by a software translator routine comprised of RISC instructions and extended instructions, [the] said translator routine loading [the] said resulting translations into a translation-lookaside buffer.

18. (amended) A microprocessor for executing instructions belonging to a reduced instruction set computer (RISC) instruction set and for executing instructions belonging to a complex instruction set computer (CISC) instruction set, [the] said microprocessor comprising:

RISC instruction decode means, for decoding instructions belonging to [the] said RISC instruction set;

CISC instruction decode means, for decoding instructions belonging to [the] said CISC instruction set;

mode register means for indicating a current operating mode of [the] said microprocessor;

enable means, coupled to [the] said RISC instruction decode means and [the] said CISC instruction decode means, for enabling [the] said decoding of instructions belonging to [the] said RISC instruction set or belonging to [the] said CISC instruction set, [the] said enable means responsive to [the] said current operating mode of [the] said microprocessor; and

an execution unit, coupled to [the] said first instruction decode means and [the] said second instruction decode means, for executing instructions belonging to [the] said first instruction set and instructions belonging to [the] said second instruction set,

whereby instructions from [the] said RISC instruction set and instructions from [the] said CISC instruction set can be executed by [the] said execution unit.

19. (amended) The microprocessor of claim 18 wherein [the] said mode register means indicates CISC mode, RISC mode, or an emulation mode, wherein a portion of [the] said CISC instruction set is decoded by [the] said CISC instruction decode means when [the] said mode register means indicates CISC mode, and wherein undecoded CISC instructions are emulated by emulation mode.

20. (amended) The microprocessor of claim 19 wherein emulation mode is entered when [the] said CISC instruction decode means signals an undecoded instruction, [the]

a2 said mode register means changing from CISC mode to emulation mode when an undecoded instruction is signaled.

**REMARKS**

Claims 2-13 were rejected under 35 USC § 112 as being indefinite. Under 35 USC § 102(b), claims 1-4 were rejected as anticipated by Ueda et al (US Pat. No. 4,821,187). Claims 1-20 were also rejected under 35 USC § 103 as obvious over de Nicolas (US Pat. No. 5,167,023). All claims were thus rejected.

Claims 1-20 have been amended to use the article "said" in place of "the" as requested by the Examiner. Claims 2, 3, and 6 were amended to fix the antecedent problems noted by the Examiner. Applicant submits that with the discussion below that claims 1-20 are allowable over the cited references. Reexamination and reconsideration of the claims, as amended, is hereby requested.

The specification has been amended to indicate cross-references to related co-pending applications. Related co-pending application serial no. 08/180,023 was not earlier mentioned because it was filed on the same date as the present application.

***Summary of Independent Claims 1, 14, 15, 18***

Claim 1 recites a first and a second instruction decoder for decoding instructions from a first and a second instruction set. A select means selects either the decoded instruction from the first decoder or from the second decoder. An execute means executes the decoded instruction selected by the select means. Thus the execute means can execute both first and second instructions provided by the select means.

Independent claim 14 is directed to a method for processing instructions from two separate instruction sets. Independent claim 15 is directed to a method for processing instructions from a CISC and a RISC instruction set in which all CISC instructions are

4c

executable, either directly by the execute unit or by emulation with RISC instructions. Independent claim 18 is directed to a microprocessor for executing RISC and CISC instructions using both RISC and CISC instruction decoders and an enable means to enable one of the instruction decoders.

***35 USC § 112 Rejection***

Claims 2, 3, and 6 have been amended to overcome the 35 USC § 112 rejection by providing a proper antecedent. Applicant submits that the claims are now clear and definite, overcoming the 35 USC § 112 rejections to the claims.

PRIOR ART REJECTIONS - 102(B) IN VIEW OF UEDA

Claims 1-4 were rejected as anticipated by Ueda et al (US Pat. No. 4,821,187). Ueda is cited as a system capable of executing two different instruction sets. The system comprises two separate decoders for decoding first and second set of instructions respectively, execution units for executing the decoded first and second instructions, control units for controlling switching execution of first and second instruction sets. Applicant respectfully disagrees and with the following argument overcomes the rejection.

