

Petitioner's Demonstratives

BMW of North America, LLC

v.

Foras Technologies Ltd.

IPR2024-01346, IPR2024-01347

December 5, 2025

Patents at Issue

U.S. Patent No. 7,624,302

System and Method for Switching the Role of Boot Processor to a Spare Processor Responsive to Detection of Loss of Lockstep in a Boot Processor

(12) **United States Patent**
Michaelis et al. (10) **Patent No.:** **US 7,624,302 B2**
(45) **Date of Patent:** ***Nov. 24, 2009**

(54) **SYSTEM AND METHOD FOR SWITCHING THE ROLE OF BOOT PROCESSOR TO A SPARE PROCESSOR RESPONSIVE TO DETECTION OF LOSS OF LOCKSTEP IN A BOOT PROCESSOR**

(75) Inventors: **Scott L. Michaelis**, Plano, TX (US); **Anurupa Rajkumari**, Round Rock, TX (US); **William B. McHardy**, Fort Collins, CO (US)

(73) Assignee: **Hewlett-Packard Development Company, L.P.**, Houston, TX (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 513 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **10/972,588**

(22) Filed: **Oct. 25, 2004**

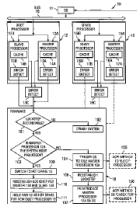
(65) **Prior Publication Data**
US 2006/009064 A1 Apr. 27, 2006

(51) **Int. Cl.** **G06F 11/00** (2006.01)
(52) **U.S. Cl.** **714/12; 714/11**
(58) **Field of Classification Search** **714/10-13; 712/31**

See application file for complete search history.

References Cited
U.S. PATENT DOCUMENTS
4,358,823 A 11/1982 McDonald et al.
5,226,152 A 7/1993 Klug et al.
5,249,188 A 9/1993 McDonald
5,537,655 A 7/1996 Truong
5,675,794 A 10/1997 Meredith
5,751,932 A 5/1998 Horst et al.
5,758,058 A 5/1998 Milburn

27 Claims, 4 Drawing Sheets



Similar specifications

Switching the role of the boot processor to a spare processor during system runtime

U.S. Patent No. 7,627,781

System and Method for Establishing a Spare Processor for Recovering from Loss of Lockstep in a Boot Processor

(12) **United States Patent**
Michaelis et al. (10) **Patent No.:** **US 7,627,781 B2**
(45) **Date of Patent:** ***Dec. 1, 2009**

(54) **SYSTEM AND METHOD FOR ESTABLISHING A SPARE PROCESSOR FOR RECOVERING FROM LOSS OF LOCKSTEP IN A BOOT PROCESSOR**

(75) Inventors: **Scott L. Michaelis**, Plano, TX (US); **Anurupa Rajkumari**, Round Rock, TX (US)

(73) Assignee: **Hewlett-Packard Development Company, L.P.**, Houston, TX (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 520 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **10/973,077**

(22) Filed: **Oct. 25, 2004**

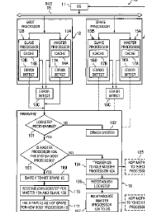
(65) **Prior Publication Data**
US 2006/0107112 A1 May 18, 2006

(51) **Int. Cl.** **G06F 11/00** (2006.01)
(52) **U.S. Cl.** **714/13; 714/11**
(58) **Field of Classification Search** **714/10-12; 714/13, 31; 712/31**

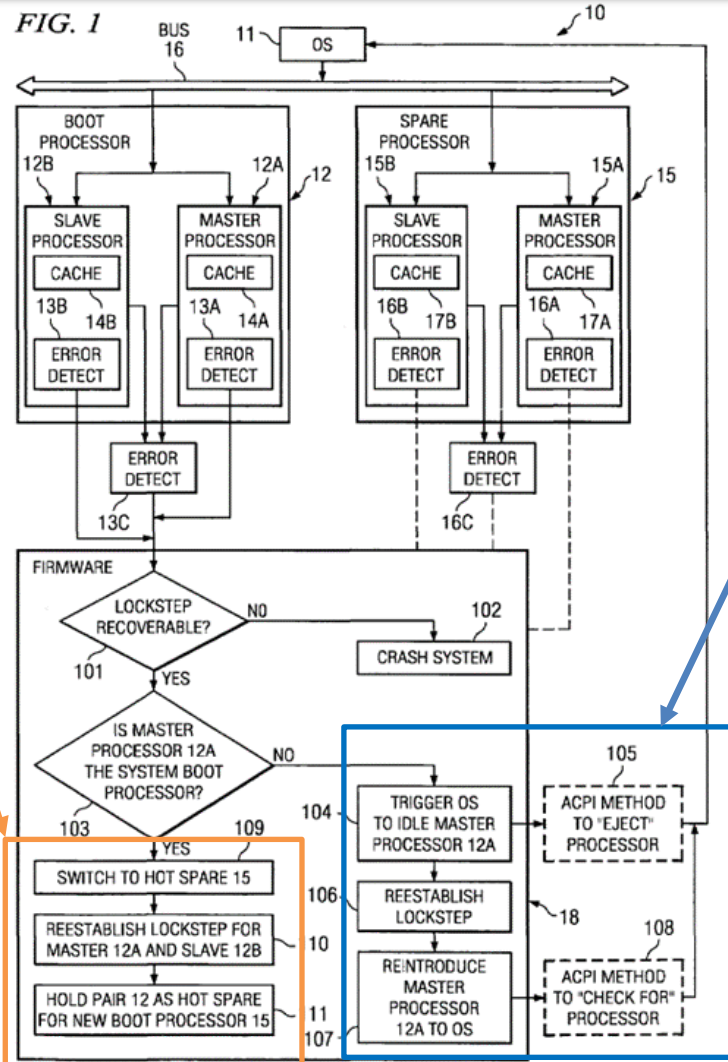
See application file for complete search history.

References Cited
U.S. PATENT DOCUMENTS
8,823 A 11/1982 McDonald et al.
6,152 A 7/1993 Klug et al.
9,188 A 9/1993 McDonald
7,655 A 7/1996 Truong
5,794 A 10/1997 Meredith
1,932 A 5/1998 Horst et al.
8,058 A 5/1998 Milburn
4,660 A 6/1998 Mohat
6,523 A 4/1999 Bissett et al.

30 Claims, 4 Drawing Sheets



Patents at Issue



Switching the role of the boot processor to a spare processor during system runtime

US 7,624,302

US 7,627,781

Terminally disclaims against each other

Lockstep recovery

US 7,502,958

IPR2023-01373:
All Challenged Claims
Unpatentable

IPR2023-01373: FWD (Paper 34), 35.

Overview of the '958 Patent IPR/EPR Proceedings

Case No.	Claims	Ground	Status
IPR2023-01373	1-8, 13-22	Bigbee, Nguyen	FWD issued finding all challenged claims unpatentable; Notice of Appeal filed
EPR90/019,245	1-8, 13-22	Bigbee	Stayed pending termination or completion of IPR2023-01373
	1-8, 13-22	Safford, Marisetty	
	9-12, 23-25	Safford, Cepulis, Marisetty	
EPR90/019,922	9	Bigbee, Nguyen, Landry	Reexam ordered 07/08/2025; Awaiting Office Action
	10-12	Bigbee, Nguyen, Landry, Safford	
	23-25	Bigbee, Landry, Safford	

Different references and combinations are used here

IPR2023-01373: Pet. (Paper 1), 76-77; FWD (Paper 34), 35.

Overview of the '781 and '302 Patents EPR Proceedings

Patent No.	Case No.	Claims	Prior Art	Status
7,627,781	EPR 90/019,244	1-6, 9-15	Kalyanasundharam, Kennedy	Claims 1-2, 5-18, 21-30 finally rejected; appealed to the PTAB
		1-6, 9-15	Safford, Kennedy	
		1-30	Kennedy, Nguyen, Fox	
		1, 6	Safford, Cepulis	
7,624,302	EPR 90/019,243	1, 5-8, 13, 17, 21, 22, 25-26	Kalyanasundharam, Kennedy	Reexam certificate issued 08/27/2024; claims 1, 5-8, 13, 17, 21-22, 25-26 affirmed
		1-27	Kennedy, Nguyen, Safford	
		1-4	Safford, Cepulis	
	EPR 90/015,657	11, 12, 17, 24	Fox, Safford, Arai	Assigned to Examiner
		11, 12, 17, 24	Fox, Landry, Safford, Arai	Same references and combinations are used here

IPR2024-01346: Pet. 70-72; Ex. 1032.

IPR2024-01347: Pet. 77-79; Ex. 1031; Paper 20 (Petitioner's Updated Mandatory Notices).

'781 Patent – Grounds of Unpatentability

Ground	Claims Challenged	References
1	1-2, 5-7, 9-18, 22-23, 25-26, 29-30	Fox, Safford, Arai
2	1-2, 5-7, 9-18, 22-23, 25-26, 29-30	Fox, Safford, Landry, Arai
3	3-4, 19-21, 27-28	Ground 1 + Jin
4	3-4, 19-21, 27-28	Ground 2 + Jin
5	8, 24	Ground 1 + Bigbee
6	8, 24	Ground 2 + Bigbee

- All claims of the '781 Patent are challenged
- Claims 1, 9, 16, and 25 are independent claims
 - Patent Owner's arguments for independent claims 9, 16, and 25 rely on same arguments for claim limitations 1[b]/[1c]
- Patent Owner *separately* argues
 - dependent claim 2 reciting a “non-volatile system memory”
 - dependent claim 14 reciting “an active spare”
- Patent Owner does not *separately* argue dependent claims 3-8, 10-13, 15, 17-24, and 26-30

IPR2024-01346: Pet. 2; Institution Decision (Paper 7), 6-7; POR (Paper 11), 6-21.

