2014-1076, -1317

IN THE
UNITED STATES COURT OF APPEALS
FOR THE FEDERAL CIRCUIT

HTC CORPORATION and HTC AMERICA, INC.,

Plaintiffs-Cross-Appellants,

v.

TECHNOLOGY PROPERTIES LIMITED,
PATRIOT SCIENTIFIC CORPORATION and ALLIACENSE LIMITED,

Defendants-Appellants,

Appeals from the United States District Court for the Northern District of California in Case No. 5:08-cv-00882-PSG, Judge Paul S. Grewal

PRINCIPAL BRIEF FOR DEFENDANTS-APPELLANTS TECHNOLOGY PROPERTIES LIMITED, PATRIOT SCIENTIFIC CORPORATION AND ALLIACENSE LIMITED

James C. Otteson
Thomas T. Carmack
Philip W. Marsh
AGILITY IP LAW, LLP
149 Commonwealth Dr.
Menlo Park, CA 94025
Telephone: (650) 227-4800
Facsimile: (650) 318-3483

Attorneys for Defendants-Appellants Technology Properties Limited, Patriot Scientific Corporation and Alliacense Limited

May 1, 2014

UNITED STATES COURT OF APPEALS FOR THE FEDERAL CIRCUIT

HTC CORPORATION and HTC AMERICA, INC. v. TECHNOLOGY PROPERTIES LIMITED, et al. Nos. 2014-1076, -1317

CERTIFICATE OF INTEREST

Counsel for the Appellants, Technology Properties Limited, Patriot Scientific Corporation and Alliacense Limited, certifies the following (use "None" if applicable; use extra sheets if necessary):

1. The full name of every party or amicus represented by me is:

Technology Properties Limited
Patriot Scientific Corporation
Alliacense Limited

2. The name of the real party in interest (if the party named in the caption is not the real party in interest) represented by me is:

Technology Properties Limited
Patriot Scientific Corporation
Alliacense Limited

3. All parent corporations and any publicly held companies that own 10 percent or more of the stock of the party or amicus curiae represented by me are:

None

4. The names of all law firms and the partners or associates that appeared for the party or amicus now represented by me in the trial court or agency or are expected to appear in this court are:

Agility IP Law, LLP:
James C. Otteson
Thomas T. Carmack
Philip W. Marsh
Brandon Baum
Michelle Breit

David Lansky
Irvin E. Tyan
Jedidiah Phillips
James R. Farmer
Vinh Pham

Kirby Noonan Lance & Hoge:
Charles T. Hoge

Farella Braun Martel LLP:*
Deepak Gupta
Eugene Mar
John Cooper
Jeffrey Fisher
Nan Joesten
Stephanie Powers Skaff

McDermott Will & Emery LLP:*
Sushila Chanana

Nixon Peabody LLP:*
Christopher Ogden
Ronald Lopez

* These firms were prior counsel for defendants that are now terminated out of the district court action and will not be appearing in the Federal Circuit.

May 1, 2014
Date

/s/ James C. Otteson
James C. Otteson
Attorney for Defendants-Appellants

TABLE OF CONTENTS

Page

TABLE OF AUTHORITIES

Cases

Statutes

Rules

Other Authorities

Statement of Related Cases

No appeal in this case was previously before this or any other appellate court.

Jurisdictional Statement

The district court had subject matter jurisdiction pursuant to 28 U.S.C. §§ 1331 and 1338(a) because the action arises under federal statutes relating to patents. This appeal is from the district court's October 3, 2013 final judgment, which was modified on January 21, 2014, and which disposes of all parties' claims. The notice of appeal was timely filed on November 4, 2013. This Court has appellate jurisdiction pursuant to 28 U.S.C. § 1295.

Statement of the Issues

(1) Whether the district court erred in finding that the doctrine of intervening rights applies to U.S. Patent No. 5,530,890 to preclude any claims of infringement before the date of the issuance of the reexamination certificate.

(2) Whether the district court erred in its construction of the phrase "separate direct memory access central processing unit" in, for example, claim 11 of U.S. Patent No. 5,530,890.

Statement of the Case

This is an appeal from the district court's January 21, 2014 order modifying judgment (A0148), January 21, 2014 order granting-in-part the motion of

Plaintiffs-Cross-Appellants HTC Corporation and HTC America, Inc. (collectively "HTC") to correct the judgment (A0145), and September 17, 2013 order granting HTC's motion for partial summary judgment of non-infringement of U.S. Patent No. 5,530,890 (the '890 patent), precluding any claims of infringement before the date of the issuance of the reexamination certificate (A0020). It is also an appeal from claim construction orders, which were issued on June 12, 2012 (A0036), December 4, 2012 (A0048), and August 21, 2013 (A0063), construing the phrase "separate direct memory access central processing unit" in, for example, claim 11 of the '890 patent.

On February 8, 2008, HTC filed this suit in the Northern District of California seeking a judicial declaration that four of the patents in the "MMP patent portfolio" – U.S. Patent Nos. 5,784,584 ("'584 patent"), 5,440,749 ("'749 patent"), 6,598,148 ("'148 patent"), and 5,809,336 ("'336 patent") – are invalid and/or not infringed. A0131. On November 21, 2008, TPL counterclaimed for infringement of the four patents at issue. *Id.*

On April 25, 2008, Defendants-Appellants Technology Properties Limited, Patriot Scientific Corporation and Alliacense Limited (collectively "TPL") filed two complaints in the Eastern District of Texas against HTC alleging infringement of the above four patents. *Id*. On June 4, 2008, TPL filed additional patent

infringement actions against HTC in the Eastern District of Texas asserting U.S. Patent 5,530,890 ("'890 patent"). *Id.*

On July 10, 2008, HTC amended its complaint before the California court, adding claims for declaratory relief with respect to the '890 patent. *Id*. On November 21, 2008, TPL counterclaimed for infringement of the '890 patent. *Id.* On February 23, 2009, the parallel Texas litigation was dismissed without prejudice following the California district court's decision to deny TPL's Motion to Dismiss, or in the Alternative, to Transfer Venue in the California action. *Id.* On March 25, 2010, the district court accepted the parties' stipulation to dismiss the '584 patent from the litigation. A0132.

On May 13, 2011 (after the March 1, 2011 *Ex Parte* Reexamination Certificate for the '890 patent), the district court issued an order limiting TPL to the assertion of claims 11, 12, 13, 17 & 19 of the '890 patent (added during reexamination), and barring the assertion of previously asserted claims 7 and 9 – even though those claims were ***not*** changed during reexamination. A0069-A0070. The district court reasoned: "Apart from one clarification to independent claim 11, new claims 11-20 track the patent's original claims 1-10 word-for-word." A0070. The court noted that "TPL concedes that '[. . .] the scope of new independent claim 11 is the same as the scope of original independent claim 1.'" Thus, the district court rejected TPL's attempt to assert claims 7 and 9 as "redundant." A0075.

On July 17, 2013, the court accepted the parties' stipulation to dismiss the '148 and '749 patents from this litigation. A0132.

In a September 17, 2013 order on summary judgment, the district court ruled that the doctrine of absolute intervening rights – raised by HTC for the first time at summary judgment – applied to the '890 patent to preclude any infringement of claims 11, 12, 13, 17, 19 before the date of the issuance of the reexamination certificate. A0020. Because of that order, TPL agreed to enter into a stipulation with HTC to dismiss all claims relating to the '890 patent from this litigation. A0132. On September 19, 2013, the court accepted the parties' stipulation. Therefore, only claims relating to the '336 patent went to trial. *Id.*

From September 23 to October 1, 2013, a jury trial was held to consider whether HTC infringed the '336 patent. A7298-8968. At trial, HTC did not contest the validity of the '336 patent. *Id.*

On October 3, 2013, after two days of deliberations, the jury found that HTC and its accused products literally infringed all asserted claims of the '336 patent: 6, 7, 9, 13, 14, and 15. A0125-0128. As to damages for the accused products, the jury found that TPL was entitled to, as a reasonable royalty for infringement, a one-time (lump sum) payment of $958,560 for the life of the patent. *Id*. The district court entered judgment in favor of TPL. A0148. The court dismissed the '890 patent pursuant to its September 19, 2013 order. *Id.*

On January 21, 2014, the district court denied HTC's renewed motion for judgment as a matter of law that its products did not infringe the '336 patent. A0130-0144. These cross-appeals follow.

Statement of the Facts

I. The Claim Language Added in the Reexamination Only Clarified – but Did Not Substantively Change – the Scope of the '890 Patent.