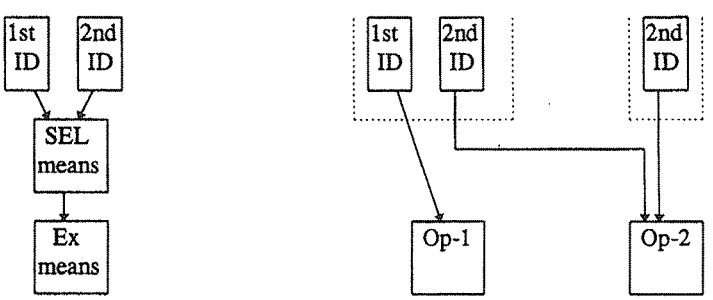
Ueda teaches a parallel processor for simultaneously executing two programs. In the Multi-program mode, "high-speed processing is attained by parallel run of the two independent programs." (col 2 line 13) The present invention does not require two independent programs, nor does it require parallel processing. Ueda is from a different field and attempts to solve a different problem than the present invention, which solves the problem of executing instructions from two different instruction sets.

Ueda teaches two instruction decoders for two different microinstructions. The first microinstruction controls the first operation unit, while the second microinstruction controls the second operation unit. It is not clear from Ueda if these microinstructions

are generated from two different instruction sets, or are merely generated in response to programs written in a single instruction set.

***Structural Differences with Ueda***

As Ueda is directed to a different field, parallel processing, it is not surprising that significant structural differences exist between Ueda and the present invention. The diagram below compares the structure of claim 1 (on the left) to Ueda (on the right). The diagram for Ueda below is simplified from his Figures 1 and 7. The select means recited in claim 1 selects either the decoded instruction from the first instruction decode means or from the second instruction decode means. Ueda neither teaches nor suggests the select means. Instead, Ueda sends first microinstructions to the first operation unit, and second microinstructions to the second operation unit. Ueda also neither teaches nor suggests execute means that can execute decoded instructions from the select means, as recited by claim 1. Since the select means recited in claim 1 can select either a first or a second decoded instruction, the recited execute means of claim 1 can execute either a first or a second decoded instruction.




Present Invention Ueda Reference

Ueda's first operation unit can only execute first microinstructions, while Ueda's second operation unit can only execute second microinstructions. Ueda thus neither teaches nor suggests an execute means that can execute both first and second

instructions. Ueda also neither teaches nor suggests the select means recited in claim 1, as Ueda has no need for a select means.

Ueda's first control means, shown in the top center of the diagram, decodes both first and second microinstructions. Ueda's second control means, on the top far right, decodes only second microinstructions. Ueda's first control means decodes a compound micro-instruction word that can have fields for both first and second microinstructions, as shown in his Figure 2A. The second operation field is sent to the second operation unit, while the first operation field is sent to the first operation unit. When multi-program mode is used, the second control means decodes a microinstruction as in Ueda's Figure 3A, sending the control field to the second execution unit. In multi-program mode, Ueda looks like two independent processors. In contrast, the present invention funnels both first and second instructions to the same execute means.

***Dependent Claims 2-4***

Dependent claims 2, 3, 4 depend upon claim 1, which is allowable for the above-stated reasons. Thus claims 2, 3, 4 are also allowable for the above-stated reasons. Claim 2 recites an instruction fetch buffer that is coupled to both the first and the second instruction decoders. Ueda teaches two separate program memories (col 3 line 5), supplying two decoders (see his Figure 1, elements 3, 4).