'302 Patent – Grounds of Unpatentability

Ground	Claims Challenged	References
1	1-25	Fox, Safford, Arai
2	1-25	Fox, Safford, Landry, Arai
3	26, 27	Ground 1 + Bigbee
4	26, 27	Ground 2 + Bigbee

- All claims of the '302 Patent are challenged
- Claims 1, 13, 21, and 26 are independent claims
 - Patent Owner's arguments for independent claims 13, 21, and 26 rely on same arguments for limitation [1d]
- Patent Owner *separately* argues
 - dependent claims 3, 4, 15, 16, and 24 reciting an “active spare”
 - dependent claims 6, 8, and 25 reciting a “non-volatile data storage”
 - dependent claims 11, 12, 17, and 24 are not addressed in the Petition
- Patent Owner does not *separately* argue dependent claims 2, 5, 7, 9, 10, 14, 18–20, 22, 23, and 27

IPR2024-01347: Pet. 1-2; Institution Decision (Paper 7), 6-7; POR (Paper 11), 11-50.

The '302 Patent Pending EPR Proceeding

Patent No.	Case No.	Claims	Prior Art	Status
7,624,302	EPR 90/015,657	11, 12, 17, 24	Fox, Safford, Arai	Assigned to Examiner
		11, 12, 17, 24	Fox, Landry, Safford, Arai	

- The Board is equipped with the arguments and evidence to review claims 11, 12, 17, and 24 based on the record
- The Board may consider consolidating IPR2024-01347 and EPR 90/015,657 for the '302 Patent (M.P.E.P. § 2286.01)

Claim	Limitations	Ground 1	Ground 2
11	[1a], [13b], [13c]	Pet. 9-25, 38-43; Brogioli Decl. ¶¶82-108, 111-114, 153-162.	Pet. 30-33, 38-43; Brogioli Decl. ¶¶129-135, 138, 153-162.
12	[1a], [13b], [13c]	Pet. 9-25, 38-43; Brogioli Decl. ¶¶82-108, 111-114, 153-162.	Pet. 30-33, 38-43; Brogioli Decl. ¶¶129-135, 138, 153-162.
17	[1d], [13e], [21e]	Pet. 9-22, 28-30, 43, 51-53; Brogioli Decl. ¶¶124-128, 164, 185-190.	Pet. 30-33, 36, 43, 51-53; Brogioli Decl. ¶¶129-135, 145, 164, 185-190.
24	Claims 6, 8, 25	Pet. 59-62; Brogioli Decl. ¶¶207-214.	Pet. 59-62; Brogioli Decl. ¶¶207-214.

IPR2024-01347: POR 48-50; Reply (Paper 14), 18-22.

'781 Patent – Claim Construction

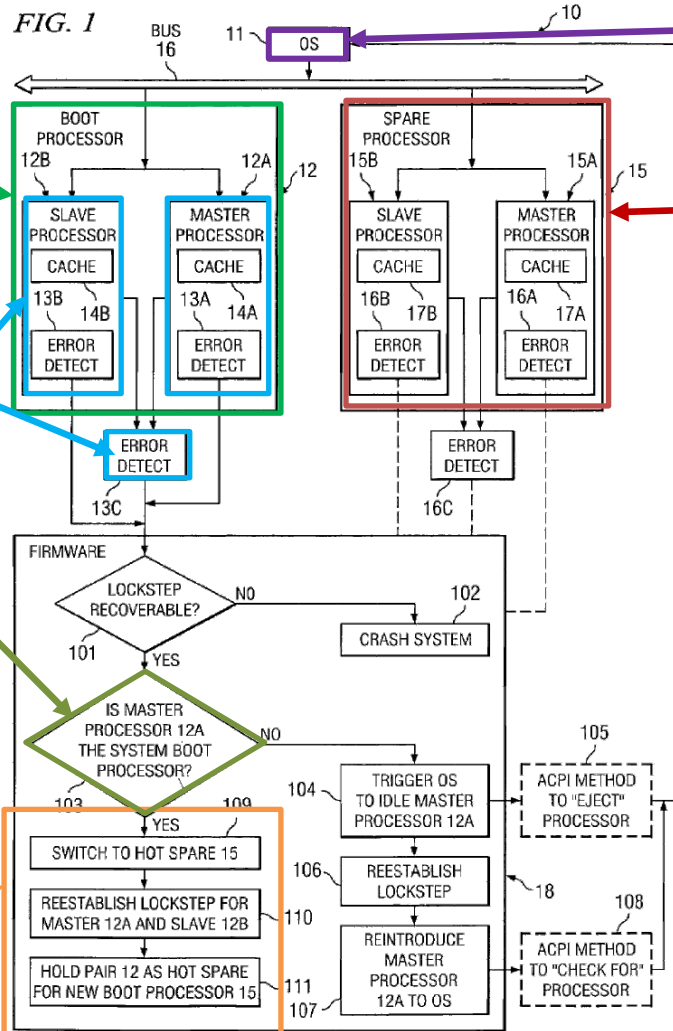
Claim Term	Petitioner	Patent Owner's Response	Institution Decision
"detecting loss of lockstep"	"including at least one of (1) detecting a lockstep mismatch error; and (2) detecting an error that may eventually cause a lockstep mismatch error"	No dispute	Not disputed; no construction necessary
"logic operable ... for transferring, during system runtime, the role of boot processor to said spare processor"	"firmware logic"	No dispute	Not disputed; no construction necessary

'302 Patent – Claim Construction

Claim Term	Petitioner	Patent Owner's Response	Institution Decision
"detection of loss of lockstep" and "detecting loss of lockstep"	"including at least one of (1) detecting a lockstep mismatch error; and (2) detecting an error that may eventually cause a lockstep mismatch error"	No dispute	Not disputed; no construction necessary
"system firmware means"	Means-plus-function	No dispute	Not disputed; no construction necessary

Overview of the '302 and '781 Patents

FIG. 1



Operating system is running during the switch

Spare Processor

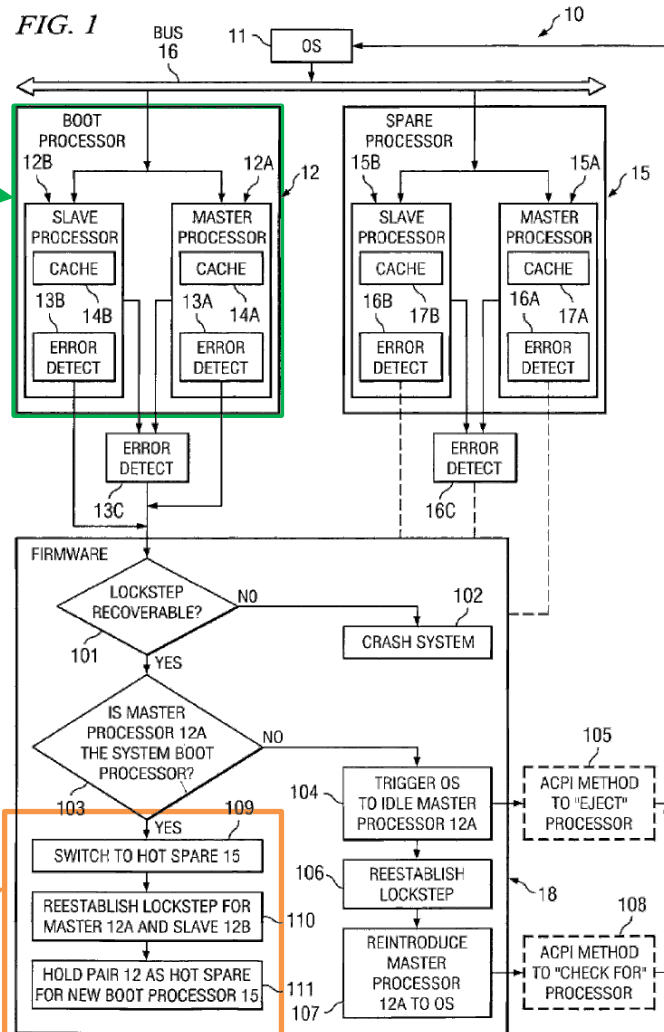
Boot Processor

Lockstep processors and LOL detection

Determining if the LOL processor is boot processor

Switching the role of the boot processor to a spare processor during system runtime

Overview of the Issues



Two core disputed issues

1. Whether the prior art teaches a boot processor

2. Whether the prior art teaches switching the *role* of the boot processor to the spare processor during system runtime

Grounds of Unpatentability

IPR2024-01346 ('781 Patent)

Ground	Claims Challenged	References
1	1-2, 5-7, 9-18, 22-23, 25-26, 29-30	Fox, Safford, Arai
2	1-2, 5-7, 9-18, 22-23, 25-26, 29-30	Fox, Safford, Landry, Arai
3	3-4, 19-21, 27-28	Ground 1 + Jin
4	3-4, 19-21, 27-28	Ground 2 + Jin
5	8, 24	Ground 1 + Bigbee
6	8, 24	Ground 2 + Bigbee

IPR2024-01347 ('302 Patent)

Ground	Claims Challenged	References
1	1-25	Fox, Safford, Arai
2	1-25	Fox, Safford, Landry, Arai
3	26, 27	Ground 1 + Bigbee
4	26, 27	Ground 2 + Bigbee

IPR2024-01346: Pet. 2; Institution Decision, 6-7; POR 6-21.