The '890 patent first issued on June 25, 1996. A0280; A0007. On January 19, 2009, the'890 patent was subject to an *ex parte* reexamination, and an amended version of the patent emerged on March 1, 2011. A0007; A5974. One reason the PTO granted an *ex parte* reexamination was because "the specific allowable features of claims 1-10 of the '890 Patent in the original prosecution [were] ***unclear***." *See* A6195 (emphasis added). Specifically, the Examiner was concerned with a prior art reference ("May") that discussed an "instruction pointer," whereas the '890 patent discussed a "stack pointer." *See, e.g.*, A6047. Under the broadest interpretation – one that would be summarily rejected by a person of ordinary skill in the art – if the *instruction* pointer were allowed to point to ***any*** position in ***any*** stack, it could possibly be considered a *stack* pointer. A6041-6042. Accordingly, the Examiner advised the patentee to make ***explicit*** the ***implicit*** assumption that the stack pointer pointed to the first push down stack: the PTO "suggested to amend Claim 1 to ***clearly*** associate the stack pointer with the first push down stack" because "the stack pointer of Claim 1 was not ***explicitly***

associated with the first push down stack of Claim 1." A6042 (emphasis added). Following this suggestion, the patentee added claim 11 by copying the language of claim 1 and adding the phrase "said stack pointer pointing to said first push down stack," thereby clarifying the association of the stack pointer and the first push down stack. A0007-0008; A5974-5975. Aside from the added phrase, claim 11 was identical to original claim 1. *Id*.

II. The '890 Patent Discloses both a DMA Co-Processor And a Traditional DMA Controller.

The '890 Patent relates to microprocessor architecture and claims a direct memory access (DMA) mechanism. A0052. The '890 patent discloses two distinct DMA embodiments. The first embodiment, shown in Figs. 1-8, includes a microprocessor 50 with a DMA CPU 72 (detailed in Fig. 5), which is a DMA co-processor that has "the ability to fetch and execute instructions [that] operates as a co-processor to the main CPU." *See* '890 (A0280), 8:1; Fig. 5; 8:22-24. In contrast, the second embodiment (shown in Fig. 9) includes a different microprocessor 310 with a "more traditional DMA Controller 314," which has replaced the DMA CPU 72 "[t]o keep chip size as small as possible" because the microprocessor 310 is "on an already crowded DRAM die." The specification refers to the traditional DMA 314 interchangeably as "DMA controller 314" or "DMA CPU 314." *See* '890 12:61-65; Fig. 9; 10:52; 13:3-4. The district court's construction of "DMA CPU," which requires a unit capable of fetching and

executing instructions independently from the CPU, includes DMA CPU 72 of the first embodiment, but excludes DMA CPU 314 of the second embodiment.

III. Per the Restriction Requirement, the '890 Patent is Drawn to a Microprocessor System That Can Have a Traditional DMA Controller or a DMA Co-Processor.

The '890 and '336 patents derive from the same original patent application that was subject to a ten-way restriction requirement, and eventually resulted in six different patents known as the Moore Microprocessor Portfolio patents, all of which share a common specification. A0052. Claim 13 of the original application was specifically directed to the DMA co-processor invention, which required the DMA processing unit to fetch its own instructions. A2143. By contrast, originally filed claim 48 (which eventually became claim 1 of the '890 patent and later claim 11 of the reexamined '890 patent, the claim at-issue here) recited the use of a traditional DMA ***controller***, which is not required to fetch its own instructions. A2153-2154.

Due to the restriction requirement, originally filed claim 13 (the DMA co-processor invention) was in Group III, constituting inventions "drawn to a microprocessor system having a DMA for fetching instruction[s] for a CPU and itself." A2172-2173. The Group III application, No. 08/480,015, was eventually abandoned. The claims that eventually issued as the '890 patent were in Group

VIII, and divided into application No. 08/480,206. Per the restriction requirement, those claims were "drawn to a microprocessor architecture." *Id.*

Summary of the Argument

The district court committed reversible error by granting summary judgment with respect to the '890 patent based on the affirmative defense of absolute intervening rights, and by narrowly construing the term "separate DMA CPU."

With respect to intervening rights, the district court erred by allowing HTC to raise the affirmative defense for the first time in a motion for summary judgment filed a couple of months before trial. Had TPL known that HTC intended to assert the affirmative defense of absolute intervening rights, it would have conducted extensive discovery on the issue to marshal additional evidence establishing that the defense was inapplicable. Instead, TPL was blindsided and was unable to defend against HTC's motion properly, thereby resulting in partial summary judgment in HTC's favor. Additionally, when TPL sought to amend its infringement contentions to include two claims that were dependent on claim 1, HTC argued that the scope of claim 1 and claim 11 was the same, and thus the claims TPL sought to add were unnecessary because they were mirror images of two claims dependent on claim 11. The district court agreed, and rejected the amendment as redundant. Had HTC argued that the scope of claims 1 and 11 was different, as it did on the eve of trial, either the claims would have been added (if

the scope was different), or HTC's intervening rights defense would have been rejected (if the scope was the same). HTC, however, adopted inconsistent positions at TPL's expense. TPL was severely prejudiced by HTC's delay, which should have resulted in a waiver of the affirmative defense.

The district court should have rejected HTC's last-minute intervening rights defense for substantive reasons as well. The intervening rights doctrine essentially serves to protect an innocent infringer when the scope of a patent's claims are substantially changed during reexamination by limiting the infringer's liability to the period beginning on the date the patent emerges from reexamination. Here, however, as the parties agreed in 2011 in connection with TPL's motion to amend its infringement contentions, the only relevant change to the '890 patent was the addition of a phrase for mere clarification. Because the reexamined claim has a scope identical to the original claim, HTC was on notice of the scope of its infringement, and the doctrine of intervening rights does not apply. Accordingly, the district court erred in granting HTC's motion regarding intervening rights.

With respect to the construction of the term "separate DMA CPU," the district court erred by improperly limiting the term to only a DMA co-processor, excluding a traditional DMA controller. The specification discloses both, and uses the term DMA CPU to refer to both. In addition, during prosecution, the USPTO restricted the invention that requires a DMA co-processor to a separate application

that is *not* the application that resulted in the '890 patent. Therefore, "separate DMA CPU" should be construed as "electrical circuit for reading and writing to memory that is separate from a main CPU."

Argument

I. Standard of Review.

This Court reviews a district court's decision granting summary judgment *de novo*, reapplying the same standard applied by the district court. *Iovate Health Scis., Inc. v. Bio–Engineered Supplements & Nutrition, Inc*., 586 F.3d 1376, 1380 (Fed. Cir. 2009). Summary judgment is appropriate only "if the movant shows that there is no genuine dispute as to any material fact and the movant is entitled to judgment as a matter of law." Fed. R. Civ. P. 56(a).

The affirmative defense of absolute intervening rights allows an infringer, because of the infringer's pre-reexamination activity, to enjoy a personal intervening right to conduct that would have otherwise been an infringing activity before the reexamination.[1] *See Seattle Box Co., Inc. v. Indus. Crating & Packing Inc*., 756 F.2d 1574, 1579 (Fed. Cir. 1985). However, to entertain a defense of

[1] The first and second paragraphs of 35 U.S.C. § 252 define both absolute and equitable intervening rights, respectively. However, HTC only asserts a defense of absolute intervening rights here.

In addition, Section 307(b) provides that the rules established for reissued patents in Section 252 apply to reexamined patents. *Bloom Eng'g Co., Inc., v. N. Am. Mfg. Co., Inc*., 129 F.3d 1247, 1249 (Fed. Cir. 1997).

intervening rights, a court must first determine if "the accused product or activity infringes a claim that existed in the original patent and remains 'without substantive change' after reissue." *Marine Polymer Techs., Inc. v. HemCon, Inc.*, 672 F.3d 1350, 1362 (Fed. Cir. 2012). "[I]n determining whether substantive changes have been made, we must discern whether the *scope* of the claims are identical, *not merely whether different words are used*." *Marine Polymer*, 672 F.3d at 1373 (citing *Laitram Corp. v. NEC Corp*., 163 F.3d 1342, 1346 (Fed. Cir. 1998) ("*Laitram IV*") (emphasis in original). HTC bears the burden to establish its affirmative defense of intervening rights, with all alleged facts viewed in the light most favorable to TPL, the non-movant. *See Linear Tech. Corp. v. Micrel, Inc*., 524 F. Supp. 2d 1147, 1156 n.4 (N.D. Cal. 2005) (citing *Jazz Photo Corp. v. Int'l Trade Comm'n*, 264 F.3d 1094, 1102 (Fed. Cir. 2001)); *Pitney Bowes, Inc. v. Hewlett-Packard Co.*, 182 F.3d 1298, 1304 (Fed. Cir. 1999).

