

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

APPLE, INC.,
Petitioners,

v.

REALTIME DATA LLC,
Patent Owner.

Case IPR2016-01366
Patent 8,090,936 B2

Before DEBRA K. STEPHENS, GEORGIANNA W. BRADEN, and
JASON J. CHUNG, *Administrative Patent Judges*.

STEPHENS, *Administrative Patent Judge*.

FINAL WRITTEN DECISION
35 U.S.C. § 318 and 37 C.F.R. § 42.73

INTRODUCTION

We have jurisdiction to hear this *inter partes* review under 35 U.S.C. § 6, and this Final Written Decision is issued pursuant to 35 U.S.C. § 318(a) and 37 C.F.R. § 42.73. For the reasons that follow, we determine that Petitioner has shown by a preponderance of the evidence that claims 1–24 of U.S. Patent No. 8,090,936 B2 (Ex. 1001, “the ’936 patent”) are unpatentable.

I. BACKGROUND

A. Procedural History

Apple Inc. (Petitioner) filed a Petition (Paper 2, “Pet.”) to institute an *inter partes* review of claims 1–24 of the ’936 patent. Realtime Data, LLC (“Patent Owner”) filed a Preliminary Response (Paper 9, “Prelim. Resp.”).

Petitioner challenged claims 1–24 of the ’936 patent on the following grounds:

Claims	Basis	References ¹
1–24	§ 103 (a)	Sukegawa and Dye
1–24	§ 103 (a)	Sukegawa, Dye, and Settsu
1–24	§ 103 (a)	Sukegawa, Dye, and Burrows
1–24	§ 103 (a)	Sukegawa, Dye, Settsu, and Burrows

¹

Reference	Patent Number	Exhibit
Sukegawa	US 5,860,083 (issued Jan. 12, 1999)	1005
Dye	US 6,145,069 (filed Apr. 26, 1999)	1008

Petitioner also relied on the Declaration of Dr. Charles J. Neuhauser (Ex. 1003) to support its challenges.

Pursuant to 35 U.S.C. § 314(a), we instituted an *inter partes* review of:

Claims	Basis	References ²
1–24	§ 103 (a)	Sukegawa and Dye
1–10 and 18–24	§ 103 (a)	Sukegawa, Dye, and Settsu
1–10	§ 103 (a)	Sukegawa, Dye, and Burrows
1–10	§ 103 (a)	Sukegawa, Dye, Settsu, and Burrows

(Paper 10, “Dec. to Inst.”).

After institution of trial, Patent Owner filed a Patent Owner Response (Paper 13, “PO Resp.”), to which Petitioner filed a Reply (Paper 15, “Reply”). An oral argument was held on September 20, 2017. A transcript of the oral argument is included in the record (Paper 24 (“Tr.”)).

Settsu	US 6,374,353 B1(filed Mar. 3, 1999)	1006
--------	-------------------------------------	------

Michael Burrows et al., *On-line Data Compression in a Log-structured File System* (1992) (hereinafter “Burrows”) (Exhibit 1007).

2

Reference	Patent Number	Exhibit
Sukegawa	US 5,860,083 (issued Jan. 12, 1999)	1005
Dye	US 6,145,069 (filed Apr. 26, 1999)	1008
Settsu	US 6,374,353 B1(filed Mar. 3, 1999)	1006

Michael Burrows et al., *On-line Data Compression in a Log-structured File System* (1992) (hereinafter “Burrows”) (Ex. 1007).

B. Related Matters

The parties identify the following cases as related to the challenged patent: *Realtime Data, LLC d/b/a IXO v. Microsoft Corporation*, Case No. 4:14-cv-00827 (E.D. Tex.) and *Realtime Data, LLC d/b/a IXO v. Apple, Inc.*, Case No. 3:16-cv02595 (N.D. Cal.) (transferred from *Realtime Data, LLC v. Apple, Inc.*, Case No. 6:15-cv-00885 (E.D. Tex.)) (Pet. 1; Paper 20, 2).

C. The '936 Patent (Ex. 1001)

The '936 patent, entitled “Systems and Methods for Accelerated Loading of Operating Systems and Application Programs,” relates to “providing accelerated loading of operating system and application programs upon system boot or application launch,” and to the use of data compression and decompression techniques for such purpose (Ex. 1001, Title, 1:16–23). The Specification discusses the limits of prior art storage devices, particularly the significant bandwidth limitations of “mass storage devices,” such as hard disk drives and their “inherent unreliability” (*id.* at 1:40–54, 2:38–49).

The Specification of the '936 patent is directed to “data storage controllers that provide increased data storage/retrieval rates that are not otherwise achievable using conventional disk controller systems and protocols to store/retrieve data to/from mass storage devices” (*id.* at 5:25–29). According to the Specification, “accelerated data storage/retrieval mitigates the traditional bottleneck associated with, e.g., local and network disk accesses” (*id.* at 5:56–59). The Specification describes that

“accelerated” data storage comprises receiving a digital data stream at a data transmission rate which is greater than the data storage rate of a target storage device, compressing the input stream at a compression rate that increases the effective data storage rate of the target storage device and storing the

compressed data in the target storage device

(Ex. 1001, 5:38–43) and further describes that

accelerated data retrieval comprises retrieving a compressed digital data stream from a target storage device at the rate equal to, e.g., the data access rate of the target storage device and then decompressing the compressed data at a rate that increases the effective data access rate of the target storage device

(*id.* at 5:51–56).

Figure 1, “a block diagram of a data storage controller according to one embodiment of the present invention” (*id.* at 4:33–34), is reproduced below.

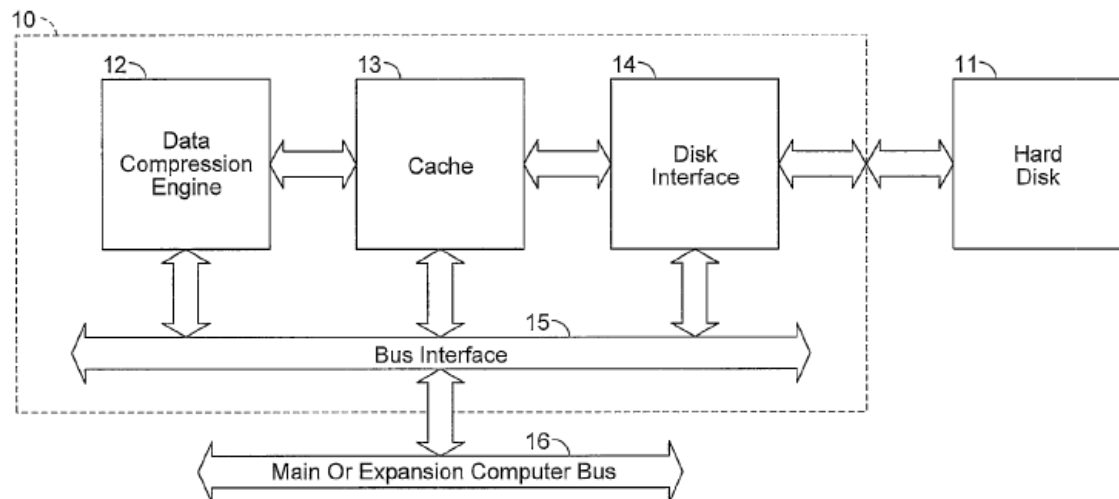


Figure 1 illustrates an embodiment of data storage controller 10 (*id.* at Fig. 1, 5:60–62). Data storage controller 10 includes data compression engine (DCE) 12 which compresses/decompresses data stored/retrieved from a mass storage unit such as hard disk 11, to provide accelerated data storage/retrieval (*id.* at 5:62–66). Data storage controller 10 also includes cache 13, disk interface (or disk controller) 14, and bus interface 15 (*id.* at 6:12–14). As shown in Figure 1, data storage controller 10 is “operatively connected” to main or expansion bus 16 and hard disk 11 (*id.* at 6:14–17).

According to the '936 patent, “[w]hen data is read from disk by the host computer, data flows from the disk **11** through the data storage controller **10** to the host computer” (Ex. 1001, 6:51–54). When disk data is read, data is transferred from hard disk 11 to cache 13 without intervention of DCE 12 (*id.* at 6:59–61). Once in cache 13, DCE 12 then reads and decompresses data, returning the decompressed data to cache 13 (*id.* at 7:12–16). The decompressed data is next transferred to main or expansion computer bus 16 via bus interface 15 (*id.* at 7:16–18).

Similarly, according to the '936 patent, when data is written to hard disk 11 from the host computer, data flows from the host computer via main computer bus 16 to data storage controller 10, where the data is compressed, and onto hard disk 11 (*id.* at 7:30–32).

The '936 further describes

upon host computer power-up or external user reset, the data storage controller 10 initializes the onboard interfaces 14, [1]5 prior to release of the external host bus 16 from reset. The processor of the host computer then requests initial data from the disk 11 to facilitate the computer's boot-up sequence

(*id.* at 7:56–61). Specifically, the '936 discloses the data storage controller described, “may employ a technique of data preloading to decrease the computer system boot time” (*id.* at 21:15–17). “[P]rior to host system reset, the data storage controller [10] can proceed to [preload] the portions of the computer operating system from the boot device (e.g., hard disk [11]) into the on-board cache memory [13]” (*id.* at 21:23–26).

Figures 7A and 7B “comprise a flow diagram of a method for providing accelerated loading of an operating system and/or application

programs upon system boot, according to one aspect of the present invention” (*id.* at 4:46–49).