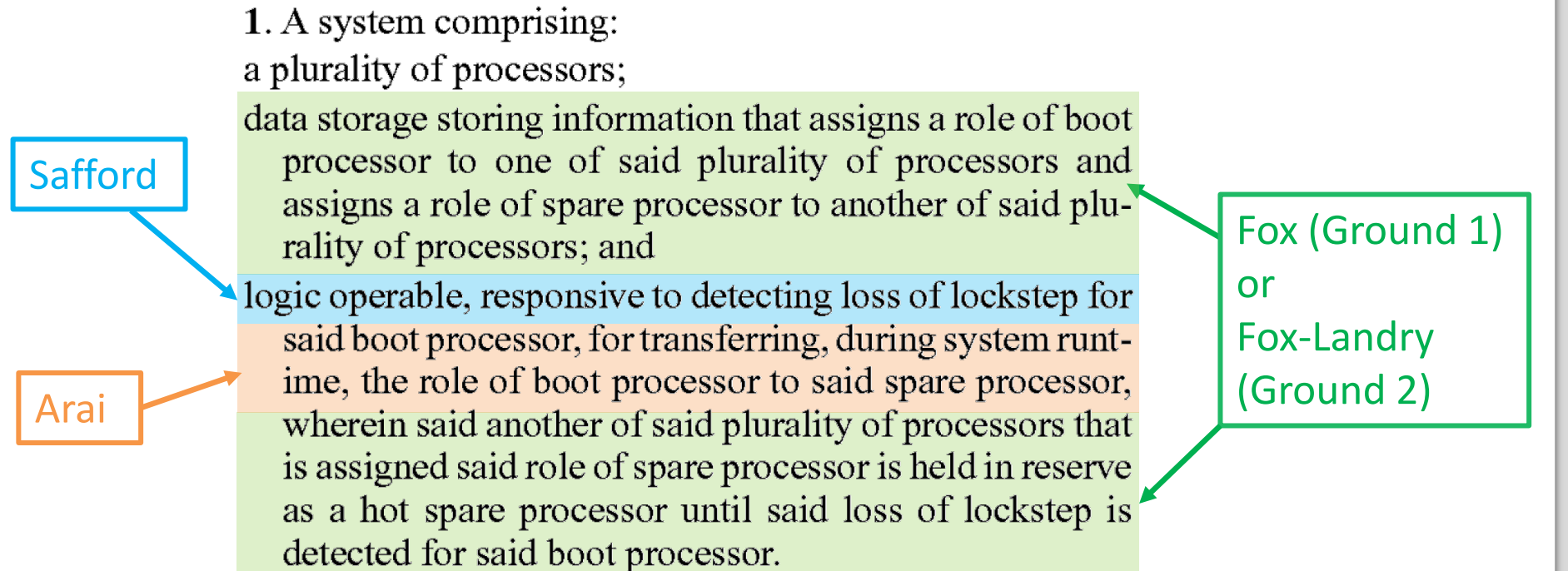
IPR2024-01347: Pet. 1-2; Institution Decision, 6-7; POR 11-50.

Overview of Grounds 1 and 2

'781 Patent – Representative Challenged Claim

1. A system comprising:
a plurality of processors;
data storage storing information that assigns a role of boot processor to one of said plurality of processors and assigns a role of spare processor to another of said plurality of processors; and
logic operable, responsive to detecting loss of lockstep for said boot processor, for transferring, during system runtime, the role of boot processor to said spare processor, wherein said another of said plurality of processors that is assigned said role of spare processor is held in reserve as a hot spare processor until said loss of lockstep is detected for said boot processor.

Safford



Arai

Fox (Ground 1)
or
Fox-Landry
(Ground 2)

Overview of Grounds 1 and 2

'302 Patent – Representative Challenged Claim

1. A method comprising:

detecting loss of lockstep (LOL) for a processor in a multi-processor system;

determining that the processor for which said LOL is detected is assigned a role of boot processor;

changing a system identifier, used by the system's operating system to recognize the boot processor, of said boot processor for which said LOL is detected, to a different value, which causes the system's operating system to not recognize the processor for which said LOL is detected as the boot processor; and

changing a system identifier of the spare processor to that of the boot processor, thereby switching the role of boot processor to a spare processor without shutting down the system's operating system.

Safford

Arai

Fox (Ground 1)
or
Fox-Landry
(Ground 2)

Ground 1: Fox, Safford, and Arai

Fox

Boot Processor
(IA-32 system)

Spare Processor

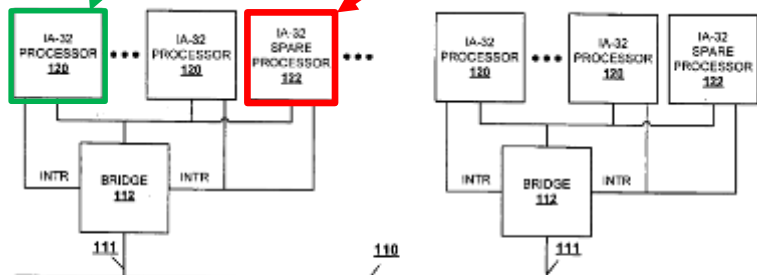
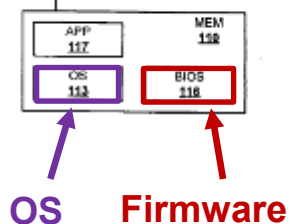


Figure 1



- Fox discloses:
 - IA-32 multiprocessor system
 - Switch a failing processor with a spare processor
- Fox does not explicitly disclose:
 1. Lockstep processors
 2. Processor switching details: changing system identifiers

Ground 1: Fox, Safford, and Arai

Fox

Boot Processor
(IA-32 system)

Spare Processor

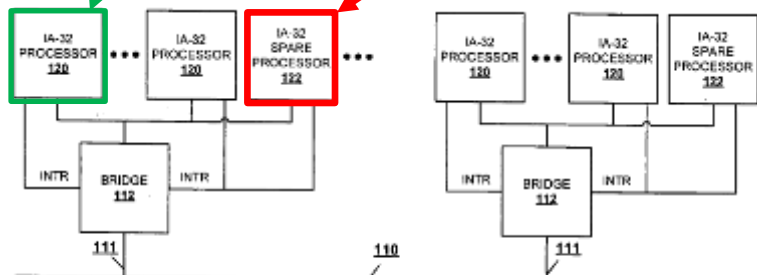
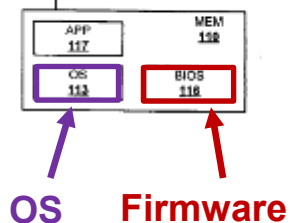
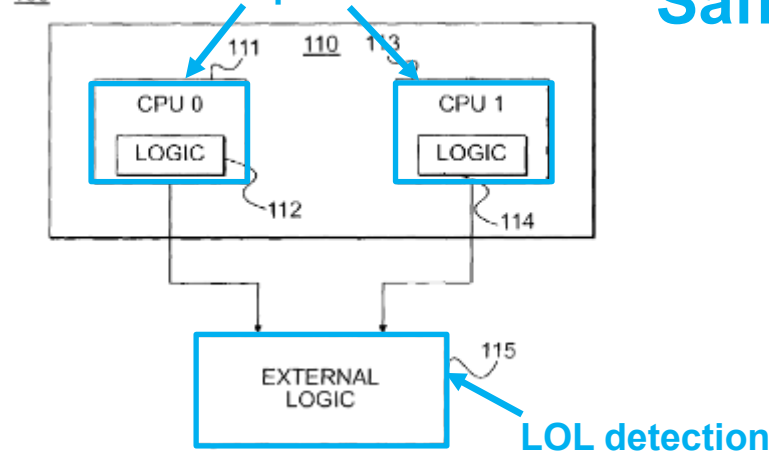


Figure 1



Lockstep Processors

Safford



- Fox in view of Safford discloses:
 1. Lockstep processors

Ground 1: Fox, Safford, and Arai

Fox

Boot Processor
(IA-32 system)

Spare Processor

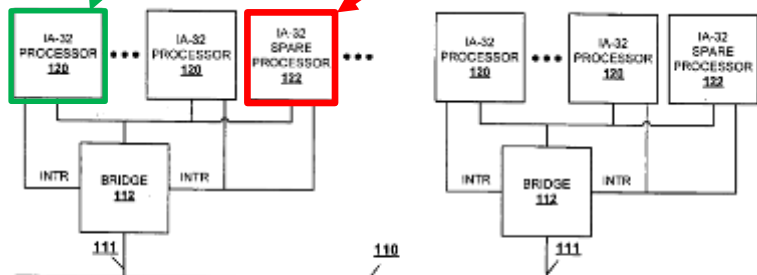
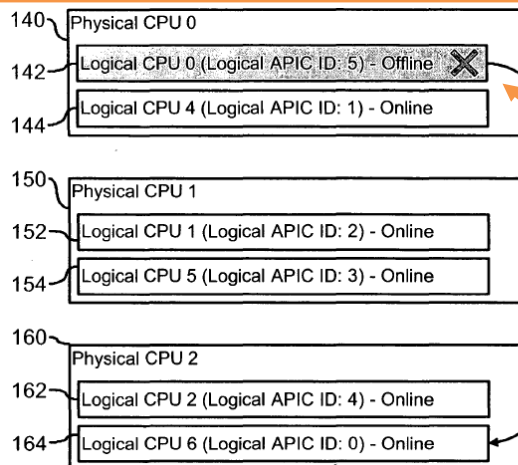