Claim 3 recites a mode register means that indicates to the select means the instruction set to be decoded and executed. Claim 4 recites a mode control means for changing the instruction set to be decoded. Ueda does not teach a mode register to indicate the instruction set. Indeed, Ueda is directed to parallel processing, and can execute both first and second microinstructions simultaneously in the two separate operation units. Ueda does not teach a mode where only second microinstructions are executed, unlike the present invention, which can execute either first or second instructions.

In summary, Ueda is absent any teaching or suggestion of the select means recited in claim 1. Ueda sends first microinstructions to a first operation unit and second microinstructions to a second operation unit. The present invention uses the select means recited in claim 1 to send either first instructions or second instructions to the execute means. Ueda neither teaches nor suggests an execute means capable of executing both first and second instructions, as selected by a select means. Ueda thus cannot render claim 1 anticipated, nor obvious. Ueda also neither teaches nor suggests an instruction fetch buffer coupled to the first and the second instruction decoders, as recited in claim 2, and the mode register means and control recited in claims 3 and 4.

PRIOR ART REJECTIONS - 35 USC § 103 — DE NICOLAS ET AL.

Claims 1-20 were rejected under 35 USC § 103 as obvious over de Nicolas (US Pat. No. 5,167,023) hereinafter Nicolas. As to claims 1-4, Nicolas is cited as a system for emulating the execution of second set of instructions which are not directly executable by the host system. The system is capable of directly executing native RISC instructions in a normal mode and executing CISC target instructions in an emulation mode. Nicholas describes a number of prior-art systems, one of which was implemented the instruction emulation with hardware. It would have been obvious that such hardware emulation would have included a second decoder for decoding target instructions, and other control units for controlling the emulation mode for executing target instructions. Applicant respectfully disagrees.

***Coprocessor Does Not Render the Invention Obvious***

Nicolas teaches a system that emulates instructions in a second instruction set using a plurality of instructions in a first instruction set. Nicolas neither teaches nor suggests a processor that can execute both CISC and RISC instructions in hardware except for briefly mentioning in the background section using a coprocessor. A coprocessor is a second entire processor with a separate execute unit. The present invention recites a select means to supply an execute means with decoded instructions from either of two

instruction sets. A system with a coprocessor has two separate execute units, one unit for executing first instructions and a second unit for executing second instructions. However, neither unit can execute both first and second instructions. Such a system would lack an execute means that executes "decoded instructions selected by the select means", which include both first and second decoded instructions. A coprocessor-based system would also lack the select means of claim 1. A coprocessor does not render the recited invention obvious.

A coprocessor is similar to the multi-program mode of Ueda, and the structural differences discussed above in reference to Ueda also apply here. There would be no connection for decoded second instructions from a second decoder in the coprocessor to the execute means in the main processor. In contrast, the recited invention uses a select means to route decoded second instructions to the execute means. Thus the coprocessor discussion in Nicolas does not teach or even suggest using a select means or an execute means for executing both first and second instructions.

Coprocessors teach away from the present invention because a separate execute unit is used for the second instructions. The present invention has the new result of having a single execute unit that can execute instructions from two separate instruction sets, eliminating the need for and cost of a second (co-) processor to execute second instructions. This new result, lower cost from eliminating an expensive component (the co-processor) argues against obviousness since there was strong financial motive for others to use the present invention, yet the prior art does not teach or suggest the present invention. Therefore a coprocessor does not render the present invention obvious and indeed its very existence argues against obviousness.

***Emulation Does Not Render the Invention Obvious***

Nicolas teaches emulating second (simulated) instructions by replacing them with a plurality of first (host machine) instructions. Nicolas is directed to the problem of

reducing the number of first instructions needed to emulate a second instruction, from 50-100 down to 4 (col 4 lines 12-26). The present invention solves Nicolas's problem by directly executing some or all of the second instructions. The present invention thus effectively reduces the "plurality of first instructions" down to *one* second instruction. This is a great speed advantage over Nicolas, and yet he does not even suggest directly executing second instructions, except with a separate coprocessor, which he says is too limiting (col 2 lines 47-55). Instead, Nicolas emulates second instructions. Nicolas certainly does not teach or suggest a combination of emulation and direct execution of second instructions, despite the great advantage that such a system would have.

