UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/661,478	10/26/2012	Akella Sastry	NVDA/SC-11-0302-US0-US1	1861

102324 7590 02/28/2018
Artegis Law Group, LLP/NVIDIA
7710 Cherry Park Drive Suite T #104
Houston, TX 77095

EXAMINER	
AQUINO, WYNUEL S	

ART UNIT	PAPER NUMBER
2199	

NOTIFICATION DATE	DELIVERY MODE
02/28/2018	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

kcruz@artegislaw.com
ALGdocketing@artegislaw.com
rsmith@artegislaw.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte AKELLA SASTRY and YUAN LIN

Appeal 2017-009801
Application 13/661,478
Technology Center 2100

Before JOHN A. JEFFERY, BRUCE R. WINSOR, and JUSTIN BUSCH, *Administrative Patent Judges*.

WINSOR, *Administrative Patent Judge*.

DECISION ON APPEAL

Pursuant to 35 U.S.C. § 134(a), Appellants[1] appeal from the Examiner's decision to reject claims 1–21, which constitute all the claims pending in this application. We have jurisdiction under 35 U.S.C. § 6(b).

We affirm.

[1] Appellants identify the real party of interest as Nvidia Corporation. App. Br. 3. Nvidia Corporation is the Applicant for the instant patent application. *See* Bib. Data Sheet.

STATEMENT OF THE CASE

Appellants' disclosed invention "relates to . . . [reducing] sign-extension instructions included in loops of a 64-bit computer program." Spec. ¶ 2. Claim 1, which is illustrative, reads as follows:

> 1. A method for reducing sign-extension instructions (SEIs) included in a computer program, the method comprising:
>
> receiving intermediate code that is associated with the computer program and includes a first SEI that is included in a loop structure within the computer program;
>
> determining that the first SEI is eligible to be moved outside of the loop structure;
>
> inserting into a preheader of the loop a second SEI that, when executed by a processor, promotes an original value targeted by the first SEI from a smaller type to a larger type; and
>
> replacing, via a first processor, the first SEI with one or more intermediate instructions that are eligible for additional compiler optimizations.

Claims 1, 2, 5, 6, 9–11, 14, 15, and 17–21 stand rejected under 35 U.S.C. § 103(a)[2] as being unpatentable over Mitran et al. (US 2011/0107068 A1; May 5, 2011) and Deneau (US 2011/0078653 A1; Mar. 31, 2011). *See* Final Act. 4–10.

Claims 3 and 12 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Mitran, Deneau, and Santhanam (US 6,286,135 B1; Sept. 4, 2001). *See* Final Act. 10–12.

[2] All rejections are under the provisions of 35 U.S.C. in effect before the effective date of the Leahy-Smith America Invents Act of 2011 ("pre-AIA"). Final Act. 2.

Claims 4 and 13 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Mitran, Deneau, Santhanam, and Tim Teitelbaum, *Introduction to Compilers*, CS412/413 Lecture 29: Control Flow Analysis and Loop Optimization (Apr. 4, 2008). *See* Final Act. 12.

Claims 7, 8, and 16 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Mitran, Deneau, and Andrew Myers, *Induction variable optimizations*, CS4120 Lecture 26 (Nov. 4, 2009). *See* Final Act. 12–14.

Rather than repeat the arguments here, we refer to the Briefs ("App. Br." filed Dec. 27, 2016; "Reply Br." filed July 10. 2017) for the positions of Appellants; the Final Office Action ("Final Act." mailed Aug. 25, 2016), and Examiner's Answer ("Ans." mailed May 9, 2017) for the reasoning, findings, and conclusions of the Examiner; and the Specification ("Spec." filed Oct. 26, 2012). Only those arguments actually made by Appellants have been considered in this decision. Arguments that Appellants did not make in the Briefs have not been considered and are deemed to be waived. *See* 37 C.F.R. § 41.37(c)(1)(iv) (2015).