Figure 1

OS Firmware

Arai



Changing
system
identifiers

- Fox in view of Safford and Arai discloses:

1. Lockstep processors
2. Processor switching details: changing system identifiers

Two Core Disputed Issues (All Claims)

No.	Issue
1.	Whether the prior art teaches a boot processor
2.	Whether the prior art teaches switching the role of the boot processor to the spare processor during system runtime

Using a Boot Processor to Initialize a Multiprocessor System Was Known and Conventional

Not disputed

Schelling (Ex. 1014)

[0003] One common approach to minimizing the difficulties associated with the early stages of firmware execution within multiprocessor systems is to select a single processor to function as the boot strap processor (BSP). As is well

Jayakumar (Ex. 1018)

Typically, the individual processors each test their components and interfaces. One processor is then selected from their number to establish communications with the rest of the system and complete the system-wide steps of the bootstrapping process. The selected processor is the bootstrap processor. In Intel processors and processors compat-

Datta (Ex. 1023)

internal components and interfaces. In multiprocessor systems, a single bootstrap processor (BSP) is usually selected to handle initialization procedures for the system as a whole. These procedures include checking the integrity of

Ground 1: Fox Teaches a Boot Processor

Fox

**Boot Processor
(IA-32 system)**

Spare Processor

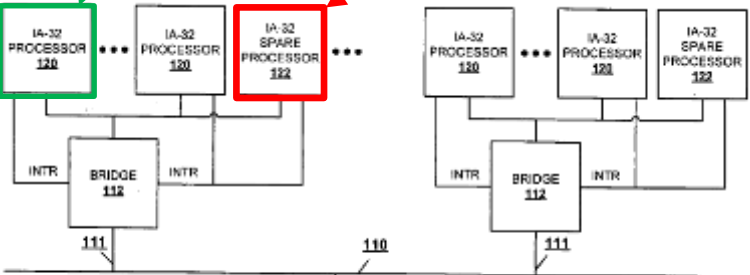


Figure 1

BIOS

Not disputed

- Fox discloses:
 - A multiprocessor system configured with IA-32 architecture
- “System boot” or “BIOS boot” utilizing the BIOS

16 Q So you agree Fox discloses the IA-32
17 configured multiprocessor system?
18 A Yes.

17 Q Okay. So you agree that Fox's system is
18 provided with BIOS of the IA-32 systems?
19 A That's not what that says.
20 I agree that Fox is provided with -- Fox
21 discloses the use of a BIOS.

[0040] Implementation of the invention involves the utilization of Systems Management Interrupt (SMI) functionality provided within basic input output system (BIOS) of the IA-32 systems. At least one processor of the multipro-

Ground 1: Fox Teaches a Boot Processor

Patent Owner's View

Fox does not disclose, teach, or suggest a boot processor.

Fox does not disclose, teach, or suggest a BSP flag.

Contradicted

Evidence in the Record

Fox's IA-32 multiprocessor system designates a boot processor during system initialization.

IA-32 Software Developer's Manual (Ex. 1008)

The MP initialization protocol defines two classes of processors: the bootstrap processor (BSP) and the application processors (APs). Following a power-up or RESET of an MP system, system hardware dynamically selects one of the processors on the system bus as the BSP. The remaining processors are designated as APs.

Fox's IA-32 system's boot processor has unique identification (e.g., APIC ID) and designation (e.g., BSP flag)

7.5.5. Identifying the Processors in an MP System

After the BIOS has completed the MP initialization protocol, each processor can be uniquely identified by its local APIC ID. Software can access these APIC IDs in either of the following ways:

BSP flag, bit 8 Indicates if the processor is the bootstrap processor (BSP) (see Section 7.5., "Multiple-Processor (MP) Initialization"). Following a power-up or RESET, this flag is set to 1 for the processor that was selected as the BSP and set to 0 for each of the remaining application processors (APs).

IPR2024-01346: Pet. 15-17; POR 7-9, 15; Reply 8-12; Brogioli Decl. ¶¶44-47, 58-62, 100-106; IA-32, 238-239, 244, 277.

IPR2024-01347: Pet. 15-18; POR 15-17, 20-23; Reply 12-16; Brogioli Decl. ¶¶45-48, 59-63, 97-102; IA-32, 238-239, 244, 277.

Ground 1: Fox Teaches a Boot Processor

Patent Owner's View

Fox does not disclose, teach, or suggest a boot processor.

Fox does not disclose, teach, or suggest a BSP flag.

Contradicted

Evidence in the Record

MultiProcessor Specification (Ex. 1028)

The BSP is responsible for initializing the system and for booting the operating system; APs are activated only after the operating system is up and running.

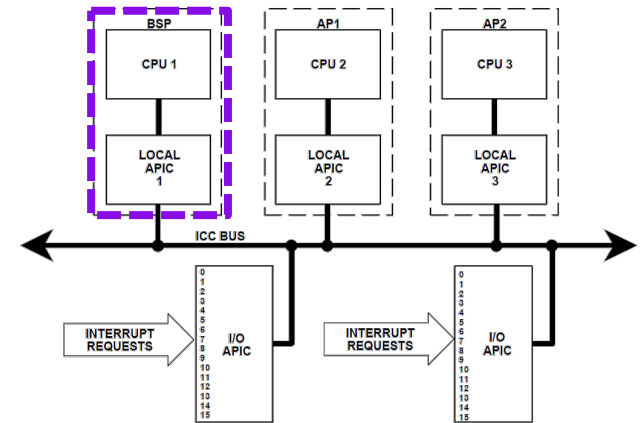


Figure 2-2. APIC Configuration

Schelling (Ex. 1014)

[0004] Within some multiprocessor systems such as, for example, Intel Corporation IA-32 based systems, selection of a BSP is automatically performed using a front-side bus (FSB) protocol. With the FSB protocol, each processor

IPR2024-01346: Pet. 15-17; POR 7-9, 15; Reply 8-12; Brogioli Decl. ¶¶44-47, 58-62, 100-106; MP Specification, 17, 75, 79.

IPR2024-01347: Pet. 15-18; POR 15-17, 20-23; Reply 12-16; Brogioli Decl. ¶¶45-48, 59-63, 97-102; MP Specification, 17, 75, 79.

Ground 1: Fox Teaches a Boot Processor

Patent Owner's Expert, Mr. Jones

- “Fox does **not explicitly** disclose a boot processor.”
- “[W]ith Fox, how you get up and running is **irrelevant**.”
- A POSITA “would be **completely agnostic** to it.”
- A POSITA “**wouldn't have to know what a boot processor is**.”
- “[O]ne of ordinary skill in the art with an IA-32 multiprocessor system would recognize that there would be **multiple ways of booting that system**.”
- Using a processor executing BIOS is “**one of multiple possibilities**” for Fox's BIOS boot.

Contradicted

Petitioner's Expert, Dr. Brogioli

- One of the processors of Fox's multiprocessor system is a boot processor because **a boot processor is necessary to execute Fox's BIOS to initialize Fox's multiprocessor system configured with IA-32 architecture**.
 - Fox, [0015], [0034], [0040], [0045];
 - IA-32, 238;
 - Schelling, [0003]-[0005];
 - MP Specification, 17.
- Fox's system would have designated, **or it would have been obvious to designate**, one of Fox's processors as the boot processor.

IPR2024-01346: Jones Decl. (Ex. 2004), ¶28; Brogioli Decl. ¶¶44-47, 58-62, 100-106; Reply 8, 11, 19-20; Jones Tr. 69:7-22, 74:2-75:1, 81:3-85:15, 94:22-98:9.
IPR2024-01347: Jones Decl. (Ex. 2004), ¶28; Brogioli Decl. ¶¶45-48, 59-63, 97-102; Reply 12, 14, 23-24; Jones Tr. 69:7-22, 74:2-75:1, 81:3-85:15, 94:22-98:9.

Fox Teaches Storing Information Assigning a Boot Processor

Patent Owner's View

'781 Patent Response:

Fox's registers or tables are "created by the boot processor" and cannot "assign[] a boot processor itself."

Contradicted

Evidence in the Record

Fox's software register or table stores boot processor identification (e.g., APIC ID) and designation (e.g., BSP flag)

[0033] FIG. 2B provides a block diagram representation of a multiprocessor chip designed with main IA-32 processors 120 and a spare processor 122. Also provided with MP chip 200 is a register 209 for tracking active processors on MP chip 200, as described below. In one alternate imple-

driver-OS-SMI handshake. Tracking which processors are active may be completed by an software table of active processors, which is available to be read by the OS, when determining work load allocation to each of the active processors.

[0038] The spare processor is interconnected similarly to each primary processor with the exception that BIOS does not initialize the spare nor initially present the spare processor to the OS following the POST. During POST, BIOS hides the spare processor from the OS. This is accomplished either through a setup of the MP table and/or an ACPI setup.

Fox Teaches Storing Information Assigning a Boot Processor

Patent Owner's View

'781 Patent Response:

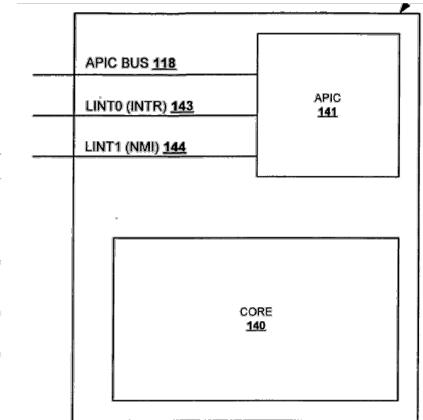
Fox's registers or tables are "created by the boot processor" and cannot "assign[] a boot processor itself."