There is nothing in emulation of a second instruction set to suggest the select means or the execute means recited in claim 1, because the select means and the execute means receive decoded instructions from both the first and the second instruction sets, whereas emulation can only replace second instructions with first instructions. Thus any emulation system would provide only decoded first instructions to the execute unit. There would be no use for a select means, since there is nothing to select — only decoded first instructions are sent to the execute unit.

An emulation system thus neither teaches nor suggests the select means and the second decode means recited in claim 1. The execute means of such an emulation system would not execute both first and second instructions, as recited in claim 1 by the connection to the select means. Thus elements in claim 1 are missing and are not even suggested by Nicolas. As no second decode means is present or suggested, claim 2's instruction fetch buffer coupled to the first and second instruction decode means is also not suggested or obvious. Likewise claim 3's mode register means is not needed since only first instructions are ever executed by an emulation system. Second instructions cannot be directly executed, so no mode register is needed.

Emulation does not render claims 1-4 obvious, and recited claim elements, such as the select means, are not taught or even suggested by Nicolas. Thus Nicolas does not render claims 1-4 obvious.

***Claims 5-10 Not Obvious***

As to claims 5-10, Nicolas is cited as teaching emulating the execution of CISC instructions using software where each CISC instruction execution is emulated by executing a routine comprising a plurality of individual RISC instructions, and where the emulation mode is initiated by an interrupt signal. Nicolas further teaches using a translation-lookaside buffer (TLB) for providing dynamic address conversion for executed instructions. Nicolas does not particularly teach incorporating both hardware and software emulation in the same machine as claimed. It would have been obvious to realize such a hardware and software combination because it would bring the advantages of both techniques into the system, e.g. the inexpensive and flexibility of the software emulation with the speed of the hardware emulation. Applicant respectfully disagrees.

As claims 5-10 are dependent upon independent claim 1, the above-mentioned reasons as stated in reference to claim 1 apply with equal force and effect to claims 5-10.

Nicolas emulates all second instructions. There is no suggestion that only a portion of the second instruction set is emulated while another portion of the second instruction set is decoded and directly executed. Claim 5 recites that the second decode means decodes only a portion of the second instruction set. Nicolas does not teach or suggest a second decode means, nor does Nicolas teach or suggest that a second decode means would only decode a portion of the second instruction set. Only by hindsight using the claims of the present invention as a blueprint can it be suggested that Nicolas teaches what is recited in the claims.

Claim 5 further recites that the second decode means indicates to the mode control means when an instruction is not in the decoded portion of the second instruction set, thus changing to the first instruction set to allow for emulation or other handling by the first instruction set. Applicant was unable to find any reference in the background section of Nicolas to using an interrupt to initiate emulation mode, and certainly such a signal would not be generated by a second instruction decoder as none exists in Nicolas. Thus claim 5 cannot be obvious in view of Nicolas.

Claim 6 recites that the mode control means be signaled to switch to the first instruction set when no translation is found in the TLB. Again, nothing in Nicolas would suggest or imply that such a signal be generated or necessary, as Nicolas only executes first instruction and has no mode control. Likewise having an exception in the execute means signaling a switch to the first instruction set is nowhere suggested in Nicolas.

Although the present invention has the advantages cited by the Examiner as the reason that it would be obvious to realize such a software and hardware combination, the low cost and flexibility of software emulation and the speed of hardware, and despite the amount of inventive activity in emulation, no cited reference teaches the invention. These advantages and the failure of others to teach the invention argue in favor of non-obviousness. Thus claims 5-10 are not obvious in view of Nicolas.