Because this inquiry requires interpretation of the scope of the claims as they existed pre- and post-reexamination, the Court reviews *de novo* a district court's conclusion whether the claims that emerged from reexamination are substantively the same as the original claims. *Laitram IV,* at 1346–47; *Cybor Corp. v. FAS Techs., Inc*., 138 F.3d 1448, 1451 (Fed. Cir. 1998) (en banc). *See, also, Minco, Inc. v. Combustion Eng'g, Inc*., 95 F.3d 1109, 1115, 40 U.S.P.Q.2d 1001, 1005 (Fed. Cir. 1996) ("This court reviews without deference the district court's

conclusion that the reexamined claims remained identical in scope") (citation omitted).

This rule flows from the general principle that "the interpretation and construction of patent claims, which define the scope of the patentee's rights under the patent, is a matter of law exclusively for the court." *See Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 970–71 (Fed. Cir. 1995) (en banc). *See also Cybor Corp.*, 138 F.3d at 1456 ("reaffirm[ing] that, as a purely legal question, we review claim construction *de novo* on appeal including any allegedly fact-based questions relating to claim construction"); *Lighting Ballast Control LLC v. Philips Electronics N. Am. Corp.*, 744 F.3d 1272, 1310 (Fed. Cir. 2014) (affirming *de novo* review of claim construction as a matter of law).

II. The District Court Erred in Finding that the Doctrine of Intervening Rights Applies to U.S. Patent No. 5,530,890 to Preclude Any Claims of Infringement Before the Date of the Issuance of the Reexamination Certificate.

A. The District Court erred by allowing HTC to raise the affirmative defense of intervening rights for the first time on the eve of trial.

Not only did the district court err by granting HTC's motion for partial summary judgment based on the affirmative defense of absolute intervening rights (as discussed below), it erred by entertaining HTC's belatedly asserted defense in the first place. "Intervening rights, as provided under 35 U.S.C. § 252, is an affirmative defense." *Underwater Devices Inc. v. Morrison-Knudsen Co., Inc.*,

717 F.2d 1380, 1389 (Fed. Cir. 1983), *overruled on other grounds, In re Seagate Tech., LLC*, 497 F.3d 1360 (Fed. Cir. 2007). As such, HTC was required to raise the defense of intervening rights in its responsive pleading. Fed. R. Civ. P. 8(c) ("In responding to a pleading, a party must affirmatively state any avoidance or affirmative defense."). "Failure to plead an affirmative defense is a waiver of that defense." *Underwater Devices Inc.*, 717 F.2d at 1389.

As the district court recognized, HTC failed to plead its affirmative defense of intervening rights in its answer to Defendants' counterclaim. A0018. Nor did HTC amend its answer after the '890 patent reissued on March 1, 2011 following reexamination. *Id*. Indeed, the first time HTC asserted the affirmative defense of intervening rights was when it moved for summary judgment roughly four years into the litigation. *Id*. Accordingly, HTC's failure to raise its affirmative defense until the eve of trial should be deemed a waiver. *Underwater Devices Inc.*, 717 F.2d at 1389; Fed. R. Civ. P. 8(c).

Although the Ninth Circuit has somewhat "liberalized" this pleading requirement, affirmative defenses may be raised for the first time on summary judgment "only if the delay does not prejudice the plaintiff." *Magana v. Commonwealth of the N. Mariana Islands*, 107 F.3d 1436, 1446 (9th Cir. 1997). Here, TPL was unfairly prejudiced by HTC's failure to assert its affirmative defense until mere months before trial. Had HTC promptly raised the issue of

intervening rights back in 2011 (as opposed to 2013), TPL would have conducted the litigation differently by, among other things, taking discovery relating to the issue. Specifically, as discussed below, the amendment to claim 11 of the '890 patent served to clarify the claim language to make plain what a person of ordinary skill in the art would have already understood when reading the claim. Had TPL known that HTC intended to argue that the added language substantively altered the claim, TPL would have conducted discovery (*e.g.*, from experts) to elicit additional evidence to show that was not the case. By ambushing TPL with its affirmative defense on the eve of trial and well after the close of fact and expert discovery, TPL was deprived of the opportunity to fully develop its response to HTC's intervening rights defense. Given that the district court granted HTC's motion for partial summary judgment based on intervening rights, resulting in the judgment dismissing the '890 patent, the prejudice to TPL is evident. The district court's decision to allow HTC to raise its affirmative defense of absolute intervening rights for the first time on summary judgment should thus be reversed.

TPL was also prejudiced by HTC's delay in raising its intervening rights defense because TPL was unfairly precluded from asserting claims 7 and 9 of the '890 patent – which did ***not*** change during reexamination. In May 2011, the district court denied TPL's motion to amend its infringement contentions to assert claims 7 and 9 of the '890 patent. A0069-A0070. HTC had opposed TPL's

motion to amend, arguing: "TPL's attempt to assert claims 7 and 9 of the '890 patent in addition to claims 17 and 19 is unnecessary and prejudicial because the latter are mirror images of the former." A0075.

As the district court recognized: "Apart from one clarification to independent claim 11, new claims 11-20 track the patent's original claims 1-10 word-for-word." A0070. Noting that "TPL concedes that '[b]ecause the scope of new independent claim 11 is the same as the scope of original independent claim 1,'" the district court rejected TPL's proposed amendment as "redundant." A0075. Had HTC raised its intervening rights defense then – by arguing that the scope of claim 11 was substantively ***different*** than the scope of claim 1 – the district court may not have barred TPL from asserting claims 7 and 9 as "redundant" of claims 17 and 19. If the scope of claim 1 (from which claims 7 and 9 depend) is ***different*** from the scope of claim 11 (from which claims 17 and 19 depend), then the scope of claims 7 and 9 necessarily differ from the scope of claims 17 and 19.

But at the time of TPL's motion to amend in 2011, both HTC and the district court argued – and TPL acknowledged – that the scope of independent claims 1 and 11 was ***identical***. On the eve of trial more than two years later, HTC argued that the scope of claims 1 and 11 was ***different*** – in a last ditch effort to avoid infringement of the '890 patent. Quite clearly, HTC's gamesmanship has unfairly prejudiced TPL. Had HTC maintained its earlier position that the scope of claims

1 and 11 was the same, it would not have prevailed on its last minute summary judgment motion regarding intervening rights. Otherwise, if HTC had not opposed TPL's effort to add claims 7 and 9 to the case in 2011 – and because those claims did ***not*** change during reexamination – TPL would have been able to proceed to trial regarding both HTC's pre-reexamination infringement of claims 7 and 9, as well as HTC's post-reexamination infringement of those claims. Allowing HTC to play both sides, however, has plainly resulted in unfair prejudice to TPL.

B. The district court erred in finding that the claim language added in the reexamination substantively changed the scope of the '890 patent.

The district court's grant of partial summary judgment in HTC's favor with respect to the '890 patent was premised on its determination that language added to Claim 11[2] narrowed the scope of the claims, thereby precluding any claims of infringement before the issuance of the reexamination certificate. A0020. The district court's ruling was in error. Viewing all facts in the light most favorable to TPL, HTC failed to carry its burden to establish a change in the scope of the claims. Thus, the district court should have denied HTC's motion.

"A determination of whether the scope of a [reexamination] claim is identical with the scope of the original claim is a question of law." *Westvaco*

[2] Claims 12, 13, 17, and 19 all depend from independent claim 11. A0018. Accordingly, the analysis of the scope of claim 11 likewise applies to the scope of claims 12, 13, 17, and 19.