Contradicted

Evidence in the Record

Fox

[0026] As shown by FIG. 2A, each processor 120, 122 includes a core unit 140 that controls processing on the processor 120, 122 and an APIC 141 that processes external and normal interrupts that the processor 120, 122 may receive at its interrupt pins 143 and 144. APIC 141 also includes a connection to the APIC bus 18, used to deliver interrupts to the application processors. The interrupt pins 143 and 144 may be programmed to receive specific types of interrupts.



IA-32

7.5.5. Identifying the Processors in an MP System

After the BIOS has completed the MP initialization protocol, each processor can be uniquely identified by its local APIC ID. Software can access these APIC IDs in either of the following

- **Read the ACPI or MP table.** As part of the MP initialization protocol, the BIOS creates an ACPI table and an MP table. These tables are defined in the Multiprocessor Specification Version 1.4 and provide software with a list of the processors in the system and their local APIC IDs. The format of the ACPI table is derived from the ACPI specification,

Fox Teaches Storing Information Assigning a Boot Processor

Patent Owner's View

'781 Patent Response:

Fox's registers or tables are "created by the boot processor" and cannot "assign[] a boot processor itself."

Contradicted

Evidence in the Record

'302 Patent, '781 Patent

In certain embodiments, information (referred to herein as
55 a "device tree") is stored to the multi-processor system,
wherein such information identifies the processor in the sys-
tem that is the boot processor and identifies the processor that
is the spare for the boot processor. The information is stored
such that it is accessible by the system's firmware, and the
60 firmware, upon detecting LOL for the boot processor, deter-
mines the spare processor from the device tree and assigns the
spare processor the role of boot processor. The firmware then

TABLE 1

Device	Status	Lockstep Enabled	Boot Processor?	Spare for Boot Processor?
Processor A	Present, Enabled, and Functioning	Yes	Yes	No
Processor B	N/A	Yes	No	Yes
Processor C	Present, Enabled, and Functioning	Yes	No	No
...	...	...	...	...

IPR2024-01346: Pet. 15-17, 22-23; POR 8-9; Reply 19; Brogioli Decl. ¶124; '781 Patent, 4:54-62, 12:59-65, 13:20-14:3, 16:31-61, 17:34-55.
IPR2024-01347: Pet. 15-18, 59-62; POR 27; Reply 12-16; Brogioli Decl. ¶212; '302 Patent, 5:44-48, 12:60-66, 13:21-14:3, 14:35-65, 16:1-17:3.

Mr. Jones Considered Only Some of the Evidence in the Record

4 Q Is it fair to say, in forming your
5 opinions in your '302 declaration and '781
6 declaration, you did not consider any Intel's
7 multiprocessor systems from before 2004?

8 A So in forming my opinions that appear in
9 my '302 declaration, I considered the documents
10 that I mention in that declaration.

11 Q That didn't answer my question. My
12 question is, in forming your opinions in your '302
13 and '781 declarations, did you consider any Intel
14 multiprocessor systems before 2004?

15 A So I disagree. I did answer your
16 question. I told you what I considered in forming
17 my opinion.

17 Q Yes. In forming your opinions in the
18 '781 declaration, did you consider any Intel
19 multiprocessor system from before 2004?

20 A So you basically asked me exactly the
21 same question, but this time with respect to the
22 '781 declaration.

42

1 Is that true?

2 Q Do you understand my question?

3 A Well, I tell you my understanding of the
4 question, but you've asked the same question but
5 with respect to the '781 declaration. And I would
6 say that the materials I considered for the '781
7 declaration are all of the documents that I
8 referred to in my declaration. In addition, I
9 considered the legal principles that are in
10 Section 3 of my declaration. I considered one of
11 ordinary skill in the art in Section 4.

12 Q Anything else?

13 A I'm nearly there.

14 So again, looking through here, I don't
15 see any reference to the petition, but I certainly
16 considered the petition as well.

17 Q Anything else?

18 A Nothing is coming to mind.

Mr. Jones Considered Only Some of the Evidence in the Record

17 Q Have you reviewed it in its entirety?

18 A It's 798 pages long. I have not read it
19 in its entirety. I have certainly looked at the
20 sections that Mr. -- excuse me, Dr. Brogioli
21 referenced in his declaration.

22 Q And those portions that you have

100

1 reviewed, do you understand what those portions
2 disclose?

3 MR. AUCHTER: Objection. Form.

4 A I had no difficulty understanding the
5 document. I'm not sure that it discloses
6 anything.

5 Q And that person would like to know how an
6 IA-32 processor works.

7 A Yes.

8 Q Would that person consult a manual
9 published by Intel for that processor?

10 MR. AUCHTER: Objection. Form.

11 A Well, it depends what they were working
12 on exactly.

13 They could be working at a level where
14 the particulars of the processor aren't relevant,
15 so no. They could consult with a more senior
16 engineer. So there are alternatives to that
17 premise.

18 Would there be some people that would
19 consult Intel documentation? Yes.

1 on the Exhibit 1008. But my opinion about Exhibit
2 1028 is the same as the Exhibit 1008, in that
3 there is no evidence that this was available to a
4 POSITA at the time of the invention of the '781
5 patent. And it's a draft document. And I would
6 be amazed if Intel had published this in this
7 form.

8 Q Do you understand what you read in this
9 document?

10 A Well, the little that I bothered to read,
11 I certainly understood. It's not a complicated
12 document. And as I said, it's a draft document,
13 so I can't believe that this was ever published in
14 this form. And so its relevance in this matter
15 escapes me.

16 Q You're not giving a legal opinion of
17 whether this document was publicly available as of
18 the priority date of the patents. Right?

19 A Well, I'm not offering any legal opinion.

IPR2024-01346: Jones Tr. 99:17-100:6, 105:1-19, 107:1-19.

IPR2024-01347: Jones Tr. 99:17-100:6, 105:1-19, 107:1-19.

Ground 1: Fox Teaches a Boot Processor

1. A POSITA has background knowledge of a boot processor
2. A POSITA would have understood Fox discloses a boot processor
3. A POSITA would have understood Fox suggests a boot processor
4. Patent Owner improperly focused on explicit teaching
5. Patent Owner's expert failed to consider all relevant evidence in the record
6. Patent Owner's expert failed to opine from the perspective of a POSITA

Ground 2: Fox, Landry, Safford, and Arai

Fox

Boot Processor (IA-32 system)

Spare Processor

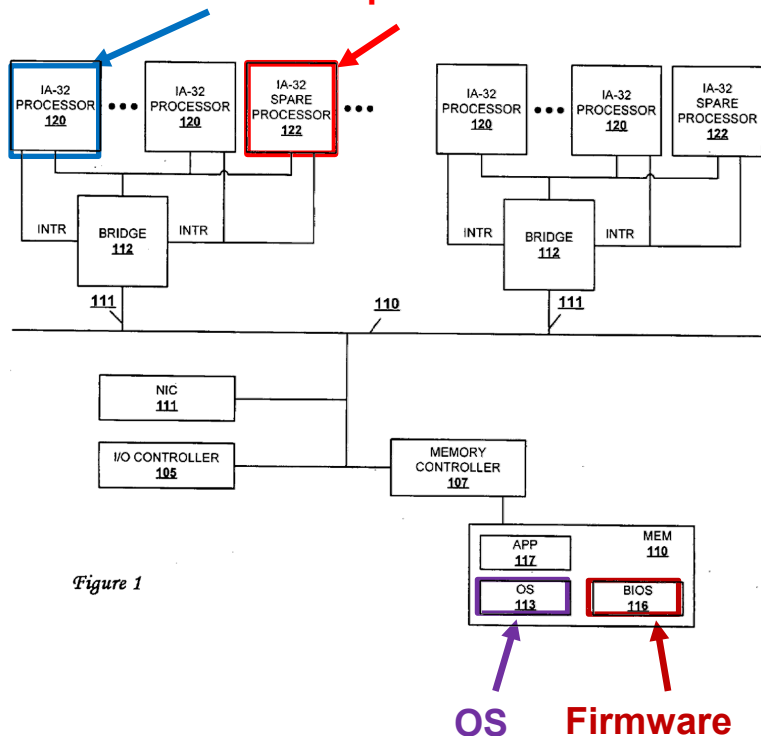


Figure 1

Landry

Logical CPU0 (boot processor) designation during system initialization

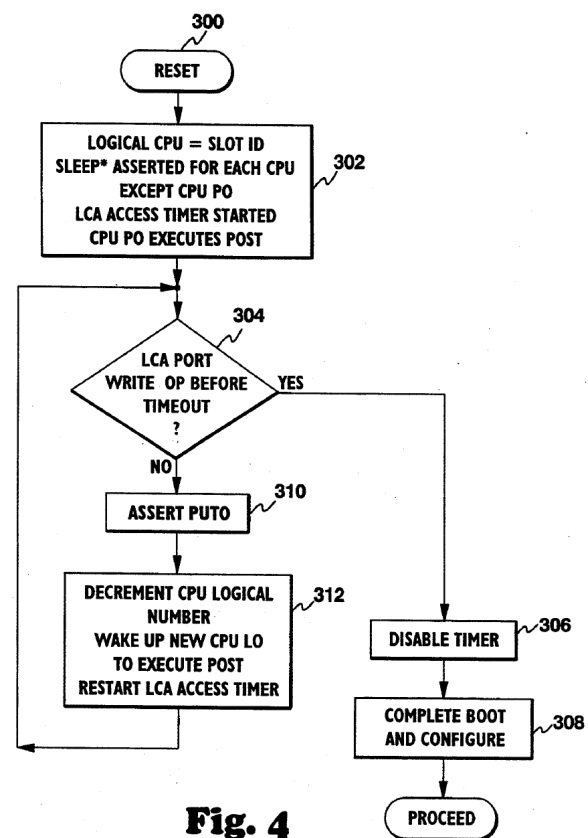


Fig. 4

IPR2024-01346: Pet. 33-37; Brogioli Decl. ¶¶47, 69-71, 144-158; Landry (Ex. 1011), 1:6-10, 1:47-62, 2:9-41, 6:44-7:52, Figs. 1, 4.