Figure 7A is reproduced below:

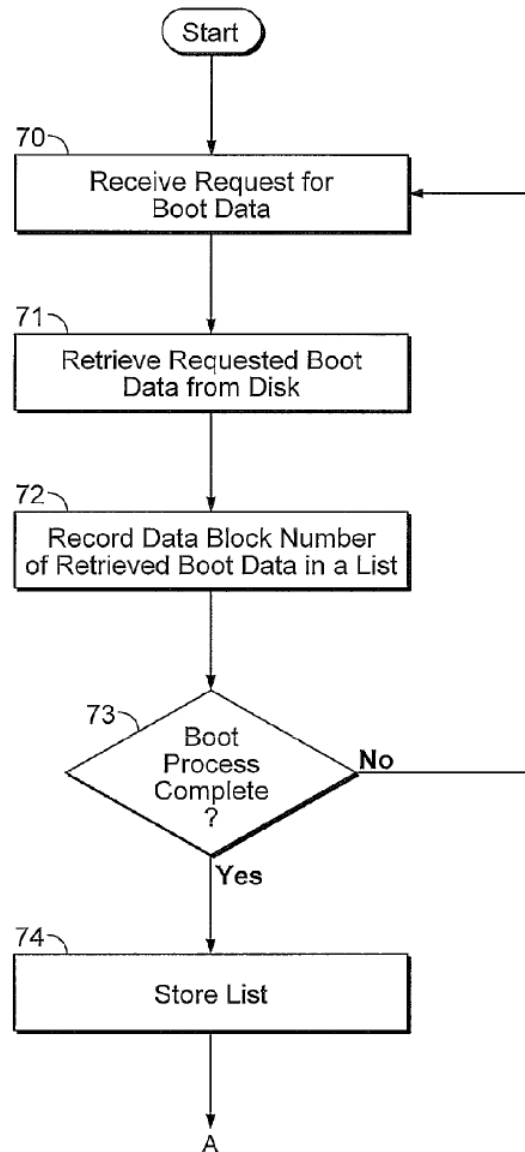


FIG. 7A

Figure 7A illustrates the steps performed during the computer boot process (Ex. 1001, 21:62–63). As illustrated in Figure 7A, during the computer boot process, data storage controller 10, receives a request for boot data (step 70) (*id.* at 21:62–64). Data storage controller 10 retrieves the requested boot

data from hard disk 11 from cache 13 (step 71) (*id.* at 21:64–67). Each requested data block is recorded in a list (step 72) until the boot process is complete (step 73), at which time, the list will be stored on the boot device (hard disk 11) (step 74) (*id.* at 21:67–22:7).

Figure 7B is reproduced below:

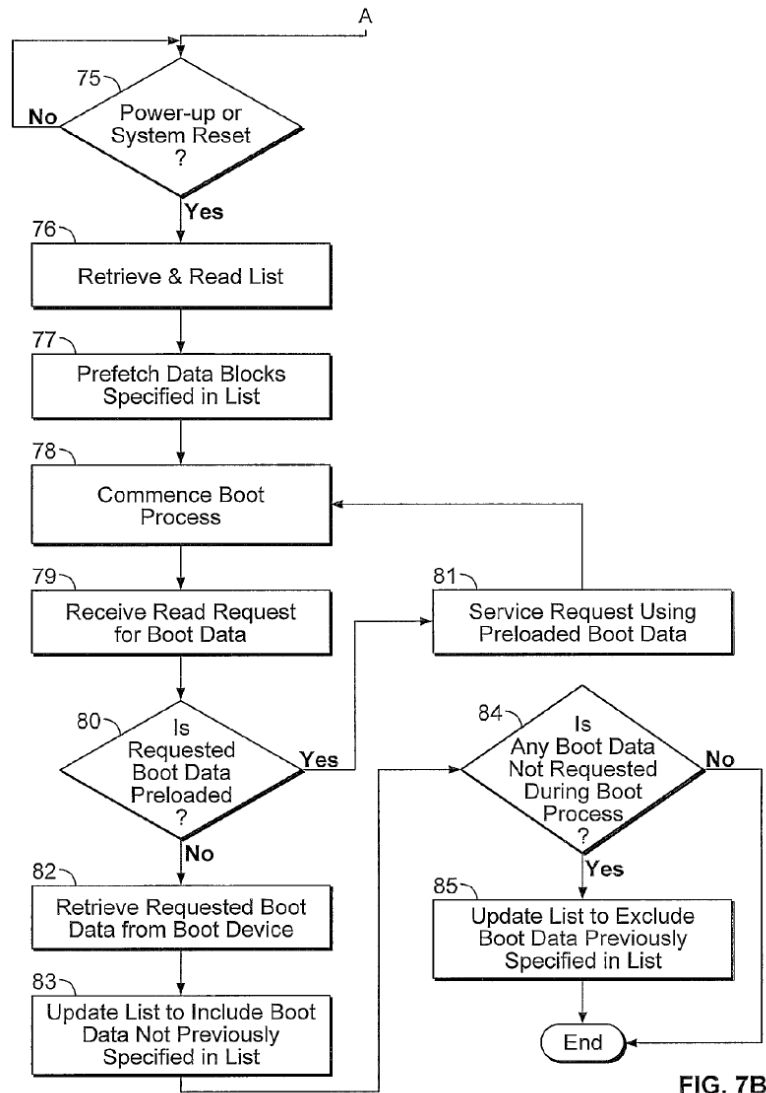


Figure 7B illustrates the steps which occur upon each subsequent power-on/reset (Ex. 1001, 11:8–9). As shown in Figure 7B of the '936 patent,

when power-up or system reset occurs (step 75), data storage controller 10 retrieve and reads the stored list (step 76) and “proceed[s] to preload the boot data specified on the list” into cache 13 (step 77) (*id.* at 8–13).

“[D]epending on the resources of the given system (e.g., memory, etc.), the preloading process may be completed prior to commencement of the boot process, or continued after the boot process begins (in which case booting and preloading are performed simultaneously” (*id.* at 22:14–17).

As described further, the boot process commences (step 78), the read request for boot data is received (step 79), and if the requested boot data is preloaded (step 80), the request is serviced using the preloaded boot data (step 81) (*id.* at 22:18–24).

D. Exemplary Claim

We instituted claims 1–24 of the ’936 Patent (Dec. to Inst. 16). The ’936 patent has three independent claims, claims 1, 11, and 18; the remaining claims are dependent (Ex. 1001, Claims), all of which are part of this proceeding. Claims 1 and 18 of the ’936 patent are exemplary of the claims at issue:

1. A method comprising:
 - maintaining a list of boot data used for booting a computer system, wherein at least a portion of said boot data is compressed by a data compression engine to provide said at least a portion of said boot data in compressed form, and stored in compressed form on a boot device;
 - initializing a central processing unit of said computer system;
 - preloading said at least a portion of said boot data in compressed form from said boot device to a memory;
 - accessing and decompressing said at least a portion of said boot

data in said compressed form from said memory; and

utilizing said decompressed at least a portion of said boot data to boot said computer system, wherein said at least a portion of said boot data is decompressed by said data compression engine.

18. A method of preloading an operating system for booting a computer system comprising:

storing substantially all of the operating system in compressed form on a boot device;

preloading a first portion of the substantially all of the operating system from said boot device to a memory;

accessing and decompressing the first portion from the memory using a data compression engine;

utilizing the decompressed portion to partially boot said computer system;

responsive to a request, locating a second portion of the substantially all of the operating system using a boot data list and preloading the second portion from the boot device to the memory;

accessing and decompressing the second portion from the memory using the data compression engine; and

utilizing the decompressed second portion to further partially boot said computer system.

II. ANALYSIS

A. Level of Ordinary Skill in the Art

The level of skill in the art is a factual determination that provides a primary guarantee of objectivity in an obviousness analysis (*Al-Site Corp. v. VSI Int'l Inc.*, 174 F.3d 1308, 1324 (Fed. Cir. 1999) (citing *Graham v. John*

Deere Co., 383 U.S. 1, 17–18 (1966); *Ryko Mfg. Co. v. Nu-Star, Inc.*, 950 F.2d 714, 718 (Fed. Cir. 1991))).

Petitioner asserts, through its declarant Dr. Neuhauser, one of ordinary skill would be a person with a Bachelor’s Degree in electrical engineering, computer engineering, or a related area of study. In addition, this person would have between three and five years of practical experience in the design and implementation of computer systems, such as personal computers. Alternatively, a person with a Master’s Degree in the area of electrical engineering, computer engineering, or a related area of study and somewhat less practical experience would be similarly qualified (Ex. 1003 ¶ 15). Patent Owner does not appear to dispute the educational level or experiential aspects of Petitioner’s definition, and indeed, Patent Owner’s declarant, Dr. Back, agrees with Dr. Neuhauser as to the level of ordinary skill in the art (Ex. 2003 ¶ 41).

We agree with and adopt Petitioner’s assessment of a level of skill in the art. We note also that the applied prior art reflects the appropriate level of skill at the time of the claimed invention (*see Okajima v. Bourdeau*, 261 F.3d 1350, 1355 (Fed. Cir. 2001)).

B. Claim Construction

In an *inter partes* review, the Board interprets claim terms in an unexpired patent according to the broadest reasonable construction in light of the specification of the patent in which they appear (37 C.F.R. § 42.100(b); *Cuozzo Speed Techs., LLC v. Lee*, 136 S. Ct. 2131, 2142 (2016)). Under that standard, and absent any special definitions, we give claim terms their ordinary and customary meaning, as would be understood by one of ordinary skill in the art at the time of the invention (*In re Translogic Tech., Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007)). Additionally,

any special definitions for claim terms must be set forth with reasonable clarity, deliberateness, and precision (*In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994)).