Corp. v. Int'l Paper Co., 991 F.2d 735, 741 (Fed. Cir. 1993). "Reexamined claims are 'identical' to their original counterparts if they are 'without substantive change.'" *Laitram IV*, 163 F.3d 1342, 1346 (citing *Seattle Box Co., Inc. v. Industrial Crating & Packing Inc.*, 731 F.2d 818, 827-28. "[I]n determining whether substantive changes have been made, we must discern whether the scope of the claims are identical, *not merely whether different words are used*." *Marine Polymer*, 672 F.3d at 1373 (citing *Laitram IV*, 163 F.3d at 1346) (emphasis in original). Thus, "substantive change" does not include minor word changes. *Slimfold Mfg. Co. Inc. v. Kinkead Indus., Inc.*, 810 F.2d 1113, 1115 (Fed. Cir. 1987). Further, "a claim amendment made during reexamination following a prior art rejection is not *per se* a substantive change. Rather, to determine whether a claim change is substantive, it is necessary to analyze the claims of the original and the reexamined patents in light of the particular facts, including the prior art, the prosecution history, other claims, and any other pertinent information . . . [including] an overall examination of the written description . . . ." *Laitram IV*, 163 F.3d at 1347-48 (internal citation and quotation omitted).[3] Moreover, an

[3] *See also, e.g., Austin v. Marco Dental Prods., Inc*., 560 F.2d 966, 973 (9th Cir. 1977) (no intervening rights when claims are "substantially identical"); *Akron Brass Co. v. Elkhart Brass Mfg. Co*., 353 F.2d 704, 708-09 & n. 5 (7th Cir. 1965) (substitution of "outlet" for "inlet" found not to be substantial change); *Richmond Eng'g Co. v. Bowser, Inc*., 264 F.2d 595, 597-98 (4th Cir. 1959) (despite rephrasing and rearrangement of elements of claim, "scope of the [reissue] claims

amendment that clarifies the text of a claim "to make specific what was always implicit or inherent," or that "makes it more definite without affecting its scope" is not a substantive change. *Laitram Corp. v. NEC Corp. ("Laitram I")*, 952 F.2d 1357, 1361 (Fed. Cir. 1991); *Bloom Engineering*, 129 F.3d at 1250.

Claim 11 of the reexamined patent is identical in scope to original claim 1 because the only change in language provides explicit clarification of a preexisting implicit limitation. As HTC conceded, "TPL added claim 11 during the reexamination by ***copying the language from claim 1,***" with the ***only change*** being the addition of the phrase, "said stack pointer pointing to said first push down stack." A5974-5975 (emphasis added). As evidenced by the prosecution history, this amendment was added only for the sake of clarification, and did not result in any substantive change. [4]

[was] substantially identical"); *Foxboro Co. v. Taylor Instrument Cos.*, 157 F.2d 226, 228 (2nd Cir. 1946) ("identical" interpreted to mean "substantially identical"); *Greer Hydraulics, Inc. v. Rusco Indus., Inc.*, No. 72-961-JWC, 1974 WL 20255, 185 U.S.P.Q. 83, 85 (C.D. Cal. Nov. 13, 1974) (no intervening rights because claims found "substantially identical" to those in original patent despite correction of spelling, clarification of language, and addition of word "activation" to conform to earlier reference in claim).

[4] Indeed, the PTO granted *ex parte* reexamination *inter alia* because "the specific allowable features of claims 1-10 of the '890 Patent in the original prosecution [were] ***unclear***." *See* A6195 (emphasis added).

Specifically, during reexamination, the examiner's primary concern with claim 1 was centered on the discussion in a prior art reference ("May") of an *instruction* pointer pointing into the *instructions* of a process, whereas the original specification of the'890 patent clearly described a ***stack*** pointer pointing into the parameter stack.[5] However, under the broadest interpretation, if the *instruction* pointer were allowed to point to ***any*** position in ***any*** stack, it could possibly be considered a *stack* pointer. A6041-6042. A person of ordinary skill in the art, however, would recognize that such a hypothetical situation is impossible.

An instruction pointer (*i.e.*, instruction address register, or program counter) points to the location in the instruction memory of the next instruction to be fetched and executed. As such, it is under program control (*i.e.*, controlled by the processor) and is not modifiable by the user. Thus, it does not fit the definition of a stack. One of ordinary skill in the art would understand that instructions are kept in the instruction memory, which is randomly accessible, and ***not*** located in a

[5] *See, e.g.*, A6047 ("Patent Owners respectfully submit that the instruction pointer of May '948(IPTR 50) does not 'point' to the stack registers A, B and C by virtue of being bidirectionally connected thereto or because it loads the contents of A during a general call. Rather, IPTR 50 holds the memory address of the next instruction to be executed and thus points to the location in memory of that next instruction, as stated by May '948. It is not a stack pointer to the stack (A, B, C) of May '948. Indeed, IPTR 50 is not a pointer to <u>any</u> stack.").

stack. To avoid any potential problems in program execution (such as a program crash), any modification of the instructions is forbidden.[6] A6546-6547 at ¶ 57.

Thus, the Examiner "suggested to amend Claim 1 to ***clearly*** associate the stack pointer with the first push down stack" because "the stack pointer of Claim 1 was not ***explicitly*** associated with the first push down stack of Claim 1." A6042 (emphasis added). *See also* A6043 (regarding the "relationship of the stack pointer to the push down stack," "[t]he Patent Owner suggested possibly amending the claim to clarify this point, which the examiner would consider.").

Accordingly, the patentees amended Claim 1 to explicitly associate the stack pointer with the first push down stack – "Claim 11 recites (in part) 'said stack pointer pointing into said first push down stack,' clarifying the association of the stack pointer and the first push down stack." A6054. This was a clarification, not a substantive change.

The claims of the '890 patent describe two stacks – a first push down stack and a return push down stack – each having its own purpose. The first stack (the

---

[6] *See also* A6048 (U.S. Patent No. 4,758,948 to May describes a multi-processor that includes "[e]ach workspace, e.g., workspace L below, includes a number of variables, a machine state 67, a link 66 to the next workspace in a queue of workspaces, ***and an instruction pointer IPTR S 65 that points to the next instruction for that process***. . . . When a process, such as L, is activated, the computer loads the computer's program counter IPTR 50 with the contents of IPTR S 65.") (emphasis added).

first push down stack) is used to store parameters and data that are being operated upon. The second stack (the return push down stack) is used for storing return addresses when the program jumps into subroutine calls. Each stack pointer is associated with a particular stack and constitutes an integral part of the stack. Interchanging the stacks (or content of the stacks) is not permitted because that would create an unpredictable and unmanageable situation and would cause the program to crash. Thus, if the stack pointer did not point into the first push down stack, but instead pointed into the return push down stack, the same undesirable situation (a crash) would occur. A6547 at ¶ 58. Reexamined claim 11 clarified this situation by stating what is obvious to someone of ordinary skill in the art – that stack pointers can only point to the stack with which they are associated. *Id*.

Original claim 1 of the ’890 patent refers to a stack pointer, a first push down stack, a return stack pointer, and a return push down stack. The district court correctly recognized that “where the stack pointer points matters,” but incorrectly hypothesized that the stack pointer might point to ***other*** push down stacks. A0020. As described above, one of ordinary skill in the art would understand that the stack pointer would point to the first push down stack, and ***only*** to the first push down stack. Although it is possible to claim the stack pointer as pointing into the *return* push down stack, one of ordinary skill in the art would have recognized instead that it was implicit that the *return* stack pointer (and ***only*** the return stack pointer)

would point into the *return* push down stack, and the stack pointer would ***only*** point into the first push down stack. Any other arrangement would be untenable as it would cause the program to crash.

While the specification discusses a second push down stack, a second push down stack is not an element of any of the original or reexamined claims.[7] Accordingly, it would be obvious to one of ordinary skill that the stack pointer would point into the only other available push down stack, namely the first push down stack. A6546-6547 at ¶¶ 56-58. In no way does the added language change the scope of the original claim. Rather, it merely clarifies that already implicit association, and consequently cannot be considered a substantive change. Because this limitation was inherent in the original claims of the '890 patent, HTC has no intervening rights with respect to its infringement of claim 11.[8] HTC failed to present evidence to dispute how a person of ordinary skill would have understood these issues – an issue of fact underlying the ultimate question of law. Accordingly, HTC failed to carry its burden, and the district court's grant of partial summary judgment in favor of HTC was in error.

---

[7] As the applicant noted during reexamination, the specification also "defines the stack pointer as a pointer into the parameter stack, which . . . is also denominated the push down stack." A6054.

[8] Nor does it have intervening rights with respect to its infringement of Claims 12, 13, 17, and 19, all of which depend from independent claim 11. A0018.

III. The District Court Erred in Its Construction of the Phrase "Separate Direct Memory Access Central Processing Unit."

The district court construed the term "separate direct memory access central processing unit" ("separate DMA CPU") to mean "a central processing unit that accesses memory and **that fetches and executes instructions** directly and separately of the main central processing unit." A0063 (emphasis added). However, the court's construction only accounted for the first of two embodiments described in the '890 patent's specification. As a result, the court's construction of DMA CPU limited that term to that first embodiment – a DMA co-processor – and improperly excluded the second embodiment: a traditional DMA controller.