IPR2024-01347: Pet. 30-36; Brogioli Decl. ¶¶48, 70-72, 129-145; Landry (Ex. 1011), 1:6-10, 1:47-62, 2:9-41, 6:44-7:52, Figs. 1, 4.

Ground 2: Fox in View of Landry Teaches a Boot Processor

Patent Owner's View

“Petitioner’s arguments fail for the same reasons they failed for this limitation in Ground 1.”

“Landry’s boot processor selection happens during system initialization, i.e., before the operating system is running.”

Petitioner’s Obviousness Combination

- Landry’s process would have been a suitable option for selecting a boot processor in Fox’s IA-32 system
 - improve system availability by allowing the system to power up despite initially starting on a non-functional processor
 - allows for transparent replacement
 - known and conventional identification
- A POSITA would have found it obvious to switch Fox-Landry’s logical CPU0 (boot processor) to the spare processor in response to detecting loss of lockstep in the logical CPU0 in view of Safford

5. On or before October 25, 2004, it was known that a boot processor of a multiprocessor system could be identified as logical CPU0. Admitted.

Not disputed

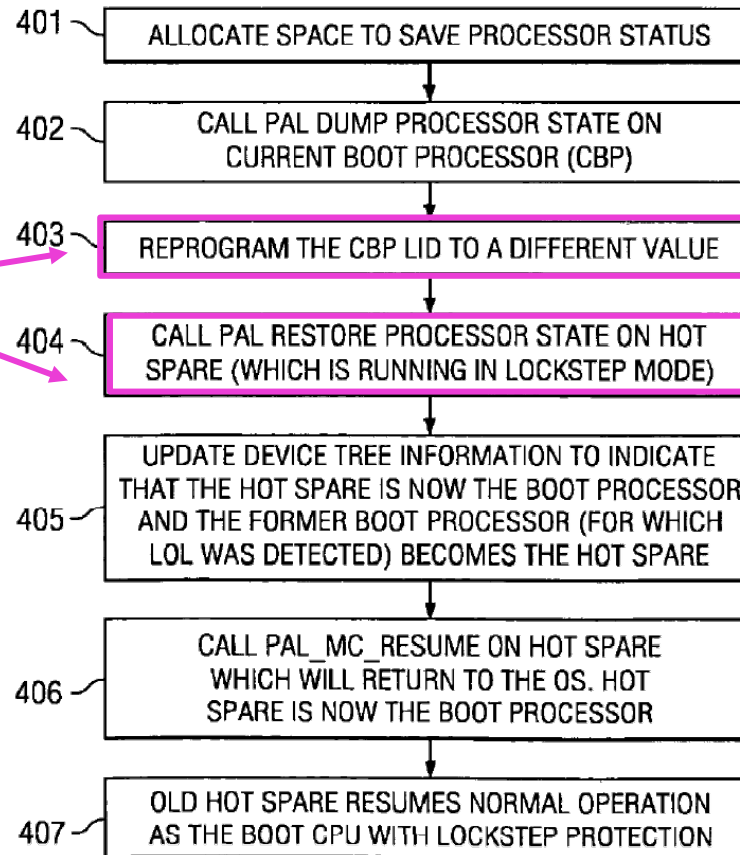
IPR2024-01346: Pet. 33-37; POR 17; Brogioli Decl. ¶¶47, 69-71, 144-158; Landry, 1:6-10, 1:47-62, 2:9-41, 6:44-7:52.
IPR2024-01347: Pet. 30-36; POR 34-35; Brogioli Decl. ¶¶48, 70-72, 129-145; Landry, 1:6-10, 1:47-62, 2:9-41, 6:44-

Two Core Disputed Issues (All Claims)

No.	Issue	
1.	Whether the prior art teaches a boot processor	Yes
2.	Whether the prior art teaches switching the role of the boot processor to the spare processor during system runtime	

Switching the Role of the Boot Processor to the Spare Processor During System Runtime

FIG. 4

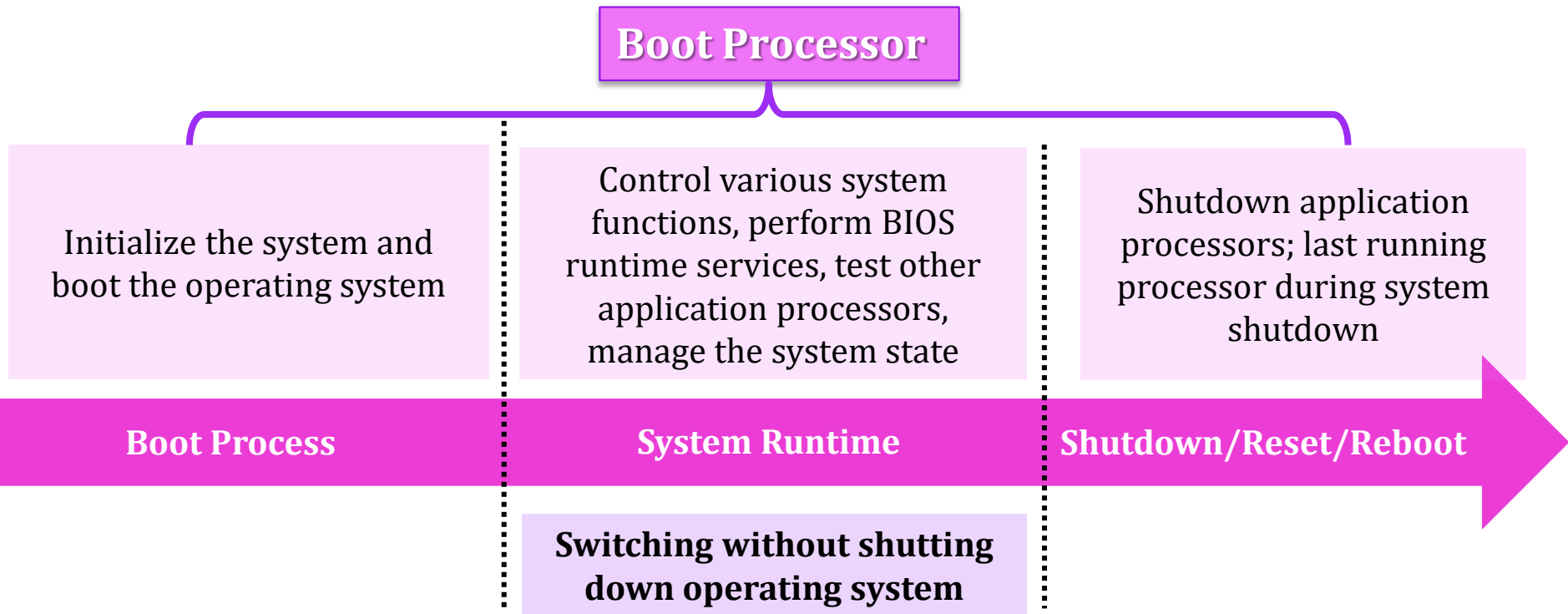


Switching the role of the boot processor without shutting down operating system

IPR2024-01346: '781 Patent, 4:54-62, 12:59-65, 13:20-14:3, 16:31-61, 17:34-55.
IPR2024-01347: '302 Patent, 5:44-48, 12:60-66, 13:21-14:3, 14:35-65, 16:1-17:3, Fig. 4.

Switching the Role of the Boot Processor to the Spare Processor During System Runtime

A POSITA would have understood that a boot processor does not stop being a boot processor after system initialization