In an *inter partes* review, claim terms in an unexpired patent are interpreted according to their broadest reasonable construction in light of the specification of the patent in which they appear (37 C.F.R. § 42.100(b); see *Cuozzo Speed Techs., LLC v. Lee*, 136 S. Ct. 2131, 2144–46 (2016) (“We conclude that the regulation represents a reasonable exercise of the rulemaking authority that Congress delegated to the Patent Office.”)).

Under that standard, and absent any special definitions, we give claim terms their ordinary and customary meaning, as would be understood by one of ordinary skill in the art at the time of the invention (*In re Translogic Tech., Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007)). An inventor may provide a meaning for a term that is different from its ordinary meaning by defining the term in the specification with “reasonable clarity, deliberateness, and precision.” (*In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994)).

Limitations, however, are not to be read from the specification into the claims (*In re Van Geuns*, 988 F.2d 1181, 1184 (Fed. Cir. 1993)). In addition, the Board may not “construe claims during [an *inter partes* review] so broadly that its constructions are unreasonable under general claim construction principles” (*Microsoft Corp. v. Proxyconn, Inc.*, 789 F.3d 1292, 1298 (Fed. Cir. 2015)).

Petitioner indicates the broadest reasonable construction should be applied (Pet. 19; Pet. Reply 1). For purposes of our analysis, we need only construe the terms “preloading” and “encoder.”

1. “preloading”

Initially, we note the terms “preloading” and “preload” are not defined explicitly in the ’936 Specification. Patent Owner urges us to interpret “preloading” as “transferring data from storage to memory in anticipation of immediate or near-in-time use” because “this construction is consistent with the claims and the intrinsic record, and is the broadest reasonable interpretation in light of the specification” (PO Resp. 13).

Claims 1 recites “preloading said at least a portion of said boot data in compressed form from said boot device to a memory” (Ex. 1001, Claims). The claim itself, however, does not recite any timing associated with the recited preloading. Thus, although Patent Owner contends claims 1, 11, 13, and 18 “recite that boot or application data is preloaded, for example, to be ‘utilized’ by the computer system immediately or in the near future,” (PO Resp. 14), we do not find that any of these claims recites that preloaded boot or application data must be “utilized by the computer system *immediately or in the near future.*”

Relying on Dr. Back’s declaration, Patent Owner asserts a person of ordinary skill in the art (POSTIA) “would have understood that this movement of data from storage into memory is performed in anticipation of immediate or near-in-time use” (PO Resp. 14 (citing Black Dec. ¶¶ 47–49)). Dr. Back asserts “[b]ased on the specification, a POSITA would have also understood that this movement of data from storage into memory is performed in anticipation of immediate or near-in-time use” (Ex. 2003 ¶50). Dr. Back contends this interpretation “reflects the common understanding of this term in the field” (Ex. 1017, 32).

Patent Owner further contends the Specification supports its interpretation (*id.*). For example, Patent Owner identifies the Abstract as supporting this interpretation (*id.* at 15). The Abstract, however, discloses “the boot data is retrieved from a boot device and *stored* in a cache memory device” (Ex. 1001, Abstract (emphasis added)). Patent Owner also directs our attention to columns 21 and 22 of the Specification as supporting the interpretation of “preloading” (PO Resp. 15). Yet throughout these descriptions, timing requirements between preloading and utilizing are not described—timing requirements are missing from the claims and not used in describing the meaning of “preloading.” Indeed, this description does not discuss any timing criteria between the preloading of the data and utilizing the data. The ’936 patent discloses

[f]urther, *prior to host system reset*, the data storage controller can proceed to *preload* the portions of the computer operating system from the boot device (e.g., hard disk) into the on-board cache memory. The data storage controller *preloads* the needed sectors of data in the order in which they will be needed. Since the same portions of the operating system must be loaded upon each boot process, it is advantageous for the boot device controller to *preload* such portions and not wait until it is commanded to load the operating system.
(Ex. 1001, 21:23–31).

Notably, Patent Owner contends “[c]ertain embodiments . . . describe ‘preloading’ as loading data in anticipation of using that data” (PO Resp. 14 (emphasis added)), acknowledging other embodiments exist. An additional portion of the ’936 patent to which Dr. Back points, describes a technique for “one embodiment” (illustrated by the flow diagram of FIGS. 7a and 7b) that may be employed (Ex. 1001, 21:56–58; Ex. 2003 ¶ 52; PO Resp. 15).

Thus, Patent Owner admits only “certain embodiments” and “one embodiment” “load data in anticipation of using that data.”

Patent Owner additionally contends the description of Figure 7b uses the term “prefetching” in step 77, and describes it as “preloading” in the Specification; therefore, according to Patent Owner, the term “‘preloading’ has a meaning similar to ‘prefetching’ [which] . . . refers to the process of retrieving data before it is needed” (PO Resp. 15–16 (citing Ex. 1001, 22:8–13, 21:56–22:7, Fig 7b; Ex. 2003 ¶¶ 50–53)).

We do not agree with Patent Owner. At the outset, Figures 7a and 7b, relied upon by Patent Owner, are described in a non-limiting context using the nomenclature “*according to one aspect of the present invention*” (emphasis added). Ex. 1001, 4:46–49. In addition, we do not agree with Patent Owner that the ’936 patent is using “preloading” and “prefetching” as synonyms rather than using “prefetching” as a type of “preloading.” Regardless, even the extrinsic evidence of “prefetching” proffered by Patent Owner does not support Patent Owner’s interpretation of “preloading.” Specifically, “prefetching” is defined as “[i]n a pipelined process, to fetch the next instruction, or instruction part, before the processing unit requires it, resulting in a performance improvement by eliminating the lag between completion of one instruction and the availability of the next” (Ex. 2007, 3). This proffered definition, however, is directed to lines of code or instruction to be executed in a pipelined process, whereas the use of “prefetch” in the ’936 patent is used to describe preloading a *group* of instructions (boot data) to cache and does not describe this is in a pipelined process. Thus, we do not agree that the proffered definition of “prefetch” is applicable.

The prefix “pre” means “before,” but absent additional information, does not convey how long “before” is. As such, Patent Owner’s proposed interpretation is construed too narrowly. For example, the IEEE Standard Computer Dictionary defines “load” as “[t]o copy computer instructions or data from external storage to internal storage or from internal storage to registers” (Ex. 2006, 3–4) (*see also* Ex. 2005, 3; Ex. 2009, 4). “Preloading,” thus, would be copying computer instructions or data from external storage to internal storage or from internal storage to registers before an unspecified event occurs.

Patent Owner further contends the ’936 Specification “draws a clear distinction between the concepts of ‘storing’ and ‘loading,’ and ‘storing’ and ‘preloading’” (PO Resp. 19). Specifically, Patent Owner points to the disclosure that “[m]ass storage devices (such as a ‘hard disk’) typically *store* the operating system of a computer system, as well as applications and data” and “boot data is *stored* in a compressed format on the boot device” (PO Resp. 19 (citing Ex. 1001, 1:41–43 (emphasis added), 3:57–58 (emphasis added))), as support for their interpretation. According to Patent Owner, “storing” in the ’936 patent means “pre-installation” or “storing software or other data on a persistent storage medium” (*id.*).

We do not agree, however, that “storing” has the proffered interpretation. Indeed, the ’936 patent does not explicitly “storing” as meaning “pre-installation” or as requiring the software or other data is stored on a persistent storage medium. Rather, the ’936 patent describes “the boot data is retrieved from a boot device and *stored* in a cache memory” (Ex. 1001, Abstract). This boot data is also described as being “preloaded” into a cache memory (*id.* at Abstract, 21:23–26). As such, Patent Owner’s

proposed interpretation is construed too narrowly. Thus, although we agree “storing” and “loading” are distinct terms, we do not agree with Patent Owner that “store” means “pre-installation” or “storage on a persistent medium,” and somehow distinguishes “preloading” from “pre-installation.”

Patent Owner asserts “the ’936 Patent states that its systems and methods are designed to ‘provid[e] accelerated loading of operating system and application programs’” (PO Resp. 14 (citing Ex. 1001, 1:17–23, Abstract, 3:32–38)).

Although “accelerated loading” is not defined explicitly, the ’936 patent describes

[i]n general, as described in the above-incorporated applications, “accelerated” *data storage* comprises receiving a digital data stream at a data transmission rate which is greater than the data storage rate of a target storage device, compressing the input stream at a compression rate that increases the effective data storage rate of the target storage device and storing the compressed data in the target storage device.

(Ex. 1001, 5:36–43 (emphasis added)). Similarly, the ’936 patent describes “accelerated *data retrieval*” as “compris[ing] retrieving a compressed digital data stream from a target storage device at the rate equal to, e.g., the data access rate of the target storage device and then decompressing the compressed data at a rate that increases the effective data access rate of the target storage device” (*id.* at 51–59). Neither of these descriptions regarding discuss any timing criteria for the use of data stored into or retrieved from cache memory. In particular, the Specification describes

[t]he present invention relates generally to systems and methods for providing accelerated loading of operating system and application programs upon system boot or application launch and, more particularly, to data storage controllers employing

lossless and/or lossy data compression and decompression to provide accelerated loading of operating systems and application programs

(Ex. 1001, 1:17–23). The '936 patent's statement, "accelerated loading of operating system and application programs upon system boot or application launch," does not address movement of data from storage to memory in anticipation of *immediate or near-in-time use* (*id.*). Rather, the '936 patent states it employs "lossless and/or lossy data compression and decompression" to achieve accelerated loading.