As per claims 11-13, it is cited as being obvious to use additional instructions to modify the TLB or to switch to emulation mode in response to a signal from the execution unit or a reset. Nowhere is this taught or suggested in Nicolas. Hindsight should not be used to reconstruct the claimed invention using the blueprint drawn by the inventor. Indeed, claimed elements of the present application are not even suggested by the cited references. Therefore the claimed invention cannot be obvious unless hindsight is used with the claimed invention as a blueprint, supplying the

missing elements. Along with the reasons set forth above, these claims are not obvious in view of Nicolas.

Claims 14-20 are likewise not obvious in light of the argument presented above. Independent claim 14 is directed to a method for processing instructions from two separate instruction sets. Independent claim 15 is directed to a method for processing instructions from a CISC and a RISC instruction set in which all CISC instructions are executable, either directly by the execute unit or by emulation with RISC instructions.

Independent claim 18 is directed to a microprocessor for executing RISC and CISC instructions using both RISC and CISC instruction decoders and an enable means to enable one of the instruction decoders. Emulation mode is entered if a CISC instruction decode unit is not able to fully decode a CISC instruction, but the microprocessor directly executes the CISC instruction if it is decodable. Emulation mode uses a RISC instruction decode unit. Nowhere in the cited references is this combination taught or even suggested.

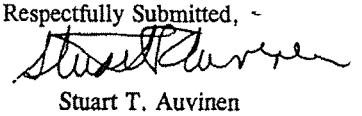
In view of the above, it is submitted that claims 1-20, as amended, are in a position for allowance. Applicant requests that the requirement for formal drawings be held in abeyance until allowance. Applicant believes that a full and complete response to the office action has been made. Reconsideration and re-examination is respectfully requested. Allowance of the claims at an early date is solicited.

If the Examiner believes that a telephone interview would expedite prosecution of this application, he is invited to telephone the undersigned at (408) 476-5506.

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

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

Respectfully Submitted,



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


PATENT

| AMENDMENT TRANSMITTAL LETTER | | | Docket No. RM-1 |
|---|---|---|---|
| Serial No: 08/179,926 | Filing Date: 1/11/94 | Examiner: V. Vu | GAU: 2315 |
| Title: Dual-Instruction-Set Architecture CPU with Hidden Software Emulation Mode | | | |

Honorable Commissioner of Patents and Trademarks
Washington, DC 20231



Transmitted herewith is an amendment in the above-identified application.

X Small entity status of this application under 37 C.F.R. § 1.27 has been established by a verified statement previously submitted.

X No additional fee is required.

☐ The fee has been calculated as shown below:

| | Claims remain. after Amend-ment | | Highest No. prev. paid for | Present Extra | Rate | Add'l Fee | Rate | Add'l Fee |
|---|---|---|---|---|---|---|---|---|
| Total | | Minus | | | x $11 | | x $22 | |
| Indepen dent | | Minus | | | x $37 | | x $74 | |
| First Presentation of Multiple Dep. Claim | | | | | + $115 | | + $230 | |
| | | | | | Total | | Total | |

☐ A check in the amount of $_____ to cover these fees is enclosed.

☐ Please charge my deposit account No. **01-2950** in the amount of $ _____ for these fees. A duplicate copy of this sheet is enclosed.

X The commissioner is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to deposit account No. **01-2950**. A duplicate copy of this sheet is enclosed.

- ☐ Any additional filing fees required under 37 C.F.R. § 1.16.
- X Any patent application processing fees under 37 C.F.R. § 1.17.

Respectfully Submitted,

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

Address all inquires to:
Stuart T. Auvinen
429 26th Ave.
Santa Cruz, CA 95062
(408) 476-5506

I hereby certify that this paper is being deposited with the United States Postal Service with sufficient postage as First Class Mail under 37 CFR 1.8 on the date indicated below and is addressed to: "Commissioner of Patents and Trademarks, Washington, DC 20231."

Signed _____
Stuart T. Auvinen, Reg. No. 36,435

Date 9/20/94