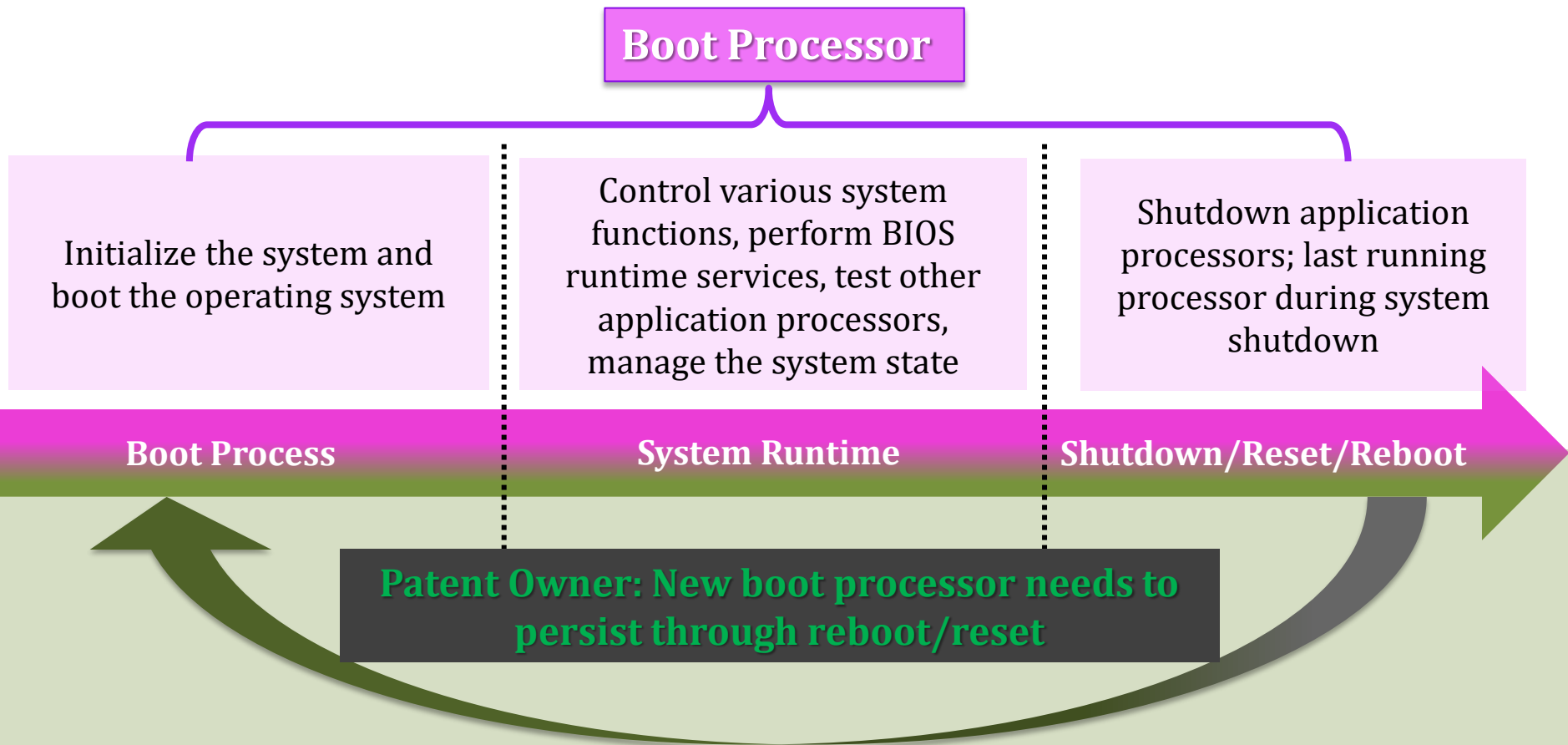


IPR2024-01346: Reply 3; Brogioli Decl. ¶¶44-47; MP Specification, 33, 75, 79.

IPR2024-01347: Reply 3; Brogioli Decl. ¶¶45-48; MP Specification, 33, 75, 79.

Switching the Role of the Boot Processor to the Spare Processor During System Runtime

A POSITA would have understood that a boot processor does not stop being a boot processor after system initialization



IPR2024-01346: POR 9-11, 13-14; Reply 2-8; Brogioli Decl. ¶¶44-47; MP Specification, 33, 75, 79.
IPR2024-01347: POR 11-12, 25-31; Reply 2-7; Brogioli Decl. ¶¶45-48; MP Specification, 33, 75, 79.

Patent Owner Misconstrues Switching the Role of the Boot Processor During System Runtime

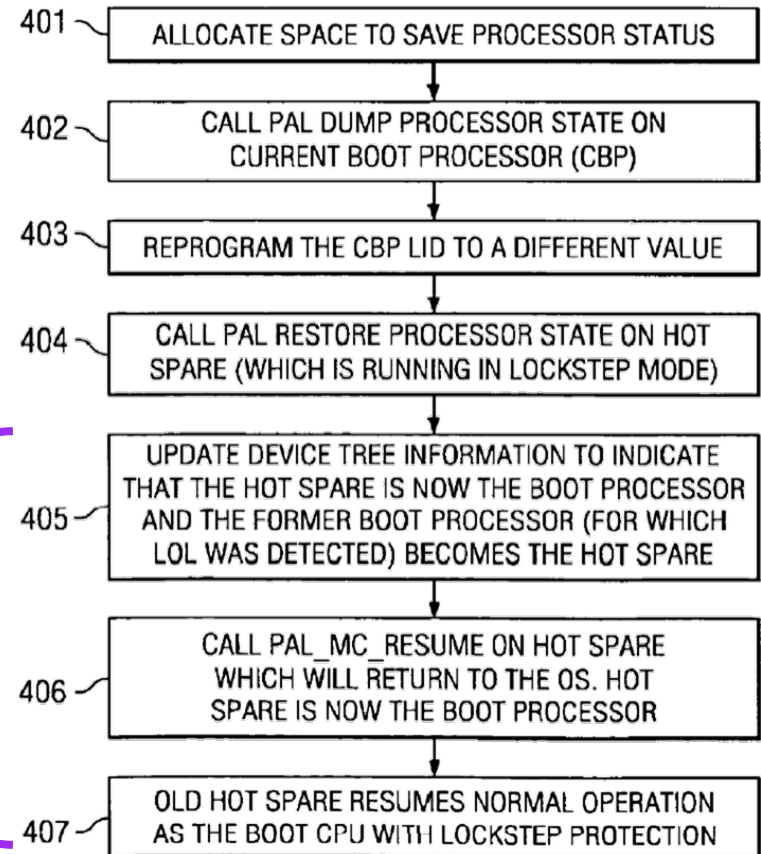
Patent Owner's View

The “switching the role” requires new boot processor identification persists through reboot/reset so that the new boot processor performs initialization after reboot/reset

Unfounded

Evidence in the Record

Hot spare processor becomes the new boot processor *during system runtime* without system reset or reboot



IPR2024-01346: POR 9-11, 13-14; Reply 2-8; Brogioli Decl. ¶¶44-47.

IPR2024-01347: POR 11-12, 25-31; Reply 2-7; Brogioli Decl. ¶¶45-48; '302 Patent, Fig. 4.

Patent Owner Misconstrues Switching the Role of the Boot Processor During System Runtime

Patent Owner's View

The '302/'781 Patent describes "a different approach wherein, for example, information regarding the **boot device path** can be changed during runtime and that is then persistently stored 'for the next time the system reboots.'"

Unfounded

Evidence in the Record

Changing boot device path ≠ Switching boot processor

55 SAL_PCI_CONFIG_WRITE); (e) EFI may be invoked to change the boot device path for the next time the system reboots (SetVariable); (f) EFI may be invoked to change the

make it available for the OS 11. When main memory is initialized and functional, the firmware 18_A is copied into the
15 main memory. Then, control is passed to EFI (not shown), which is responsible for activating boot devices, which typically includes the disk. The EFI reads the disk to load a program into memory, typically referred to as an operating system loader. The EFI loads the OS loader into memory, and
20 then passes it control of system 10_A by branching the boot processor into the entry point of such OS loader program.

IPR2024-01346: POR 13-14; Reply 7-8; Jones Decl. ¶¶37; Brogioli Decl. ¶¶44-47; '781 Patent, 12:8-20, 12:54-56.
IPR2024-01347: POR 27; Reply 6-7; Jones Decl. ¶¶37; Brogioli Decl. ¶¶45-48; '302 Patent, 12:9-21, 12:55-57.

Two Core Disputed Issues (All Claims)

No.	Issue	
1.	Whether the prior art teaches a boot processor	Yes
2.	Whether the prior art teaches switching the role of the boot processor to the spare processor during system runtime	Yes

Two Additional Issues (Some Dependent Claims)

No.	Issue
3.	Whether the prior art teaches a “non-volatile data storage” for storing information assigning a boot processor
4.	Whether the prior art teaches an “active spare”

Fox in View of Arai Teaches Storing Processor Information in a “Non-Volatile Data Storage”

Fox

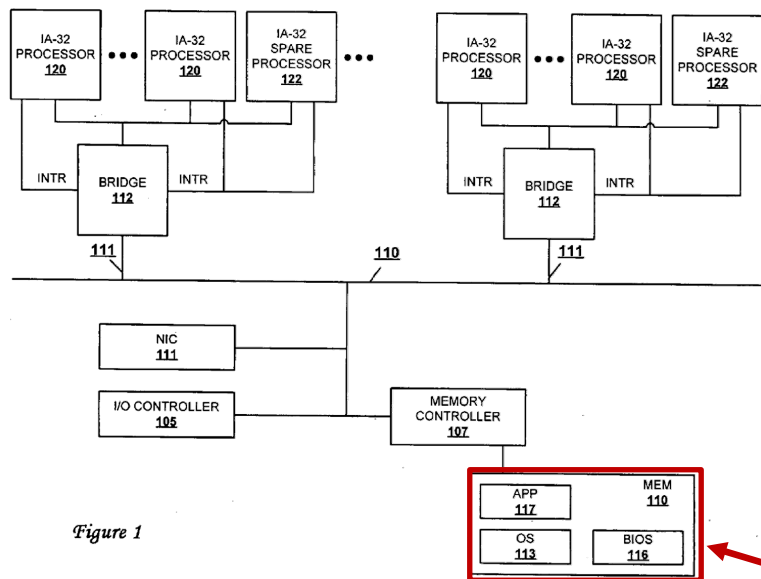


Figure 1

Not disputed

Arai

has logical CPU₃ (92) and logical CPU₇ (94). The BIOS creates a table in the main memory (RAM) during start-up. In one embodiment, the table is identified as an ACPI (Advanced Configuration and Power Interface) MAPIC (Multiple Advanced Programmable Interrupt Controller) table, where this table is a structure used by ACPI to describe

Arai discloses storing a processor table in the main memory

Fox discloses system memory that is a “non-volatile data storage”

Fox in View of Arai Teaches Storing Processor Information in a “Non-Volatile Data Storage”

Patent Owner's View

The alleged “data storage” in the Fox-Arai combination is RAM.