The '936 patent uses the term "preload" throughout, both in reference to preloading the operating system and application programs. The description set forth in column 21 describes that "prior to host system reset, the data storage controller can proceed to pre[]load the portions of the computer operating system from the boot device (e.g., hard disk) into the on-board cache memory" (Ex. 1001, 21:23–26). The '936 patent further describes the preloading process may be completed prior to commencement of the boot process (*id.* at 22:8–17). Thus, we agree with Petitioner that in the '936 patent, preloading may be performed prior to the start of the boot process (Pet. Reply 4). Additionally, the '936 patent describes "cache **13** comprises volatile or non-volatile memory, or any combination thereof" (*id.* at 6:47–49). Therefore, the portion of the computer operating system preloaded from the boot device into cache is described as being preloaded into non-volatile memory, and as a result, the data would remain available for use after subsequent power off/on (*see* Pet. Reply 5; Ex. 2003, 25:9–14). Accordingly, the '936 patent describes preloading data for use after subsequent power off/on, which is not "in anticipation of immediate or near-in-time use."

We do not read the Specification as clearly and unmistakably requiring that data is preloaded in anticipation of immediate or near-in-time use. Descriptions in the Specification include data being preloaded in anticipation of immediate use; however, additional description does not limit “preloading” in this manner. And we will not import any such limitation into the claims at issue. Accordingly, taking a broad, but reasonable, interpretation in light of the Specification, we conclude an ordinarily skilled artisan would not interpret “preload” or “preloading” as narrowly as proffered by Patent Owner.

Patent Owner argues because the '936 Patent discloses “preloading” may be performed using all types of memory—volatile or non-volatile or a combination thereof—this confirms “preloading” is different than “storing” (PO Resp. 24). That all types of memory may be used does not preclude the data from being loaded prior to reset and into non-volatile cache, however.

Patent Owner also asserts during prosecution of the '936 patent, the Examiner issued an initial rejection based on a prior art reference directed to prefetching and thus, the Examiner “concluded that preloading and prefetching are similar” (*id.* at 16–17 (citing Ex. 1002, 759–764)). We do not find this to be informative as the Examiner has not provided any explanation regarding this reference and specifically, what aspect of the reference the Examiner reviewed or found relevant.

Therefore, we decline to interpret “preloading” as narrowly as proffered by Patent Owner. Rather, we interpret “preloading” as “transferring data from external storage to internal memory or from internal memory to registers prior to an event.”

2. “encoder”

Initially, we note the '936 patent does not define “encoder” explicitly. Petitioner asserts “a component that performs encoding operations is commonly understood to be an encoder” (Pet. Reply 24). Patent Owner asserts an encoder would use a single algorithm (PO Resp. 39; Tr. 61).

The '936 patent discloses a single encoder may be comprised of multiple encoders, each performing parts of the encoding process of the input data (Ex. 1001, 24:59–61, 25:22–27). The '936 patent further indicates that “each of the encoders E1...En [(where n may =1)], processes a given input data block and outputs a corresponding set of encoded data blocks (*id.* at 24:35–38). In the described embodiment, “the data compression system **110** accepts data blocks from an input data stream and stores the input data block in an input buffer or cache **115**. . . . [T]he system processes the input data stream in data blocks that may range in size from individual bits through complete files or collections of multiple files” (*id.* at 24:4–10). Thus, the '936 patent describes each encoder may be processing only one bit of the file. Furthermore, the portions of the Specification identified by Patent Owner (Ex. 1001, 25:5–8, Fig. 9), are discussing “an exemplary data compression system” (*id.* at 24:1–3).

Although Patent Owner contends the proffered definition from *Microsoft Computer Dictionary* 4 (5th ed. 2002) (Exhibit 1021), provides examples that “undercut” Petitioner’s interpretation of “encoder” (Tr. 62:18–63:2), we do not agree as these are provided as *examples*. Rather, the definition provided is “[i]n general, any hardware or software that encodes information — that is, converts the information to a particular form or format” (Ex. 1021, 4). Indeed, the extrinsic evidence of definitions

proffered by Petitioner’s support that an ordinarily skilled artisan at the time of the invention would not have interpreted “encoder” as narrowly as proposed by Patent Owner (*see, e.g.*, Exhibit 1021, 3; Exhibit 1022, 3 (“a hardware device or software that assigns a code to represent data”)). As previously stated, the ’936 patent does not narrow the definition as proposed by Patent Owner.

Additionally, Patent Owner and Dr. Back’s assertion that Dr. Neuhauser “appears to acknowledge that Dye discloses a single encoder, when concluding his analysis of the ‘a plurality of encoders’ element by stating that ‘the parallel LZSS *encoder* of Dye is used to compress data’” (Ex. 2003 ¶ 96 (citing Ex. 1003 ¶ 210³); PO Resp. 42 (citing Ex. 1003 ¶ 210⁴)). The ’936 patent, however, states a single encoder may be comprised of multiple encoders, each performing parts of the encoding process of the input data (Ex. 1001, 24:59–61, 25:22–27). Thus, the ’936 patent discloses an encoder may be comprised of encoders. Accordingly, we do not agree with Patent Owner’s assertion.

Therefore, taking a broad, but reasonable interpretation, in light of the Specification, we construe encoder to mean “any hardware or software that converts information to a particular form or format.”

3. *Additional terms*

We determine that no additional claim terms require express construction at this stage (*see Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999) (Only terms which are in controversy

³ Dr. Back cites to paragraph 287 of the Neuhauser Declaration; however, we believe this is a cite to paragraph 210, and is read as such.

⁴ Patent Owner cites to paragraph 207 of the Neuhauser Declaration; however, we believe this is a cite to paragraph 210, and is read as such.

need to be construed, and only to the extent necessary to resolve the controversy)).

C. Overview of Asserted Prior Art

1. Reference Sukegawa (Ex. 1005)

Sukegawa relates to “a data storage system using a flash memory unit and an HDD [(hard disk drive)]” (Ex. 1005, Abstract). Figure 1 of Sukegawa is reproduced below.

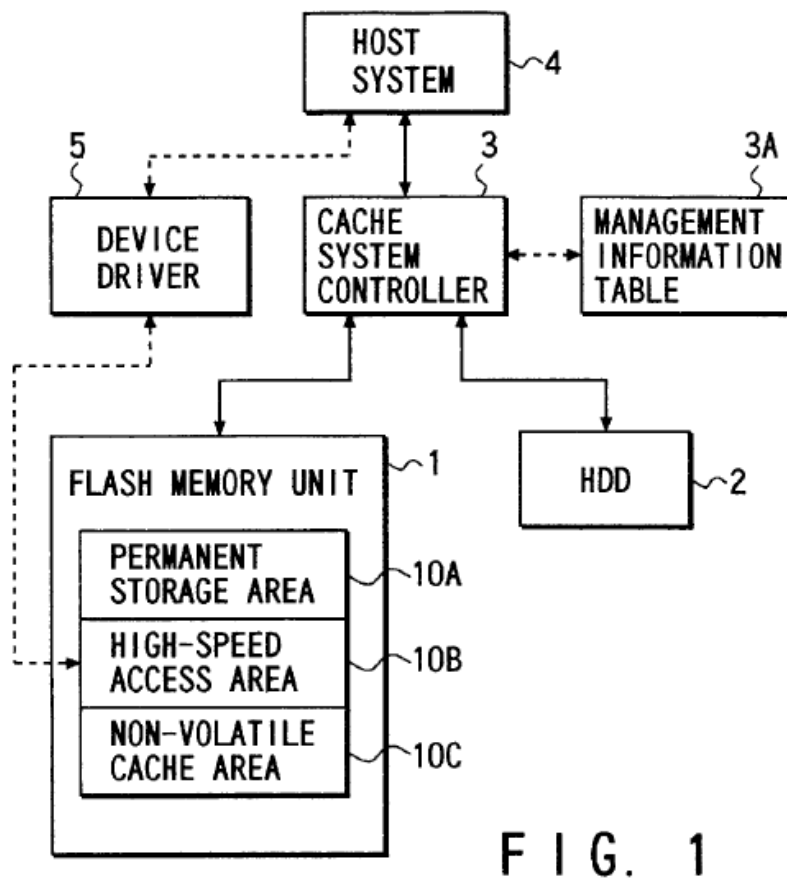


Figure 1 is “a block diagram showing a main part of a data storage system according to the present invention” (*id.* at 3:44–45). As shown in Figure 1, a data storage system includes flash memory unit 1, a hard disk drive (HDD) 2, cache system controller 3, and device driver 5 (*id.* at 4:4–

14). Sukegawa describes device driver 5 controls the flash memory 1 under management of the operation system of host system 4 (*id.* at 4:12–14). Controller 3 performs data input/output control for flash memory unit 1 and HDD 2 via respective device driver 5 (*id.* at 4:17–21). The flash memory unit is used, for example, to store “data which is used frequently for a relatively long time period” (*id.* at Abstract). Such data could include “control information necessary for starting an application program (AP) and an OS [(operating system)]” (*id.* at 2:65–3:3). Although such control information is stored on the HDD, the data may be stored also on the flash memory unit so that the OS may be started using the control information on the flash memory unit instead of the HDD (*id.* at 6:45–54). This is advantageous because the flash memory unit has a “higher access speed,” which allows the OS to be started more quickly (*id.* at 6:54–58).

Figure 4 of Sukegawa teaches an embodiment of a system having a data storage mode for storing control information necessary for storing the OS in the permanent storage area 10A of flash memory unit 1, when the OS is started in a series of operations from turn-on of power to completion of the starting operation (*id.* at 6:20–26).

When the system is switched on and the user sets the data storage mode via the user interface, controller 3 stores control information in permanent storage area 10A in flash memory unit 1 and when the OS is started at the time of the next turning-on of power, the control information necessary for starting the OS is read out not from permanent storage area 10A or cache memory area, and transferred to the host system 4 (*id.* at 6:27–54). “Thus, the control information can be accessed from the permanent storage area 10A in the flash memory unit 1 having a higher access speed

than the HDD 2. As a result, the OS can be started at higher speed” (*id.* at 6:45–58).