ISSUES

The issues presented by Appellants' arguments are as follows:

Does the Examiner err in finding the combination of Mitran and Deneau teaches or suggests "receiving intermediate code that is associated with the computer program and includes a first SEI [sign-extension instruction] that is included in a loop structure within the computer program" (hereinafter the "loop limitation"), as recited in claim 1?

Is the Examiner's reason to combine the teachings of Mitran and Deneau supported by articulated reasoning with some rational underpinning to justify the Examiner’s obviousness conclusion? This issue turns on whether Deneau teaches away from a first SEI converting a 32-bit operand to a 64-bit operand without changing the sign or value associated with the operand, as recited in claim 2.

ANALYSIS

Claims 1, 5–10, 14–19, and 21

The Examiner relies on Mitran to disclose the loop limitation of claim 1. Final Act. 2–3, 6 (citing Mitran ¶¶ 32–33, 36, 39–40, 59–61; Figs. 4, 6–8); Ans. 14 (additionally citing Mitran ¶ 62). According to the Examiner, “Ep is the ser of operations to establish [a] common property P,’” as taught by Mitran, and “may include [a] sign extension operation.” Ans. 13 (emphasis omitted). The Examiner further finds “Fig. 6 [of] Mitran [teaches], prior to optimization, a loop (‘BB_2 Lloop’) that includes a ‘common property p’ as a ‘sign extension instruction’ as ‘$V_2[R_1]$ = *set_upper_word_to_C()*.’” *Id.* (emphasis omitted).

Appellants argue Mitran’s common property P refers to the contents of a register and not an operation. Reply Br. 3. Appellants assert Mitran’s operations of Figures 6 and 7 “assign values to registers (e.g., the v_2 virtual register) so that the values stored in those registers can be re-used across multiple basic blocks” (*id.* at 5), but argue “these portions of Mitran do not speak to any type of sign extension instruction,” (*id.*). We are not persuaded of error.

Claim construction is an issue of law that is reviewed *de novo*. *Cordis Corp. v. Boston Scientific Corp.*, 561 F.3d 1319, 1331 (Fed. Cir. 2009). We give claims their broadest reasonable interpretation consistent with the

Specification. *In re Am. Acad. of Sci. Tech. Ctr.*, 367 F.3d 1359, 1369 (Fed. Cir. 2004). Although claims are interpreted in light of the Specification, "limitations are not to be read into the claims from the [S]pecification." *In re Van Geuns*, 988 F.2d 1181, 1184 (Fed. Cir. 1993). Accordingly, arguments must be commensurate in scope with the actual claim language. *In re Self*, 671 F.2d 1344, 1348 (CCPA 1982).

Appellants' Specification discloses

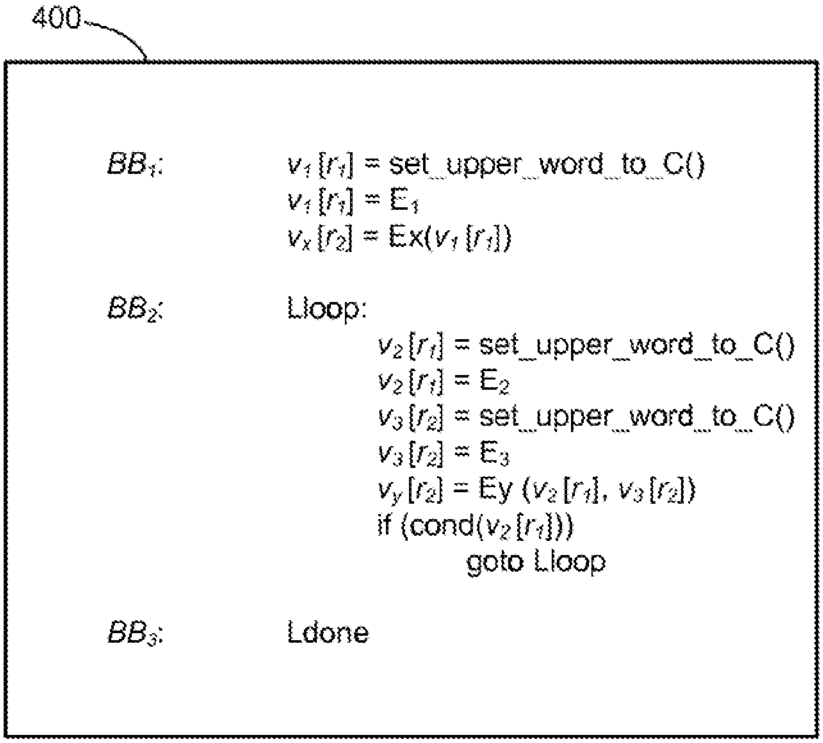
> the processor must first convert the 32-bit memory offset to a 64-bit memory offset so that the number of bits associated with the memory offset is in alignment with the number of bits associated with the 64-bit base address. Such conversion is referred to herein as "sign-extension," which, in particular, involves increasing the number of bits of a binary number while preserving the number's sign (i.e., positive / negative) and value.