In addition, early on in the prosecution of the parent application that led to the '890 patent (and the entire MMP portfolio), the USPTO issued a ten-way restriction requirement requiring the applicant to pursue ten different divisional applications. Most relevant here, the USPTO restricted the invention that requires a DMA ***co-processor*** to a separate application that is ***not*** the application that resulted in the '890 patent. In other words, while the district court was correct that the specification describes, among other things, a DMA co-processor, ***that*** invention was pursued in a ***different*** application. The '890 application was specifically restricted to a microprocessor architecture that can include either a ***traditional*** *DMA* controller or a DMA co-processor.

In light of that history, and for other reasons discussed below, the district court's construction – which excludes a traditional DMA controller – is incorrect. Therefore, "separate DMA CPU" should be construed as "electrical circuit for reading and writing to memory that is separate from a main CPU."

A. The intrinsic record as a whole supports Appellants' proposed construction.

The intrinsic record supports Appellants' proposed construction of DMA CPU, which encompasses both embodiments of the DMA CPU described in the '890 patent's specification. The specification discloses and describes two embodiments involving two different microprocessors, each of which involves a distinct type of DMA CPU.

The first embodiment involves microprocessor 50 (shown in connection with Figures 1-8 of the '890 patent), which includes DMA co-processor 72. "Details of the DMA CPU 72 are provided in FIG. 5." '890 8:1. The specification expressly states that the DMA CPU 72 "operates as a co-processor to the main CPU." '890 8:23-24. In particular, the specification observes that the "DMA CPU 72 controls itself and has the ability to fetch and execute instructions." '890 8:22-23.

The specification also describes a second embodiment with a different microprocessor 310 and a traditional DMA controller 314, which differs from the DMA co-processor 72. The specification expressly identifies this second embodiment in connection with Fig. 9. '890 patent at 4:60-62 ("FIG. 9 is a layout

diagram of ***a second embodiment of a microprocessor in accordance with the invention*** in a data processing system on a single integrated circuit.") (emphasis added). In particular, the specification notes that the microprocessor 310 is used on "an already crowded DRAM die 312." '890 8:61-62. Because of the tight quarters associated with this second embodiment, the patent notes: "To keep chip size as small as possible, the DMA processor 72 of the microprocessor 50 ***has been replaced with a more traditional DMA controller 314***." '890 8:62-65 (emphasis added). In other words, the second embodiment uses a more traditional DMA controller, which is shown as DMA CPU 314 in Fig. 9.

The district court's construction of the term "DMA CPU" encompassed the first embodiment described in the '890 patent, but excluded the second embodiment. In other words, the court's construction included the DMA CPU 72 of Figures 1-8, "which controls itself and has the ability to fetch and execute instructions" ('890 8:22-23), but did ***not*** include the DMA CPU 314, which is "a more traditional DMA controller" and functions only when "used with the microprocessor 310" in a way "supported by the microprocessor 310." *See* '890 8:62-65; 12:65-13:12.

The district court reasoned that because a DMA controller is distinct from a DMA CPU, where the patent claims a DMA CPU, it means a DMA CPU and not a DMA controller. A0062 (Final Claim Construction Order at 13). However, as the

court itself acknowledged, the more traditional DMA controller associated with the second embodiment is referred to interchangeably as "DMA controller 314" or "***DMA CPU 314***" in the specification. *Id.*; *see, e.g.*, '890 12:61-65; Fig. 9; 10:52; 13:3-4. But unlike the DMA co-processor DMA CPU 72 of the first embodiment, the DMA CPU 314 of the second embodiment is ***not*** described as fetching or executing its own commands independent of the CPU. *Cf.* '890 8:22-23. Instead, the specification states that the DMA CPU 314 of the second embodiment "is used ***with the microprocessor 310***" to perform certain functions, such as video output, multiprocessor serial communications, and 8-bit parallel I/O. '890 12:65-13:2 (emphasis added). The specification makes clear that this "more traditional DMA controller 314" of the second embodiment—unlike the self-sufficient DMA co-processor 72 of the first embodiment—can only function when controlled by and "used with the microprocessor 310" to accomplish functions "supported by the microprocessor 310." '890 12:65-13:12. Thus, the term "DMA CPU" in claim 11 encompasses DMA CPU 314, which does ***not*** fetch or execute instructions.

The district court's first claim construction order (June 12, 2012) reasoned that, because the phrase being construed included the term "central processing unit" or CPU, it would be understood to mean a unit of a computing system that fetches, decodes, and executes programmed instructions. A0035 (First Claim Construction Order at 12 & n.26) (citing Modern Dictionary of Electronics 107

(7th ed. 1999)). The court then observed that the '890 specification used the term CPU consistently with that meaning, citing a selection in the patent that referred to the DMA CPU 72 of the first embodiment. A0035 at n.27 (citing '890 8:22-24). However, the court's reasoning relied exclusively on the DMA of the first embodiment (DMA CPU 72) without regard to the second embodiment, which replaces "the DMA processor 72 . . . with a more traditional DMA controller 314." '890 12:61-65. Although the court correctly recognized that the DMA CPU 72 of the first embodiment is described consistently with the dictionary definition of CPU that it relied on, the court failed to account for the DMA CPU 314 of the second embodiment, which is different.

The court also correctly recognized that the DMA CPU 72 of the first embodiment was considered advantageous because it "does not require use of the main CPU during DMA requests and responses . . . which provides very rapid DMA response with predictable response times." A0035 (First Claim Construction Order at 12 & n. 29). However, the court's reasoning only considered the specification's discussion of the advantages of the first embodiment, which uses the DMA CPU 72, and ignored the second embodiment of the invention described in connection with Figure 9, which uses "a more traditional DMA controller 314," instead of the DMA co-processor DMA CPU 72. *See* '890 12:61:65.

The patent states that the second type of microprocessor 310 of the second embodiment (distinguished from microprocessor 50 of the first embodiment) is used because it resides on a more crowded DRAM die." *Id.* Because of this more crowded arrangement, the second embodiment uses the more traditional DMA controller 314 "[t]o keep chip size as small as possible." *Id.* This is directly in line with several "objects" of the invention recited in the specification but not explicitly acknowledged in the court's claim construction order. *See e.g.*, '890 1:61-63 ("to provide a microprocessor with a reduced pin count and cost compared to conventional microprocessors"); '890 1:65-67 ("to provide a high performance microprocessor that can be directly connected to DRAMs without sacrificing microprocessor speed"). Indeed, the second embodiment shown in Figure 9 is presented as a "solution to the bandwidth/bus path problem" associated with the microprocessor 50 of the first embodiment. *See* '890 8:60-9:1.

Patent claims "must be read in view of the specification, of which they are a part." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1315 (Fed. Cir. 2005) (citing *Markman v. Westview Instruments, Inc.*, 52 F.3d 967 at 979. The "proper definition is the 'definition that one of ordinary skill in the art could ascertain from the intrinsic evidence in the record.'" *Id.* at 1314. One of ordinary skill in the art

would understand the term DMA CPU to be consistent with the two embodiments of the '890 patent discussed above. *See Phillips*, 415 F.3d at 1315-16.[9]

Appellants respectfully submit that the intrinsic record in this case—including the specification and the file history (which includes the restriction requirement and reexamination proceeding discussed below)—supports Appellants' proposed construction for DMA CPU: "electrical circuit for reading and writing to memory that is separate from a main CPU." Only this construction avoids improperly limiting the claim to a particular embodiment. On the other hand, the district court's construction improperly limits the claim to the first embodiment. *Phillips*, 415 F.3d at 1323 ("[A]lthough the specification often describes very specific embodiments of the invention, we have repeatedly warned against confining the claims to those embodiments.") (citations omitted); *see also DSW, Inc. v. Shoe Pavilion, Inc.*, 537 F.3d 1342, 1348 (Fed. Cir. 2008) ("[W]hen

[9] It is fundamental that an inventor may act as his or her own lexicographer. Thus, the district court's reliance on a technical dictionary definition that is consistent with one embodiment but inconsistent with another is misplaced. *See* A0035 (First Claim Construction Order at 12 n.26) (citing Modern Dictionary of Electronics 107 (7th ed. 1999)). Indeed, this Court has stated that in cases where the specification reveals that the inventor gave a claim term a special meaning that differs from the meaning it might otherwise possess, "***the inventor's lexicography governs***." *Phillips*, 415 F.3d at 1318 (emphasis added), citing *CCS Fitness, Inc. v. Brunswick Corp.*, 288 F.3d 1359, 1366 (Fed. Cir. 2002). Consistent with this principle, the Court has "repeatedly warned against confining the claims to [particular] embodiments." *Phillips*, 415 F.3d at 1323.

claim language is broader than the preferred embodiment, it is well-settled that claims are not to be confined to that embodiment."). Indeed, even if a patent only describes a single embodiment, it is well settled that the claims need not be limited to that embodiment alone. *Gemstar-TV Guide Int'l, Inc. v. Int'l Trade Comm'n*, 383 F.3d 1352, 1366 (Fed. Cir. 2004). But this case is even clearer: the patent describes multiple embodiments. The claim should not be limited to one embodiment.