2. Reference Dye (Ex. 1008)

Dye relates to controllers for flash or embedded memory that include data compression and decompression engines “for increased effective memory density and improved bandwidth” (Ex. 1008, 1:17–22, 2:42–46). According to Dye, such a controller enables conventional flash memory to “achieve higher bandwidth, more effective density, with less system power and noise” (*id.* at 3:3–12, 3:23–28). The technology permits data to be “saved in either a normal or compressed format, retrieved from the Flash Memory Array for MPU [(microprocessing unit)] execution in a normal or compressed format, or transmitted and stored on a medium in a normal or compressed format” (*id.* at 3:66–4:8).

Figure 10B is reproduced below.

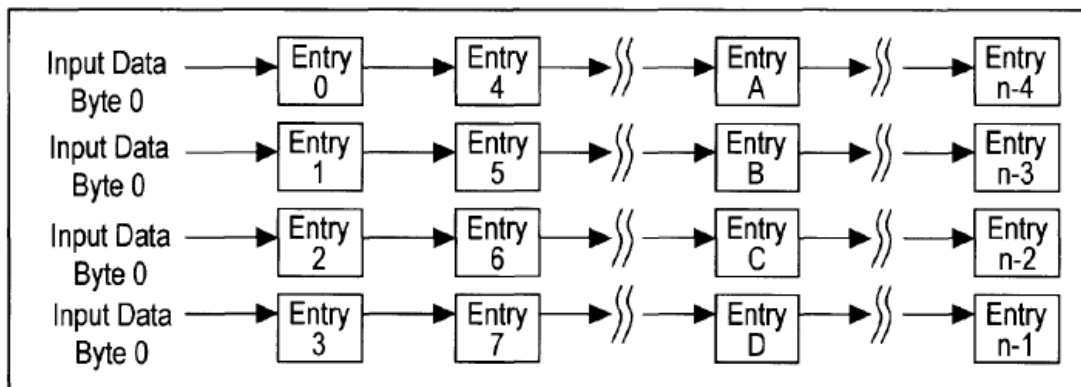


Fig. 10B
(New Art)

Figure 10B illustrates a preferred embodiment “provid[ing] a parallel implementation of dictionary based (or history table based) compression/decompression” (Ex. 1008 18:61–63). In the preferred

embodiment, Dye's Figure 10B illustrates the history table becomes a four symbol parallel flow (*id.* at 19:15–17). “[Four] symbols are analyzed in parallel, and multiple compressed outputs may also be provided in parallel” (*id.* at 19:17–19). “Other alternate embodiments may contain a plurality of compression windows for decompression of multiple streams” (*id.* at 19:19–21).

Figure 13 is reproduced below.

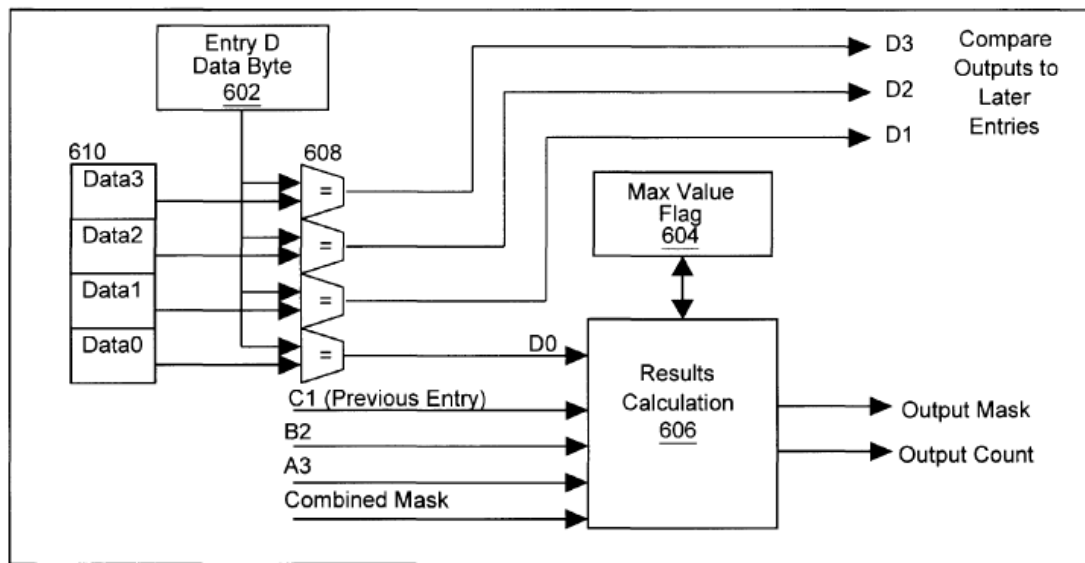


Fig. 13

Figure 13 is a hardware diagram illustrating “operation of the parallel compression algorithm” (*id.* at 22:66–67). Each entry of the history table contains a symbol of data, which is compared with the input stream (*id.* at 23:1–3). In Figure 13, Entry D Data Byte 602 is compared with each symbol of input data stream 610, shown as four data bytes, Data 0, 1, 2, and 3 (*id.* at 23:5–7). Comparators 608 compare each data byte to Entry D Data Byte 602, generating four compare signals (D0 through D3), with D1 through D3 being used by the next entry in the history table and D0 being

used by Results Calculation 606 (*id.* at 23:7–55). Results Calculation 606 sends Output Mask and Output Count to logic shown in Dye’s Figure 14 (*id.* at 23:19–24).

3. *Reference Settsu (Ex. 1006)*

Settsu relates to “[a] method of booting up an information processing apparatus” (Ex. 1006 at [57]). One embodiment described in Settsu involves dividing the main body of an operating system into modules and storing each module as compressed files on a boot device (*id.* at 14:58–63). Each of these modules is decompressed each time it is loaded into memory, and “the time required for I/O [(input/output)] processing can be reduced” as a result, which “provides an advantage of being able to further reduce the time required for booting up the information processing apparatus” (*id.* at 14:64–15:4).

D. *Alleged Obviousness of Claims 1–24 over Sukegawa and Dye*

Petitioner contends the combination of Sukegawa and Dye teaches or suggests the invention as recited in claim 1 (Pet. 23–37).

1. *Analysis: Claims 1 and 11*

a) *“preloading said at least a portion of said boot data in compressed form from said boot device to a memory”*

Petitioner contends the combination of Sukegawa and Dye teaches “preloading said at least a portion of said boot data in compressed form from said boot device to a memory” (Pet. 32–33). Specifically, Petitioner contends Sukegawa’s system includes flash memory unit 1, HDD2 (hard disk drive), and controller 3, that together perform data input/input control for flash memory unit 1 and HDD 2 (Pet. 32). According to Petitioner,

“Sukegawa’s controller 3 performs ‘a cache function’ by ‘using the flash memory unit 1 as a cache memory’” (*id.* (citing Ex. 1005, 4:7–10; Ex. 1003 ¶ 128)). More specifically, Petitioner contends Sukegawa teaches two techniques for preloading boot data into memory: (1) preloading user selected data and (2) preloading automatically selected data (*id.* (citing Ex. 1005, 5:10– 6:17, 6:19–58, 7:28–55)).

Petitioner further explains in the first technique, user-specified control information is read by cache system controller 3, out of HDD2, and preloads the information “‘in the permanent storage area 10A’ of flash memory unit 1, which has ‘a higher access speed than the HDD2’” (*id.* (citing Ex. 1005, 5:10—40, 6:19—58)). In the second technique, Sukegawa automatically preloads control information in flash memory unit 1 (*id.* at 33 (citing Ex. 1005, 7:28–55)).

Patent Owner asserts Sukegawa, upon which Petitioner relies, does not teach “preloading said at least a portion of said boot data in compressed form from said boot device to a memory,” as recited in claim 1 (PO Resp. 1–28). Specifically, Patent Owner contends in both of Sukegawa’s techniques, “data is permanently saved to non-volatile cache area 10C” and thus, is “storing” data, not “preloading” data (*id.* at 18–19). According to Patent Owner, Sukegawa is storing the data in persistent medium and therefore, is pre-installing the data in contrast to the ’936 patent which discloses preloading (*id.* at 20).

Initially, we note the ’936 patent describes use of a persistent memory for cache (see Ex. 1001, 6:47–49). Indeed, during his deposition, Patent Owner’s declarant, Dr. Back, acknowledges cache memory 13 in the ’936

patent “could be either volatile or nonvolatile memory” (Ex. 1017, 29:16–18).

Patent Owner further contends because the ’936 Patent discloses “preloading” may be performed using all types of memory—volatile or non-volatile or a combination thereof—this confirms “preloading” is different than Sukegawa’s “storing” (PO Resp. 24). This disclosure, however, supports a broader reading of preloading, i.e., that the ’936 patent’s description includes preloading data into persistent memory and because, once loaded the data will persist even when power is turned-off, for later use. Moreover, as noted by Petitioner, the ’936 patent uses “stored” to describe boot data being “stored in a cache memory device” (Pet. Reply 12 (citing Ex. 1001, Abstract, 3:57–58)). Thus, we do not interpret “preloading” as narrowly as Patent Owner proffers.