Spec. ¶ 5. The term "instruction," in accordance with its plain meaning as known in the art, is "[a] statement or expression consisting of an operation and its operands (if any), which can be interpreted by a computer in order to perform some function or operation." THE AUTHORITATIVE DICTIONARY OF IEEE STANDARDS TERMS 563 (7th ed. 2000). Thus, a "sign extension instruction," under a broadest reasonable interpretation consistent with the Specification, is an operation for a computer to increase the number of bits of a binary number while preserving the number's sign.

Mitran is directed to "eliminating redundant operations establishing common properties using shared real registers." Mitran ¶ 2. Mitran initially determines a common property P to optimize. *Id.* ¶ 32. Mitran's example of "a common property for 32-bit non-negative expressions is that the upper 32 bits are always zero when the values are zero-extended to 64 bits." *Id.*

Mitran's Ep is a set of operations that establish the common property P for each element in a set of virtual registers (Vp), each Vp holding values with the common property P. *Id.* ¶ 33. Mitran discloses "where the common property P for a [set of] virtual registers is that [the] upper 32 bits are always zero, the set of operations Ep may include the sign extension operation that extends a 32-bit non-negative value to 64 bits." *Id.* Thus, Mitran's set of operations Ep, then, at least suggests a sign extension instruction.

Mitran's Figure 6 is reproduced below:



```
BB1:        v1[r1] = set_upper_word_to_C()
            v1[r1] = E1
            vx[r2] = Ex(v1[r1])

BB2:        Lloop:
                 v2[r1] = set_upper_word_to_C()
                 v2[r1] = E2
                 v3[r2] = set_upper_word_to_C()
                 v3[r2] = E3
                 vy[r2] = Ey (v2[r1], v3[r2])
                 if (cond(v2[r1]))
                      goto Lloop

BB3:        Ldone
```

Fig. 6

Mitran's Figure 6 illustrates an example of program code (**400**) "having real registers allocated for each of the variables contained therein." *Id.* ¶ 15.

Mitran's "program code **400** includes . . . a natural loop consisting of basic block BB_2. In this example, the common property P is that the upper 32-bit words of the 64-bit expressions E_1, E_2, and E_3 contain a constant C, while the upper 32-bit words of the 64-bit expressions Ex and Ey do not." *Id.* ¶ 51. Mitran's "set of instructions Ep establishing the common property P for the set Vp is: {v_1=set_upper_word_to_C( ), v_2=set_upper_word_to_C( ), and v_3 =set_upper_word_to_C( )}." *Id.*

Contrary to Appellants' arguments (Reply Br. 5), because Mitran's v_2 is included in the set of instructions Ep establishing the common property P, and because Ep at least suggests a sign extension instruction as discussed above, then Mitran's v_2 at least suggests a sign extension instruction. That is, despite Appellants' arguments that Mitran's v_2 (1) may be removed and replaced with a real register, and (2) stores assigned values that can be re-used across multiple basic blocks (Reply Br. 5), Mitran's v_2 is an Ep operation that extends a 32-bit non-negative value to 64 bits. Moreover, Appellants' argument that Mitran's virtual register is not moved to a loop pre-header (*id.*) is unavailing as not commensurate with the scope of the claim. Thus, Mitran at least suggests receiving program code 400 (the claimed "intermediate code") that is associated with a computer program and includes v_2 (the claimed "first SEI") that is included in Lloop of BB_2 (the claimed "loop structure") within a computer program.