B. The originally filed claims included both a DMA co-processor embodiment and a traditional DMA controller embodiment.

Appellants' MMP Portfolio includes file histories covering 37 applications resulting in seven issued U.S. patents. In addition, the Appellees in the present case and other parties have filed sixteen reexamination requests in the U.S. Patent and Trademark Office, and a nullity action in the European Patent Office, that has greatly multiplied the volume of the file histories for the MMP Portfolio. In total, the MMP Portfolio file histories (including reexamination proceedings) comprise approximately 291 U.S. patent references, 33 foreign patent references, 382 non-patent references, 134 litigation-related pleadings or transcripts, and 205 office actions and responses, leading to over 30,000 pages of correspondence between the applicants and PTO and over 1,000 references. The '890 patent, as well as the other patents in suit, arose from a single application filed on August 3, 1989, which

ultimately resulted in the MMP Portfolio.[10] That original application included 70 claims, disclosing a large number of independent and distinct inventions.

Claim 13 of the original application was specifically directed to the DMA co-processor invention:

> 13. A microprocessor system, comprising a central processing unit, a direct memory access processing unit, a memory, a bus connecting said central processing unit and said direct memory processing unit to said memory, said memory containing instructions for said central processing unit and said direct memory access processing unit, said **direct memory access processing unit including means** for fetching instructions for said central processing unit on said bus and **for fetching instructions for said direct memory processing unit** on said bus.

A2143. Much like the district court's construction of DMA CPU, originally filed claim 13 required the DMA to fetch and process its own instructions.

By contrast, originally filed claim 48 (which became claim 1 of the '890 patent and later claim 11 of the reexamined '890 patent, the claim at issue here) recited the use of a traditional DMA ***controller*** in a microprocessor architecture:

> A microprocessor, which comprises a main central processing unit **and a separate direct memory access central processing unit** in a single integrated circuit comprising said microprocessor, said main central processing unit having an arithmetic logic unit, a first push down stack with a top item register and a next item register, connected to provide inputs to said arithmetic logic unit, an output of said arithmetic logic unit being connected to said top item register, said top item register also being connected to provide inputs to an internal data bus, said internal data bus being bidirectionally connected to a loop counter, said loop counter being connected to a decrementer, said

[10] The original application, No. 07/389,334, eventually issued as U.S. Patent No. 5,440,749, one of the patents-in-suit.

> internal data bus being bidirectionally connected to a stack pointer, return stack pointer, mode register and instruction register, said internal data bus being connected to a memory controller, to a Y register of a return push down stack, an X register and a program counter, said Y register, X register and program counter providing outputs to an internal address bus, said internal address bus providing inputs to said memory controller and to an incrementer, said incrementer being connected to said internal data bus, said direct memory access central processing unit providing inputs to said memory controller, said memory controller having an address/data bus and a plurality of control lines for connection to a random access memory.

A2153-2154. Because originally filed claim 13 included the limitations of "including means for fetching instructions" for its DMA processing unit – while originally filed claim 48 did not – the latter's DMA processing unit should not be construed to include those limitations. *Curtiss-Wright Flow Control Corp. v. Velan, Inc.*, 438 F.3d 1374, 1381 (Fed. Cir. 2006) ("[C]laim differentiation takes on relevance in the context of a claim construction that would render additional, or different, language in another independent claim superfluous[.]"). Therefore, originally filed claim 48, which eventually became claim 11 here, encompasses both a DMA controller and a DMA co-processor.

C. Per the restriction requirement, the '890 patent is drawn to a microprocessor system that can have a traditional DMA controller or a DMA co-processor.

Because the original application contained so many different inventions, the examiner imposed a remarkable *ten-way* restriction requirement on August 31,

1992.[11] The restriction requirement divided the disclosed inventions into ten categories as follows:

> Group I, claims 1 and 2, "drawn to [a] microprocessor system having a multiplex bus, was filed as a divisional application on 07-Jun-95, US application number 08/480,462. That application was abandoned.
>
> Group II, claims 3, 6-11, 26-30 and 32-47, "drawn to a processor system having means for fetching multiple instructions in parallel during a single machine cycle" (the description of the group given by the examiner), and were patented as US '749.
>
> **Group III, claim 13, "drawn to a microprocessor system having a DMA for fetching instruction[s] for a CPU and itself," was filed as a divisional application on 07-Jun-95 as patent application number 08/480,015. That application was abandoned.**
>
> Group IV, claims 16-18 and 63-64, "drawn to a processing system configured to provide different memory access time[s] for different amounts of memory," was filed on 07-Jun-95 as US application number 08/485,031. That application issued as US patent number 5,604,915 on 18-Feb-97.
>
> Group V, claims 19-20 and 65-67, "drawn to [a] method and apparatus which operates as a variable clock speed," was filed on 07-Jun-95 as US application number 08/484,918. That application issued as US Pat number 5,809,336 on 15-Sep-98.
>
> Group VI, claims 22-23, "drawn to a CPU having stacks and pointers," was filed on 07-Jun-95 as US patent application number 08/484, 230. That application was abandoned.
>
> Group VII, claims 24-25 and 69-78, "drawn to a processing system for processing polynomial instruction[s]," was filed on 07-Jun-95 as US application number 08/484,720. That application was abandoned.

[11] *See* 35 U.S.C. § 121 ("If two or more independent and distinct inventions are claimed in one application, the Director may require the application to be restricted to one of the inventions.")

Group VIII, claims 48-57, "drawn to a microprocessor architecture," was filed on 07-Jun-95 as US patent application number 08/480, 206. That application issued as US patent number 5,530,890.

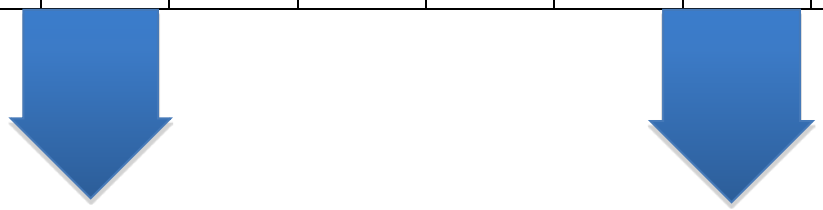
Group IX, claims 58-62, "drawn to method for prefetching," was filed on 07-Jun-95 as US patent application number 08/484,935. That application issued as US Pat number 5,784,584 on 21-Jul-98.

Group X, claim 68, "method for operating a stack," was filed on 07-Jun-95 as US application number 08/482,185. That application issued as US patent number 5,659,703 on 19-Aug-97.

A2172-2173 (August 31, 1992 Restriction Requirement) (emphasis added).

SUMMARY OF RESTRICTION REQUIREMENT

I	II	**III**	IV	V	VI	VII	**VIII**	IX	X
micro-processor system having a multiplex bus	fetching multiple instructions in parallel during a single machine cycle	micro-processor system having a DMA for fetching instruction[s] for a CPU and itself	different memory access time[s] for different amounts of memory	method and apparatus which operates as a variable clock speed	CPU having stacks and pointers	system for processing polynomial instruction[s]	micro-processor architecture	method for prefetching	for operating a stack



Abandoned ’890 patent

Originally filed claim 13, the DMA co-processor invention, was in Group III, constituting inventions "drawn to a microprocessor system having a DMA for fetching instruction[s] for a CPU and itself." A2172-2173. In other words, Group III was limited to a microprocessor system having a DMA co-processor of the type described in connection with the specification's first embodiment (*e.g.*, DMA CPU 72). *See* ’890 8:1; Fig. 5; 8:22-24 ("The DMA CPU 72 controls itself and has the

ability to fetch and execute instructions."). The Group III application, No. 08/480,015, was eventually abandoned.