Patent Owner further contends “preloading” in the ’936 patent is not dependent on physical characteristics of the memory itself; rather, “‘preloading’ speeds up operating system and application load times by loading data into memory during the ‘unproductive’ wait times during other aspects of boot-up or application launch, so that it will be in memory when needed” (PO Resp. 25 (citing Ex. 1001, 21:3–14, 15–44, 50–23–58)). Sukegawa also is directed to “providing accelerated loading of operation system and application programs upon system boot or application launch” (*id.* at 1:17–20). To that end, Sukegawa teaches flash memory unit 1 stores data such as control information necessary for starting an operating system (boot data) (Ex. 1005, 2:65–3:3). Sukegawa further teaches the control information, pre-stored in HDD2, is read out and stored in permanent storage area 10A and “when the OS is started at the time of the next turning-

on of power, the control information necessary for starting the OS is read out . . . from the permanent storage area **10A** or cache memory area” (*id.* at 6:36–54). Neither claim 1 nor claim 11 recites that the preloading occurs during the “unproductive” wait times. Moreover, Patent Owner’s arguments are predicated on “*if* [Sukegawa’s] flash memory unit 1 had the same (or slower) access rate than HDD2” (*id.* (emphasis added)). Sukegawa teaches preloading data into memory and thus, provides a faster access rate than if the system were to access data from HDD2.

Patent Owner’s next contention is that “[t]he ’936 patent explains, *in one exemplary embodiment*, that preloading occurs upon each boot process” (*id.* at 26 (emphasis added)). As discussed above, however, we do not construe the term “preloading” as narrowly as proposed by Patent Owner. Indeed, Patent Owner would have us construe the claim term as limited to the preferred embodiment without a clear intent to redefine the term or a clear disavowal of claim scope and most importantly, such an interpretation would *exclude* examples of preloading in the ’936 patent as discussed above.

Patent Owner also contends Sukegawa is distinguishable because the ’936 patent describes “the recurrence of the ‘preloading’ operation, and the lack of such recurrence for ‘storing’” (PO Resp. 26). According to Patent Owner, “*in one exemplary embodiment*, that preloading occurs upon each boot process” (*id.* (emphasis added)). As noted by Patent Owner, however, this is one exemplary embodiment (*see also* Pet. Reply 13). Patent Owner contends “Sukegawa’s techniques are performed only once, and need not be performed again for subsequent boot cycles” (*id.* at 27). Yet the recitation of claims 1 and 11 does not require “the recurrence of the ‘preloading’

operation” however. Rather, the claims recite “preloading said at least a portion of said boot data” without any repeating of the “preloading” step.

In light of this interpretation, we agree with Petitioner that Sukegawa teaches “preloading said at least a portion of said boot data . . . from said boot device to a memory,” as recited in claim 1 and “said at least a portion of said boot data . . . is preloaded into said memory,” as recited in claim 11.

b) “boot data in compressed form . . . stored in compressed form on a boot device”

Petitioner contends US Patent 7,190,284 B1 (Dye ’284), which is incorporated by reference in Dye, teaches “data may be stored on the hard disk . . . in compressed format” (Pet. 30 (citing Ex. 1009, 8:44–53)). Petitioner argues “Dye unequivocally incorporates Dye ’284” and “is unambiguous in its incorporation of the entire content of Dye ’284” (Pet. Reply 26).

Patent Owner contends Dye does not disclose storing compressed data in a hard disk drive (PO Resp. 44). According to Patent Owner, even if an ordinarily skilled artisan would have modified Sukegawa based on Dye’s teachings, “such a modification would have been to add Dye’s controller to Sukegawa’s flash memory unit 1 to compress data as it is being transferred into flash memory unit 1 and to decompress data output from unit 1” (*id.* at 46). As such, Patent Owner asserts, the modification “fails to render obvious storage of compressed boot data on Sukegawa’s HDD2 as the boot data stored on HDD2 would remain uncompressed” (*id.*).

Petitioner relies on Dye as teaching compression/decompression and Dye ’284 as teaching compressing data on a hard drive (Pet. 29–30). Patent

Owner argues “Dye ’284 is not properly incorporated by reference into Dye . . . since Dye does not specifically identify portions of Dye ’284 that are incorporated by reference” (PO Resp. 47). Furthermore, Patent Owner contends “the Petition does not assert that the elements of ’936 claims lacking in Sukegawa are alone disclosed by Dye” and thus, Petitioner has not shown the combination of Sukegawa and Dye teaches the disputed limitation (*id.* at 48–49).

Based on the entirety of the record before us, we find that the combination of Sukegawa and Dye teaches the disputed limitation. “To incorporate material by reference, the host document must identify with detailed particularity what specific material it incorporates and clearly indicate where that material is found in the various documents” (*Apple Inc. v. Interantional Trade Comm’n*, 725 F.3d 1356 (Fed. Cir. 2013)). The host document “must cite the material in a manner that makes clear that it is effectively part of the host document as if it were explicitly contained therein” (*Advanced Display Sys. Inc. v. Kent State Univ.*, 212 F.3d 1272, 1282 (Fed.Cir.2000)).

Here, unlike in the cases cited by Patent Owner (PO Resp. 47, n.150), Dye specifically states the U.S. patent application (now Dye ’284) is “incorporated by reference in its entirety as though fully and completely set forth herein” (Ex. 1008, 6:3–9). Indeed, the specific material to be incorporated is identified with detailed particularity:

U.S. patent application Ser. No. 09/239,659 titled “Bands width Reducing Memory Controller Including Scalable Embedded Parallel Data Compression and Decompression Engines[,]” which was filed on Jan. 29, 1999 (5143-01700), is hereby incorporated by reference in its entirety as though fully and completely set forth herein

(*id.*; see also *id.* at 3:66–4:19 (discussing specific features discussed in Dye '284)). Thus, Dye cited Dye '284 in a manner that makes clear that Dye '284 is effectively part of Dye, as if it were explicitly therein and accordingly, Dye properly incorporates Dye '284 by reference. We further note that Dye is a continuation-in-part of Dye '284 and thus, has a significant amount of overlapping disclosure (*id.* at 1:9–13, [63]).

Petitioner relies on the combination of Sukegawa's HDD2 (boot drive) and Dye's teaching of compressed data on the boot device (Pet. 28–29). Petitioner asserts because “the storage area of the flash memory is limited” in Sukegawa, a POSITA would have appreciated how compression would have improved storage capacity of Sukegawa's flash memory unit 1, as well as speed of data access” (Pet. 10 (citing Ex. 1003 ¶ 60; Ex. 1005, 2:28–29)). According to Petitioner, “Dye would have motivated a POSITA to use compression and compressed boot data in Sukegawa's system” (Pet. 11 (citing Ex. 1003 ¶¶ 61, 67), 29 (citing Ex. 1003 ¶ 108)). Petitioner argues Dye teaches “it was well-known to increase the speed of accessing data from non-volatile storage devices, such as Sukegawa's HDD2 and flash memory unit 1” (*id.* at 29 (citing Ex. 1003 ¶¶ 108–113)). Petitioner further asserts “[a]lso, through incorporated disclosure from Dye '284, Dye confirms that the controller ‘may couple to any of various types of memory, as desired,’ enabling it to ‘greatly increas[e] the performance of the computer system’” (*id.* (citing Ex. 1009, 10:19–41)). Petitioner additionally contends

Dye [‘284] also describes using “the compression/decompression engine” to compress the same types of data that are stored in Sukegawa's HDD2 and flash memory unit 1: namely, “application data” and “operating system data.” Dye '284, 11:32–35, 38:29–39:44, FIGS. 22, 23. And Sukegawa contemplates the need for updates to application/OS data where

“the updated data is written in the HDD 2.” Sukegawa, 9:22-10:3.

This disclosure would have motivated a POSITA to modify Sukegawa’s controller 3 to use Dye’s compression/decompression engine for compressing control information and to provide/store the compressed control information to HDD2. Dec., ¶113. Indeed, a POSITA would have understood that Sukegawa’s modified controller 3, which performs data input/output control for HDD2, compresses control information, and provides the compressed control information to HDD2 for storage. *Id.*; Dye ’284, 11:28–12:33.

(Pet. 30–31).

Dye teaches use of parallel compression and decompression technology (Ex. 1008, Abstract). In particular, Dye teaches Compression Enhanced Flash Memory controller (CEPMC) allow data to be stored in normal or compressed format or transmitted and stored on a medium in normal or compressed format (*id.* at 3:66–4:8, 7:35–43, 8:6–12). Dye ’284 teaches data may be stored in compressed form on non-volatile memory (Ex. 1009, 4:16–24) and that such data may include program data and operating system data (*id.* at 11:32–35, 38:30–42, Figs. 22, 23). As set forth by Petitioner’s declarant, Dr. Neuhauser, Dye ’284 “describes benefits of using compression in several types of computer system memory devices, including hard disks and flash memory” (Ex. 1003 ¶ 45 (citing Ex. 1009, 4:16–24, 11:28–37, 12:19–22; Ex. 1008, Abstract, 7:34–43)). We credit Dr. Neuhauser’s statement that Sukegawa’s HDD2 and flash memory unit 1 are “hard disks and flash memory” as described in Dye ’284 (*id.* ¶ 62). We further credit Dr. Neuhauser’s statement that “[b]ased on the clear teachings of Dye with respect to the advantages of data compression, one of ordinary skill [would] have been motivated to apply Dye to the system of Sukegawa

to obtain improved performance of [Sukegawa's] Flash Memory Unit 1" (*id.* ¶ 67; *see also id.* ¶¶ 112–114).

Furthermore, Petitioner is not relying on incorporation of Dye's controller, but rather on Dye's data compression/decompression and storage of compressed/decompressed data (Pet. 30–31).

To justify combining reference teachings in support of a rejection it is not necessary that a device shown in one reference can be physically inserted into the device shown in the other. The test for obviousness is not whether the features of a secondary reference may be bodily incorporated into the structure of the primary reference; nor is it that the claimed invention must be expressly suggested in any one or all of the references. Rather, the test is what the combined teachings of the references would have suggested to those of ordinary skill in the art.

(*In re Keller*, 642 F.2d 413, 425 (CCPA 1981) (citations omitted)).