Appellants do not persuade us of error in the rejection of claim 1. Accordingly, we sustain the rejections of (1) claim 1; (2) independent claims 10 and 19, which are argued relying on the arguments made for claim 1 (*see* App. Br. 11, 13); (3) claims 5–9, 14–18, and 21, which variously depend, directly or indirectly, from claims 1, 10, and 19, and were not separately argued with particularity (*see id.* at 15).

Claims 2–4, 11–13, and 20

We also sustain the Examiner's obviousness rejection of claim 2 (Final Act. 7), which recites an operand associated with the first SEI comprises 32-bits, and the first SEI, when executed by the processor, converts the operand to 64-bits but does not change the sign or value associated with the operand.

Appellants contend the Examiner errs for the following reason:

> Deneau explicitly discloses that, in order to preserve integer semantics (i.e., integer wraparound/overflow behavior), the instructions that convert the smaller-length integer into a 64-bit integer are required to (1) subtract an offset from a 64-bit integer and (2) add the offset to the smaller-length integer. See Deneau at ¶¶ [0006], [0017], [0030], and [0034]. The resulting value is then zero-extended to generate a 64-bit integer. Id. at ¶ [0030]. Thus, by stating that the value associated with the smaller-length integer is required to be changed – by adding an offset to the smaller-length integer – in order to convert the smaller-length integer into a 64-bit integer, Deneau is expressly teaching away from the approach recited in claims 2, 11, and 20, which requires that the sign-extension instruction (SEI) converts the operand to 64-bit but does not change the sign or value associated with the operand.

App. Br. 13–14. We are unpersuaded of error.

We find Deneau does not discuss changing or not changing the sign or value associated with an operand associated with an SEI that converts the operand from 32-bits to 64-bits. Thus, we do not see how Deneau criticizes, discredits, or otherwise discourages investigation into the recited operand conversion for teaching away. *See Norgren Inc. v. Int'l Trade Comm'n*, 699 F.3d 1317, 1326 (Fed. Cir. 2012) (citation omitted); *see also In re Kahn*, 441 F.3d 977, 990 (Fed. Cir. 2006).

Appellants contend, for the first time in the Reply Brief, that Mitran does not teach the limitations of claim 2. Reply Br. 7–8. According to Appellants, "Mitran fails to disclose how integer semantics, such as integer wraparound and/or overflow behavior, are preserved when a 32-bit non-negative expression is converted into a 64-bit value." *Id.* at 8.

These newly-raised arguments are waived as untimely. *See* 37 C.F.R. § 41.41(b)(2). The Examiner made a similar finding on page 7 of the Final Rejection to which Appellants could have responded in the Appeal Brief. Appellants' belated argument provides the Examiner no opportunity to respond to those arguments made only in the Reply Brief. *See Ex parte Borden*, 93 USPQ2d 1473, 1475 (BPAI 2010) (informative). Arguments not raised in an appeal brief and raised for the first time in a reply brief and "not responsive to an argument raised in the examiner's answer, including any designated new ground of rejection, will not be considered by the Board for purposes of the present appeal, unless good cause is shown." 37 C.F.R. § 41.41(b)(2).

Accordingly, we sustain the rejection of (1) claim 2; (2) claims 11 and 20, which were argued together with claim 2 (*see* App. Br. 13–14); and (3) claims 3, 4, 12, and 13, which variously depend, directly or indirectly, from claims 2 and 11, and were not separately argued with particularity (*see id.* at 15).

DECISION

The Examiner's decision to reject claims 1–21 under 35 U.S.C. § 103(a) is affirmed.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a). *See* 37 C.F.R. § 41.50(f).

AFFIRMED