Relevant here, the claims that eventually issued as the '890 patent were in Group VIII, and divided into application No. 08/480,206. Per the restriction requirement, those claims were "drawn to a microprocessor architecture." *Id*. The '890 patent issued on a first action allowance. Importantly, Group VIII was never limited to a DMA co-processor. Unlike Group III, Group VIII did not specify that that the DMA be capable of fetching or executing instructions for a CPU or itself. The broader scope of Group VIII was consistent with the specification, which describes two distinct microprocessor architectures:

> (1) microprocessor 50 described in connection with Figs 1-8, which uses a DMA co-processor DMA CPU 72 capable of fetching and executing its own instructions (*see* '890 8:1; Fig. 5; 8:22-24); and
>
> (2) microprocessor 310 described in connection with Figure 9, which uses "a more traditional DMA controller 314" also referred to as "DMA CPU 314," in order "[t]o keep chip size as small as possible" (*see* '890 12:61-66; Fig. 9; 10:52; 13:3-4).

There is nothing in the file history that limits the claims of Group VIII to one of these two architectures. Hence, Group VIII should cover ***both*** embodiments described in the specification, absent some limiting language in the file history. *See Phillips v. AWH Corp*., 415 F.3d 1303, 1323 (Fed. Cir. 2005) (*en banc*) ("[A]lthough the specification often describes very specific embodiments of the invention, we have repeatedly warned against confining the claims to those

embodiments."). Accordingly, the district court's construction of DMA CPU in a way that limits it to the first embodiment is improper. *Id*.

This prosecution history demonstrates that there were two separate DMA inventions in the original application: (1) a DMA co-processor that could fetch and execute its own instructions; and (2) a microprocessor architecture with traditional DMA controller. These two inventions were prosecuted separately, and invention (2) issued as the '890 patent. The objects of invention language relied upon by the district court for its construction was directed to the DMA co-processor claimed in originally filed claim 13, and not the microprocessor architecture invention claimed in the '890 patent.

D. The reexamination proceedings confirm that "DMA CPU" must include a traditional DMA controller.

The '890 patent was reexamined. Although claim 11, the claim under construction, was added during reexamination, it varies from pre-reexamination claim 1 only in that "said stack pointer pointing to said first push down stack" was added. Nothing about the claimed "DMA CPU" was changed.

The consistent meaning of the claim term "a separate direct memory access central procession unit" – which was used by the reexamination requester, the USPTO, and the applicant – included a traditional DMA controller.

For example, in the reexamination request filed by Fish & Richardson, the requestor argued:

> The **'890 patent teaches a direct memory access controller** and states that "conventional microprocessors provide direct memory access (DMA) for system peripheral units through DMA controllers, which may be located on the microprocessor integrated circuit" (*Id*., 1:52-55)

A2186 (*Ex Parte* Request for Reexamination, at 8) (emphasis added). The requestor further argued:

> . . . Requestor submits that the **DMA controllers were conventionally placed on the same chip as of the '890 patent's priority date** and thus this feature would have been considered obvious by one of ordinary skill in the art. For example, US patent number 4,783,764 to **Tsuchiya** et al. **describes** a **Direct Memory Access Controller** on a single integrated circuit with a CPU. . .

Id., at 11 (describing the "mode exchange circuit 9" shown in Tsuchiya, FIG. 3 as a DMA controller) (emphasis added).

The USPTO granted the reexamination request on April 8, 2009, in an order stating:

> . . . **Tsuchiya** describes a microprocessor further including a separate **direct memory access central processing unit** . . .

A2250 (Order Granting Ex Parte Reexamination, at 7) (emphasis added). In the first action on the merits, dated November 5, 2009, the examiner stated:

> . . . [T]he "Transputer Manual" . . . is seen to describe an **on-chip DMA controller**. . . .

A2262 (PTO Non-Final Office Action, at 4) (emphasis added). The Examiner went on to note:

> . . . [T]he references cited in the request for re-examination on page 11 (as well is pages 26 and 27) that teach of an **on-chip DMA controllers**. . . .

Id. (emphasis added).

These excerpts confirm that during reexamination, the patent owner and the USPTO considered the DMA controller of the ’890 patent to be a traditional DMA controller – not a DMA co-processor capable of fetching and executing its own instructions. At no time did the patent owner ever try to distinguish the prior art on the ground that the ’890 patent required a DMA co-processor, even though that would have been an obvious basis for distinction, if true. Accordingly, the district court erred in its construction, which limited the term “DMA CPU” to a DMA co-processor.

Conclusion and Statement of Relief Sought

For the forgoing reasons, this Court should reverse the district court’s January 21, 2014 and September 17, 2013 orders to the extent that they hold that the doctrine of intervening rights applies to the ’890 patent to preclude any claims of infringement before the date of the issuance of the reexamination certificate. In addition, this Court should reverse the district court’s construction of the term “separate DMA CPU” and construe it as “electrical circuit for reading and writing to memory that is separate from a main CPU.”

Respectfully Submitted,

May 1, 2014

/s/ James C. Otteson

James C. Otteson
Thomas T. Carmack
Philip W. Marsh
AGILITY IP LAW, LLP
149 Commonwealth Dr.
Menlo Park, CA 94025
Telephone: (650) 227-4800
Facsimile: (650) 318-3483
Email:
jim@agilityiplaw.com
tom@agilityiplaw.com
phil@agilityiplaw.com

Attorneys for Defendants-Appellants Technology Properties Limited, Patriot Scientific Corporation and Alliacense Limited

ADDENDUM

ADDENDUM

Page

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN JOSE DIVISION

HTC CORPORATION AND HTC AMERICA, INC., Plaintiffs, v. TECHNOLOGY PROPERTIES LIMITED, et al., Defendants.	Case No.: 5:08-cv-00882-PSG **ORDER RE: HTC'S MOTIONS FOR SUMMARY JUDGMENT OF NON-INFRINGEMENT AND NO WILLFULNESS** **(Re: Docket Nos. 457, 458)**

Before the court in this patent case are two motions for summary judgment brought by Plaintiffs HTC Corporation and HTC America, (collectively "HTC"). HTC first moves for "full" summary judgment of non-infringement and no willful infringement of U.S. Patent No. 5,809,336 ("the '336 patent"). HTC separately moves for partial summary judgment of non-infringement of the '336 patent and U.S. Patent No. 5,530,890 ("the '890 patent") and no willful infringement of the '890 patent. On August 13, 2013, the parties appeared for a hearing. Having considered the papers and arguments of counsel:

The court DENIES HTC's motion for summary judgment of "full" non-infringement of the '336 patent.

The court DENIES HTC's motion for partial summary judgment of non-infringement of the '336 patent.

The court DENIES HTC's motion for summary judgment of no willful infringement of the '336 patent.

The court GRANTS HTC's motion for partial summary judgment of non-infringement of the '890 patent.

The court GRANTS-IN-PART HTC's motion for partial summary judgment of no willful infringement of the '890 patent.

The court sets forth its reasoning below.

I. BACKGROUND

HTC Corporation is a Taiwan corporation with its principal place of business in Taoyuan, Taiwan, R.O.C. HTC's subsidiary, HTC America, is a Texas corporation with its principal place of business in Bellevue, Washington. Defendants Technology Properties Limited and Alliacense, Limited ("Alliacense") are California corporations with their principal place of business in Cupertino, California; Patriot Scientific Corporation ("Patriot") is a Delaware corporation with its principal place of business in Carlsbad, California. These defendants – Technology Properties Limited, Alliacense, and Patriot (collectively "TPL") – claim ownership of a family of related microprocessor patents. TPL refers to those patents as the Moore Microprocessor Portfolio patents ("MMP patents"), in recognition of co-inventor Charles Moore's contributions. HTC filed this suit on February 8, 2008, seeking a judicial declaration that four of the MMP patents – U.S. Patent Nos. 5,809,336 ("the '336 patent"), 5,784,584 ("the '584 patent"), 5,440,749 ("the '749 patent"), and 6,598,148 ("the '148 patent") – are invalid and/or not infringed.[1] TPL counterclaimed for

[1] *See* Docket No. 1.

infringement of the '336, '749, '148, and '890 patents on November 21, 2008.[2] On April 25, 2008, TPL filed two complaints in the Eastern District of Texas against HTC alleging infringement of the four patents at issue in the pending declaratory judgment action.[3] On June 4, 2008, TPL filed additional patent infringement actions against HTC in the Eastern District of Texas asserting U.S. Patent No. 5,530,890 ("the '890 patent").[4] On July 10, 2008, HTC amended its complaint before this court, adding claims for declaratory relief with respect to the '890 patent.[5] On February 23, 2009 the parallel Texas litigation was dismissed without prejudice following Judge Fogel's decision to deny TPL's Motion to Dismiss, or in the Alternative, to Transfer Venue in the California action.[6] On March 25, 2010, the court accepted the parties' stipulation to dismiss the '584 patent from this litigation.[7] On August 24, 2012, Technology Properties Limited, Patriot, and Phoenix Digital Solutions initiated an International Trade Commission ("ITC") investigation regarding HTC's alleged infringement of the '336 patent.[8] On July 17, 2013, the court accepted the parties' stipulation to dismiss the '148 and '749 patents from this litigation.[9]

The bottom line is that only the '336 and '890 patents remain at issue for the purposes of this litigation.