We find that Petitioner has articulated reasoning why an ordinarily skilled artisan would have been motivated to combine the teachings of Sukegawa and Dye and determine that the teachings and suggestions of Sukegawa and Dye are combined properly.

c) Remaining Limitations

We have reviewed the remaining limitations and based on the entirety of record before us, we determine that the combination of Sukegawa and Dye teaches or suggests claims 1 and 11. Furthermore, based on the entirety of the record before us, Petitioner has established by a preponderance of the evidence that claims 1 and 11 of the '936 patent are unpatentable under 35 U.S.C. § 103(a) as obvious over Sukegawa and Dye.

2. Analysis - Claim 18

Petitioner asserts Sukegawa's cache memory teaches the recited "memory" (Pet. 51). Petitioner further asserts Sukegawa teaches two

techniques for preloading control information which includes program code associated with an OS, into memory and specifically, teaches (1) user selection of data to preload and (2) automatic selection of data to preload (*id.*). According to Petitioner, Sukegawa teaches “preloaded control information includes program code associated with an OS of its host computer system . . . [including] the OS itself” (*id.* at 52 (citing Ex. 1005, 2:65–3:3, 2:11–16; Ex. 1003 ¶ 291)). Thus, Petitioner asserts, Sukegawa’s “preloading OS control information from HDD2 into flash memory unit 1” teaches “preloading, from a boot device into a memory, a portion of the OS (e.g., a user-selected portion or an automatically-selected portion)” (*id.* (citing Ex. 1003 ¶ 291)).

Petitioner additionally contends “[b]oth preloading techniques are employed responsive to a request (by the user and/or host system) (*id.* at 55 (citing Ex. 1003 ¶¶ 311–315)). More specifically, Petitioner contends “both techniques use table 3A to locate control information, and a POSITA would have understood that both techniques are repeatable (*id.* (citing Ex. 1003 ¶¶ 311–315)). Petitioner further contends “[a] POSITA would have further understood that repeating either technique for a different portion of the OS after preloading a first portion of the OS (e.g., a first portion of an OS) would result in preloading a second, different portion of the OS (e.g., a second portion of the OS)” (*id.* (citing Ex. 1003 ¶¶ 311–315)).

Patent Owner contends Sukegawa fails to teach “both preloading ‘a first portion’ of the OS into memory ‘to partially boot’ the system and preloading ‘a second portion’ of the OS into memory ‘to further partially boot’ the system” (PO Resp. 29). More specifically, Patent Owner contends Petitioner has set forth Sukegawa’s storage of control information to flash

memory is relied on for teaching both “preloading a first portion” and “preloading the second portion” (*id.* at 30). Moreover, Patent Owner contends “Sukegawa does not disclose using any of its control information ‘to partially boot’ using a first portion of the OS and ‘to further partially boot’ using a second portion of the OS” (*id.*). Additionally, Patent Owner asserts Petitioner’s contention that “repeating either technique for a different portion of the OS after preloading a first portion of the OS,” does not remedy that Sukegawa does not disclose that the OS data or additional OS control information read out from HDD2 “serves the same purpose as the control information . . . identifie[d] as the ‘first portion’ of the operating system — namely, to start up the OS in the first instance” (*id.* at 32). Thus, according to Patent Owner, “this data would not be used ‘to further partially boot’ the operating system, but to directly boot the operating system in the first instance”; rather, this data “would not be available for use until the next power-on cycle” (*id.* at 34–35).

Initially, we note the preloading of the second portion does not have to occur *after* the “accessing and decompressing” necessarily and “utilizing” steps for the first portion. “[A]lthough a method claim necessarily recites the steps of the method in a particular order, as a general rule the claim is not limited to performance of the steps in the order recited, unless the claim explicitly or implicitly requires a specific order” (*Baldwin Graphic Sys., Inc. v. Siebert, Inc.*, 512 F.3d 1338, 1345 (Fed. Cir. 2008)). Here, no sequential language is present to indicate that the steps must occur in the order written, except for the preloading of a *first* portion and locating and preloading a *second* portion.

Sukegawa teaches, in a “first modification of the first embodiment,” storing the operating system in flash memory unit 1 (functioning as cache memory area) (Ex. 1005, 6:61–66). Specifically, Sukegawa teaches

[f]or example, at the time of shipment of the personal computer, if the OS is preinstalled in the flash memory unit 1 functioning as cache 65 memory area, the user can delete the OS control information based on the user’s judgment. Specifically, the OS control information may be deleted by a user who is used to starting the OS by making use of a resume function of BIOS28, without re-starting the personal computer.

(*id.*). Petitioner further relies on Sukegawa’s teaching in a second embodiment, controller 3 determines whether data to be accessed is stored in permanent storage area 10A or non-volatile cache area 10C (*id.* at 7:30–35). “if the data to be accessed is ‘hit,’ the controller 3 reads the data from the permanent storage area 10A or non-volatile cache area 10C and transfers the read-out data to the host system 4” (*id.* at 7:35–39).

We agree with Patent Owner that Sukegawa does not teach or suggest the operating system is preloaded in two different portions. Although Sukegawa teaches two methods, we find Sukegawa does not teach using the two methods together. Rather, Sukegawa presents them as alternate techniques. Additionally, although we agree that performing a step may be repeatable, this would only teach preloading the operating system and then preloading the operating system again.

Therefore, based on the record before us, we determine the combination of Sukegawa and Dye fails to teach or suggest claim 18. It follows, Petitioner has not established by a preponderance of the evidence that claim 18 of the ’936 patent is unpatentable under 35 U.S.C. § 103(a) as obvious over Sukegawa and Dye. Additionally, Petitioner has not

established by a preponderance of the evidence that dependent claims 19–24 are unpatentable under 35 U.S.C. § 103(a) as obvious over Sukegawa and Dye.

3. *Analysis - Claims 9, 10, 16, and 17*

Claims 9 and 16 recite “a plurality of encoders are utilized to provide said at least a portion of compressed data in compressed form” and claims 10 and 17 recite “a plurality of encoders in a parallel configuration are utilized to provide said at least a portion of said data in compressed form” (Ex. 1001, Claims).

Petitioner contends Dye teaches a compression/decompression engine that uses a “parallel lossless compression/decompression” “designed to process stream data at more than a single byte or symbol (character) at one time” (Pet. 42 (citing Ex. 1009, 4:9–30)). Specifically, Petitioner asserts “Dye’s compression algorithm analyzes multiple symbols in parallel, and provides ‘multiple compressed outputs’ in parallel, and provides ‘multiple compressed outputs’ in parallel” (*id.* (citing Ex. 1008, 18:43–19:30, 22:66–23:24, Fig. 13; Ex. 1009, 12:61–13:7, 13:52–56, 14:49–50, 29:13–14)).

Petitioner further contends “a POSITA would have modified Sukegawa’s controller 3 to include Dye’s compression/decompression engine to provide compressed boot data” (Pet. 42 (citing Ex. 1003 ¶¶ 61–71, 191)).

Patent Owner asserts Dye and Patent Owner’s and Petitioner’s experts all “agree that Dye only discloses a single encoder” (PO Resp. 38). Specifically, Patent Owner asserts Figure 10 depicts a single *algorithm* and thus, teaches a “single encoder that distributes the encoding calculations among several stages” (*id.* at 39). Rather, according to Patent Owner, “each

of these stages is not a separate encoder — on the contrary, each unit is a part of Dye’s single encoder” (*id.* at 39–40). Patent Owner further contends Petitioner’s declarant, Dr. Neuhauser, “ultimately confirms that Dye discloses a single encoder, concluding his analysis of the ‘a plurality of encoders’ element by stating that ‘the parallel LZSS *encoder* of Dye is used to compress data’” (*id.* at 40 (citing Ex. 1003 ¶ 207)).

Patent Owner contends each of the plurality of encoders has its own type of compression (Tr. 60:18–21) whereas Dye teaches one single encoder (*id.* at 61:8–9). Petitioner responds that Figure 13 of Dye “serves ‘as an encoder of the Input Stream’ ‘because this logic **encodes** four bytes of the input stream 610 against the four bytes in the history table” and Dye further discloses in Figure 13 logic, for one stream – “there are 16 such encoders used in the system of Dye” (Pet. Reply 23 (citing Ex. 1003 ¶¶ 207–209)).

Dye teaches, in Figure 10B, entries A, B, C, and D (Ex. 1008, Fig. 10B). Figure 13 teaches processing using entry D, and, as such, logic shown in Figure 13 would be present for processing each of entries A, B, and C (Ex. 1008, Figs. 10B, 13).

Initially, as set forth above, we interpret “encoder” as “any hardware or software that converts information to a particular form or format.” In light of this interpretation, we determine Dye teaches a plurality of encoders. Specifically, we credit Dr. Neuhauser that Dye teaches the logic shown in Figure 13 would be present for each Data Entry (e.g., Data Entry A, B, C, D) and thus, teaches a plurality of encoders (Ex. 1003 ¶¶ 205–209; Ex. 1008, Figs. 10B, 13). Each logic shown would perform a series of steps for performing a task (an algorithm).

Additionally, we find Dye teaches these encoders may be in a parallel configuration (Ex. 1008, 18:60–63, 22:64–23:4, Figs. 10B, 13). Thus, based on the entirety of the record before us, we find Dye teaches “a plurality of encoders are utilized to provide said at least a portion of compressed data in compressed form,” as recited in claims 9 and 16 and “a plurality of encoders in a parallel configuration are utilized to provide said at least a portion of said data in compressed form,” as recited in claims 10 and 17. It follows, Petitioner has established by a preponderance of the evidence that claims 9, 10, 16, and 17 of the ’936 patent are unpatentable under 35 U.S.C. § 103(a) as obvious over Sukegawa and Dye.