Contradicted

Petitioner's Combination

198. As discussed in Grounds 1-2, limitation [1b], a POSITA would have found it suitable, routine, and ordinary to store Fox's register or table that stores information assigning the roles of boot and spare processors in its system memory (“data storage”), like in Arai. A POSITA would have understood Fox's system memory would be “non-volatile” because it was common and well known that the BIOS would be embedded in a non-volatile memory (e.g., ROM, EEPROM, etc.). *See id.*, [0028], FIG. 1; MP Specification, 38 (teaching storing MP configuration table within the BIOS read-only memory space).

IPR2024-01346: Pet. 22-24, 50; POR 11-14; Jones Decl. ¶¶36, 55; Reply 14-15; Brogioli Decl. ¶¶52, 120-124, 198; '781 Patent, 4:54-58, 8:43-48, 13:20-14:3, 17:37-40.
IPR2024-01347: Pet. 59-62; POR 47; Jones Decl. ¶36; Reply 19-20; Brogioli Decl. ¶¶207-214; '302 Patent, 9:4-15, 12:60-66, 13:21-44.

Fox in View of Arai Teaches Storing Processor Information in a “Non-Volatile Data Storage”

Patent Owner's View

“A POSITA would also understand that information identifying a boot processor or a spare processor would not be stored in memory that could not be written to, such as ROM ... The '302 Patent discloses and claims transferring or switching the role of boot processor, something which could not be done with memory that cannot be written to.”

Evidence in the Record

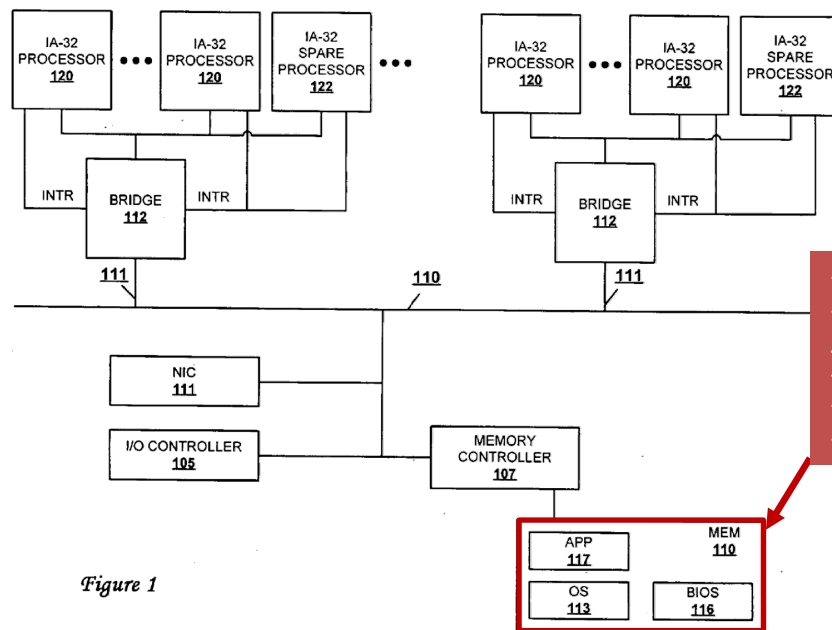


Figure 1

Fox's system memory is not Read-Only

IPR2024-01346: Pet. 22-24, 50; POR, 11-14; Jones Decl. ¶¶36, 55; Reply 14-15; Brogioli Decl. ¶¶52, 120-124, 198; Fox, [0028], Fig. 1.

IPR2024-01347: Pet. 59-62; POR, 47; Jones Decl. ¶36; Reply 19-20; Brogioli Decl. ¶¶207-214; Fox, [0028], Fig. 1.

Fox in View of Arai Teaches Storing Processor Information in a “Non-Volatile Data Storage”

Patent Owner’s View

“RAM is volatile memory.”

NVRAM in the ’781 Patent is the “non-volatile data storage.”

Unfounded

Evidence in the Record

55 In certain embodiments, information (referred to herein as a “device tree”) is stored to the multi-processor system, wherein such information identifies the processor in the system that is the boot processor and identifies the processor that is the spare for the boot processor. The information is stored

60 A “device tree” is provided, which is shown as device tree 203 in this example. Device tree 203 is stored in SRAM (Scratch RAM) on the cell, which is RAM that is reinitialized.

TABLE 1

Device	Status	Lockstep Enabled	Boot Processor?	Spare for Boot Processor?
Processor A	Present, Enabled, and Functioning	Yes	Yes	No
Processor B	N/A	Yes	No	Yes
Processor C	Present, Enabled, and Functioning	Yes	No	No
...	...	...	...	...

IPR2024-01346: Pet. 22-24, 50; POR 11-14; Jones Decl. ¶¶36, 55; Reply 14-15; Brogioli Decl. ¶¶52, 120-124, 198; ’781 Patent, 4:54-58, 8:43-48, 12:59-65, 13:20-14:3, 17:37-40.
IPR2024-01347: Pet. 59-62; POR 47; Jones Decl. ¶36; Reply 19-20; Brogioli Decl. ¶¶207-214; ’302 Patent, 5:40-44, 9:4-15, 12:60-66, 13:21-44.

Two Additional Issues (Some Dependent Claims)

No.	Issue	
3.	Whether the prior art teaches a “non-volatile data storage” for storing information assigning a boot processor	Yes
4.	Whether the prior art teaches an “active spare”	

Fox Teaches an “Active Spare”

Patent Owner’s View

“[T]he disclosure in Fox relied upon by Petitioner is still performed during system initialization.”

“Fox only teaches a hot-spare processor” that “remains hidden until one of the primary processors fails.”

Contradicted

Evidence in the Record

- Fox’s enabling the “hot spare processor” “on-the-fly” means the “hot spare processor” is configured dynamically as needed while the operating system is running.

[0035] In one embodiment, during boot, BIOS utilizes advanced configuration and power interface (ACPI) specification and/or the S3 state to put the processor in a low-power, standby, or “off” state. The S3 state is a computer sleep state defined by the ACPI specification in which the system is not powered nor running, but the memory remains intact, powered and in a continuous refresh cycle. This initial removal of the spare processor from the group of executing processors occurs before the BIOS hands over control to the OS. In another embodiment the system is made to enable the “hot spare processor” feature on-the-fly by the OS with a driver-OS-SMI handshake. Tracking which processors are active may be completed by an software table of active processors, which is available to be read by the OS, when determining work load allocation to each of the active processors.

IPR2024-01346: Pet. 50-52, 56-57; POR 14-16; Reply 16-18; Brogioli Decl. ¶¶201-203, 217-218; Fox, [0015], [0034]-[0035].

IPR2024-01347: Pet. 56-58; POR 38-44; Reply 16-19; Brogioli Decl. ¶¶199-204; Fox, [0015], [0034]-[0035].

Fox Teaches an “Active Spare”

Patent Owner’s View

“Fox only teaches a hot-spare processor.”

“Fox only teaches idling a failing processor.”

Contradicted

Petitioner’s Combination

- Fox discloses or renders obvious an “active spare”
 - Improves the processing capacity of the system when no processor failure occurs
 - The active spare processor would be initialized like other active processors in the system
 - When a processor failure is detected, the state of the active spare would have been set to idle to be switched with the failing processor, using software tools as taught in Fox and known in the art

IPR2024-01346: Pet. 50-52, 56-57; POR 14-16; Reply 16-18; Brogioli Decl. ¶¶201-203, 217-218; Fox, [0015], [0034]-[0035].

IPR2024-01347: Pet. 56-58; POR 38-44; Reply 16-19; Brogioli Decl. ¶¶199-204; Fox, [0015], [0034]-[0035].

Two Additional Issues (Some Dependent Claims)

No.	Issue	
3.	Whether the prior art teaches a “non-volatile data storage” for storing information assigning a boot processor	Yes
4.	Whether the prior art teaches an “active spare”	Yes

Summary of Disputed Issues

No.	Issue	Petitioner's Showing Supported by Evidence
1	Whether the prior art teaches a boot processor	1. A POSITA has background knowledge of a boot processor and IA-32 multiprocessor systems 2. A POSITA would have understood Fox discloses or suggests a boot processor 3. A POSITA would have understood Fox stores information assigning a boot processor 4. Patent Owner's expert failed to consider all relevant evidence in the record 5. Patent Owner's expert failed to opine from the perspective of a POSITA 6. Patent Owner improperly focused on explicit teaching and failed to address Petitioner's obviousness combination
2	Whether the prior art teaches switching the role of the boot processor to the spare processor during system runtime	1. The patents disclose switching the role of the boot processor without shutting down the operating system (during system runtime) 2. The prior art teaches switching the role of the boot processor during system runtime as described in the patents 3. Patent Owner's implicit construction of switching the role of the boot processor as requiring the new boot processor to perform system initiation after reboot/reset lacks basis
3	Whether the prior art teaches a "non-volatile data storage" for storing information assigning a boot processor	1. No dispute that Fox's system memory is non-volatile 2. The prior art teaches storing information assigning a boot processor in Fox's system memory 3. Patent Owner mischaracterizes Petitioner's combination as storing information in RAM 4. Patent Owner's argument that Fox's system memory is read-only is unfounded
4	Whether the prior art teaches an "active spare"	1. Fox's alternative embodiment teaches an "active spare" 2. Patent Owner mischaracterizes Fox's alternative embodiment 3. Fox's alternative embodiment teaches idling the "active spare" to dynamically configure the hot spare as needed 4. Patent Owner failed to address Petitioner's obviousness combination