4. Analysis – Claims 2–8 and 12–15

We have reviewed the remaining dependent claims, claims 2–8 and 12–15. Based on the entirety of the record before us, Petitioner has established by a preponderance of the evidence that claims 2–8 and 12–15 of the ’936 patent are unpatentable under 35 U.S.C. § 103(a) as obvious over Sukegawa and Dye.

E. Alleged Obviousness of Claims 18–24 over Settsu

1. Analysis – Claim 18

Petitioner contends the combination of Sukegawa, Dye, and Settsu teaches the limitations recited in claim 18. As set forth above with respect to claims 1 and 11, Patent Owner asserts neither Sukegawa nor Dye, alone or in combination teaches “preloading” data into memory (PO Resp. 1–28). As set forth above, Petitioner has persuaded us Sukegawa teaches “preloading” data into memory.

Petitioner contends “Settsu describes preloading, from boot device 3 into memory 2, a mini OS module 7 and then individually preloading each

OS main body module (*id.* at 64–65 (citing Ex. 1006, Abstract, 8:21–35, 8:66–9:3, 9:7–11, 10:67–11:3, 11:7–9, 11:24–33)). Thus, Petitioner argues, “a POSITA would have been motivated to modify Sukegawa to preload a first portion of the OS (e.g., a mini OS module or a first OS main body module into Sukegawa’s flash memory unit” (*id.* at 65 (citing Ex. 1003 ¶ 294)).

Petitioner additionally argues Settsu teaches checking “whether or not the whole of the main body of the OS has been loaded, that is whether or not all the functional modules 16 to 22 have been loaded into memory 2” in response to “a request to boot and/or a request for an additional OS module in the process of loading the OS main body from boot device 3” (*id.* at 67 (citing Ex. 1006, 14:44–52, Fig. 14)). Petitioner further argues “a POSITA would have found it obvious to use a list of boot data modules” as taught by Settsu’s “function definition file,” “to determine whether or not, responsive to a request, all of the functional OS modules have been loaded” (*id.* at 69 (citing Ex. 1003 ¶¶ 319–321)). Therefore, according to Petitioner,

a POSITA would have found it obvious to use a boot data list (e.g., a function definition file) to locate a second functional module of the OS main body (e.g., an OS main body module different than the first module . . .) and would have preloaded the second functional module into Sukegawa’s flash memory unit 1 (*id.*).

Patent Owner asserts Settsu fails to teach “loading a first OS portion to partially boot the system and loading a second OS portion to further boot the system” (PO Resp. 36). More specifically, according to Patent Owner, “Settsu does not disclose ‘preloading’ a first OS portion ‘to partially boot’

the system and ‘preloading’ a second OS portion ‘to further partially boot’ the system” (*id.* at 36–37).

We agree with Petitioner that Settsu teaches the disputed limitation. Specifically, Settsu teaches a mini operating system (OS) module located in boot block 4 within the boot device, “the mini OS module having OS functions required for bootstrap processing” and OS main body module 8, located with the boot device and “provided with OS functions except the OS functions included in the mini OS module 7” (Ex. 1006, 8:13–21). “The F/W code module 6 loads the mini OS module 7 into the memory 2 and transfers control to the mini OS module 7” (*id.* at 9:7–11). “The OS main body module 8 is located within the file system 5 of the boot device 3 and is loaded into the memory 2 by the mini OS module 7” (*id.* at 8:66–9:2).

Thus, we find Settsu teaches “preloading a first portion of the . . . operating system from said boot device to a memory [2]” and preloading a second portion of the operating system to memory [2].

Settsu further teaches the mini OS module (the first portion) has “OS functions required for bootstrap processing” (*id.* at 8:16–21). Moreover, as shown in Figures 14 and 20, Settsu loads OS functional modules of the main body, decompresses each OS functional module until loading of all the modules is complete (*id.* at Figs. 14, 20). Thus, the mini OS module “partially boots” the system and the OS main body module (the second portion — the OS functions except those in the mini OS module) and is used to “further partially boot” the system. Accordingly, we agree with Petitioner that the combination of Sukegawa, Dye, and Settsu teaches or suggests “utilizing the decompressed first portion to partially boot said computer

system” and “utilizing the decompressed second portion to further partially boot said computer system,” as recited in claim 18.

Patent Owner, however, contends the combination of teachings of Sukegawa, Dye, and Settsu fails to render obvious “storing *substantially all* of the operating system in compressed form on a boot device” (*id.* at 37). More specifically, Patent Owner contend Settsu’s storage of mini OS module 7 and OS main body module 8 has not been shown to store “‘substantially all’ of the operating system” (*id.*).

The term “substantially all” is not defined explicitly in the Specification nor is the term used in the Specification (except for the claims). Rather, as shown in Figures 7A and 7B, the ’936 patent discloses prefetching data blocks specified in the list and then determining if the requested boot data is preloaded (Ex. 1001, Fig. 7B, elements 77, 80, 22:8–30). If the requested boot data is not preloaded, the controller will retrieve the requested data from the boot device (Ex. 1001, 22:25–30).

Settsu teaches both mini OS module 7 and OS main body module 8 are stored in boot device 3 (Ex. 1006, 8:2–8). Moreover, as set forth by Petitioner (Pet. Reply 21), Sukegawa teaches storing the operating system in the cache memory area (*id.* at 6:63–66). We agree with Petitioner and credit Dr. Neuhauser’s statement that, based on Sukegawa’s discussion, “[o]ne of ordinary skill, therefore, would have understood that substantially all (if not all) of the operating system is stored on HDD2” (Ex. 1003 ¶ 280 (citing Ex. 1005, 1:39–49, 6:35–39, 6:49–58)). Indeed, we determine, in light of the teachings of Settsu and Sukegawa that an ordinarily skilled artisan would have understood the references to teach or suggest preloading “substantially all” of the operating system. Moreover, we find that storing “substantially

all” of the operating system instead of “all” of the operating system, would not have been “uniquely challenging or difficult for one of ordinary skill in the art” or “represented an unobvious step over the prior art” (see *Leapfrog Enters., Inc. v. Fisher-Price, Inc.*, 485 F.3d 1157, 1162 (Fed. Cir. 2007)).

Thus, based on the entirety of the record before us, we determine Petitioner has established by a preponderance of the evidence that claim 18 of the ’936 patent is unpatentable under 35 U.S.C. § 103(a) as obvious over Sukegawa, Dye, and Settsu.

2. Analysis - Claims 23 and 24

Claim 23 recites “a plurality of encoders are utilized to provide said at least a portion of compressed data in compressed form” and claim 24 recites “a plurality of encoders in a parallel configuration are utilized to provide said at least a portion of said data in compressed form” (Ex. 1001, Claims).

Claim 23 is commensurately recited to claim 9 and claim 24 is commensurately recited to claim 10. For the reasons set forth above with respect to these claims, we agree with Petitioner that Dye teaches “a plurality of encoders are utilized to provide said at least a portion of compressed data in compressed form,” as recited in claim 23 and “a plurality of encoders in a parallel configuration are utilized to provide said at least a portion of said data in compressed form,” as recited in claim 24. It follows, based on the entirety of the record before us, we determine Petitioner has established by a preponderance of the evidence that claims 23 and 24 of the ’936 patent are unpatentable under 35 U.S.C. § 103(a) as obvious over Sukegawa, Dye, and Settsu.

3. Analysis – Claims 19–22

We have reviewed the entirety of the record and agree with Petitioner that claims 19–22 are taught or suggested by the combination of Sukegawa, Dye, and Settsu. Thus, we determine, based on the entirety of the record before us, Petitioner has established by a preponderance of the evidence that claims 19–22 of the '936 patent are unpatentable under 35 U.S.C. § 103(a) as obvious over Sukegawa, Dye, and Settsu.

F. Alleged Obviousness of Claims 1–10 over Sukegawa, Dye, and Settsu; Sukegawa, Dye, and Burrows; and Sukegawa, Dye, Settsu, and Burrows

Petitioner contends claims 1–10 are unpatentable under 35 U.S.C. § 103 over Sukegawa, Dye, and Settsu; Sukegawa, Dye, and Burrows; and Sukegawa, Dye, Settsu, and Burrows (Pet. 58–62, 71). In light of our determination that Petitioner has established by a preponderance of the evidence that claims 1–10 of the '936 patent are unpatentable under 35 U.S.C. § 103(a) as obvious over Sukegawa and Dye, we take no position on whether the same claims also are obvious over the combination of the teachings and suggestions of Sukegawa, Dye, and Settsu; Sukegawa, Dye, and Burrows; and Sukegawa, Dye, Settsu, and Burrows.

G. Conclusion

For the foregoing reasons, we determine that Petitioner has demonstrated, by a preponderance of the evidence, that claims 1–17 are unpatentable under 35 U.S.C. § 103 as obvious over Sukegawa and Dye, and claims 18–24 are unpatentable under 35 U.S.C. § 103 over Sukegawa, Dye, and Settsu.

III. ORDER

In consideration of the foregoing, it is hereby:

ORDERED claims 1–24 of U.S. Patent No. 8,090,936 B2 are held

unpatentable; and

FURTHER ORDERED that, because this is a Final Written Decision, parties to the proceeding seeking judicial review of the decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

Case IPR2016-01366

Patent 8,090,936 B2

PETITIONER:

Walter Renner

Jeremy Monaldo

Andrew Patrick

Katherine Vidal

FISH & RICHARDSON P.C.

IPR39521-0024IP1@fr.com

PTABInbound@fr.com

PATENT OWNER:

Richard Z. Zhang

Joseph E. Edell

FISCH SIGLER LLP

richard.zhang.IPR@fischllp.com

joe.edell.IPR@fischllp.com