A. The '336 Patent

---

[2] *See* Docket No. 60 at 6-8.

[3] *See* Docket No. 16 at 3.

[4] *See* Docket No. 35 at 5.

[5] *See* Docket No. 34.

[6] See Docket Nos. 49 (denying motion to dismiss, to transfer venue, and to stay) and 88 (granting motion for leave to file motion for reconsideration and denying motion for reconsideration).

[7] *See* Docket No. 152.

[8] *See* Docket No. 561-1. Claims 1, 6, 7, 9-11, and 13-16 were asserted in the investigation. On September 6, 2013, Administrative Law Judge James Gildea issued an Initial Determination from in the ITC proceeding holding that HTC did not violate Section 337 of the Tariff Act of 1930. *See id.*

[9] *See* Docket No. 462.

The '336 patent issued on September 15, 1998 and describes a microprocessor with an internal variable speed clock, or oscillator, that drives the processor's central processing unit ("CPU"). Traditional microprocessors use external, fixed speed crystals to clock the CPU. A CPU's maximum possible processing capacity depends on process, voltage, and temperature ("PVT parameters"). An external clock must therefore set the timing of the CPU to suboptimal PVT conditions, resulting in waste of the CPU's processing speed under optimal conditions. The internal, variable clock described in the '336 patent claims real-time adjustment of the timing of the CPU by placing the clock on the chip itself. Thus, the CPU can perform optimally under any set of parameters. The microprocessor nevertheless requires a second external clock because devices other than the CPU do not operate at variable speed.

TPL claims that HTC's accused products infringe the '336 patent by their internal, variable speed oscillator on their microprocessors. At issue are claims 1, 6, 10, 11, 13, and 16.[10]

Claim 1 provides:

> A microprocessor system, comprising a single integrated circuit including a central processing unit and an entire ring oscillator variable speed system clock in said single integrated circuit and connected to said central processing unit for clocking said central processing unit, said central processing unit and said ring oscillator variable speed system clock each including a plurality of electronic devices correspondingly constructed of the same process technology with corresponding manufacturing variations, a processing frequency capability of said central processing unit and a speed of said ring oscillator variable speed system clock varying together due to said manufacturing variations and due to at least operating voltage and temperature of said single integrated circuit; an on-chip input/output interface connected to exchange coupling control signals, addresses and data with said central processing unit; and a second clock independent of said ring oscillator variable speed system clock connected to said input/output interface, wherein a clock signal of said second clock originates from a source other than said ring oscillator variable speed system clock.

Claim 6 provides:

> A microprocessor system comprising:

[10] Docket No. 494 at 7.

a central processing unit disposed upon an integrated circuit substrate, said central processing unit operating at a processing frequency and being constructed of a first plurality of electronic devices; an entire oscillator disposed upon said integrated circuit substrate and connected to said central processing unit, said oscillator clocking said central processing unit at a clock rate and being constructed of a second plurality of electronic devices, thus varying the processing frequency of said first plurality of electronic devices and the clock rate of said second plurality of electronic devices in the same way as a function of parameter variation in one or more fabrication or operational parameters associated with said integrated circuit substrate, thereby enabling said processing frequency to track said clock rate in response to said parameter variation; an on-chip input/output interface, connected between said central processing unit and an off-chip external memory bus, for facilitating exchanging coupling control signals, addresses and data with said central processing unit; and an off-chip external clock, independent of said oscillator, connected to said input/output interface wherein said off-chip external clock is operative at a frequency independent of a clock frequency of said oscillator and wherein a clock signal from said off-chip external clock originates from a source other than said oscillator.

Claim 10 provides:

In a microprocessor system including a central processing unit, a method for clocking said central processing unit comprising the steps of: providing said central processing unit upon an integrated circuit substrate, said central processing unit being constructed of a first plurality of transistors and being operative at a processing frequency; providing an entire variable speed clock disposed upon said integrated circuit substrate, said variable speed clock being constructed of a second plurality of transistors; clocking said central processing unit at a clock rate using said variable speed clock with said central processing unit being clocked by said variable speed clock at a variable frequency dependent upon variation in one or more fabrication or operational parameters associated with said integrated circuit substrate, said processing frequency and said clock rate varying in the same way relative to said variation in said one or more fabrication or operational parameters associated with said integrated circuit substrate; connecting an [on chip] on-chip input/output interface between said central processing unit and an off-chip external memory bus, and exchanging coupling control signals, addresses and data between said input/output interface and said central processing unit; and clocking said input/output interface using an off-chip external clock wherein said off-chip external clock is operative at a frequency independent of a clock frequency of said variable speed clock and wherein a clock signal from said off-chip external clock originates from a source other than said variable speed clock.

Claim 11 provides:

A microprocessor system, comprising a single integrated circuit including a central processing unit and an entire ring oscillator variable speed system clock in said single integrated circuit and connected to said central processing unit for clocking said central processing unit, said central processing unit and said ring oscillator

variable speed system clock each including a plurality of electronic devices correspondingly constructed of the same process technology with corresponding manufacturing variations, a processing frequency capability of said central processing unit and a speed of said ring oscillator variable speed system clock varying together due to said manufacturing variations and due to at least operating voltage and temperatureof said single integrated circuit; an on-chip input/output interface connected to exchange coupling control signals, addresses and data with said central processing unit; and a second clock independent of said ring oscillator variable speed system clock connected to said input/output interface, wherein said central processing unit operates asynchronously to said input/output interface.

Claim 13 provides:

A microprocessor system comprising: a central processing unit disposed upon an integrated circuit substrate, said central processing unit operating at a processing frequency and being constructed of a first plurality of electronic devices; an entire oscillator disposed upon said integrated circuit substrate and connected to said central processing unit, said oscillator clocking said central processing unit at a clock rate and being constructed of a second plurality of electronic devices, thus varying the processing frequency of said first plurality of electronic devices and the clock rate of said second plurality of electronic devices in the same way as a function of parameter variation in one or more fabrication or operational parameters associated with said integrated circuit substrate, thereby enabling said processing frequency to track said clock rate in response to said parameter variation; an on-chip input/output interface, connected between said central processing unit and an off-chip external memory bus, for facilitating exchanging coupling control signals, addresses and data with said central processing unit; and an off-chip external clock, independent of said oscillator, connected to said input/output interface wherein said off-chip external clock is operative at a frequency independent of a clock frequency of said oscillator and further wherein said central processing unit operates asynchronously to said input/output interface.

Claim 16 provides:

In a microprocessor system including a central processing unit, a method for locking said central processing unit comprising the steps of providing said central processing unit upon an integrated circuit substrate, said central processing unit being constructed of a first plurality of transistors and being operative at a processing frequency; providing an entire variable speed clock disposed upon said integrated circuit substrate, said variable speed clock being constructed of a second plurality of transistors; clocking said central processing unit at a clock rate using said variable speed clock with said central processing unit being clocked by said variable speed clock at a variable frequency dependent upon variation in one or more fabrication or operational parameters associated with said integrated circuit substrate, said processing frequency and said clock rate varying in the same way relative to said variation in said one or more fabrication or operational parameters associated with said integrated circuit substrate; connecting an on-chip input/output interface between said central processing unit and an off-chip external memory bus,

and exchanging coupling control signals, addresses and data between said input/output interface and said central processing unit; and clocking said input/output interface using an off-chip external clock wherein said off-chip external clock is operative at a frequency independent of a clock frequency of said variable speed clock, wherein said central processing unit operates asynchronously to said input/output interface.

B. The '890 Patent

The '890 patent first issued on June 25, 1996 and originally included ten claims, nine of which depended from the sole independent claim, claim 1.[11] On January 19, 2009, the '890 patent was subjected to ex parte reexamination.[12] An amended version of the patent emerged on March 1, 2011.[13] The reexamination proceeding resulted in the cancellation of claims 1-4, confirmation of the patentability of claims 5-10, and addition of claims 11-20. At issue in this suit are claims 11, 12, 13, 17, and 19.[14]

Claim 11, the amended independent claim on which all of the other claims depend, describes:

> A microprocessor, which comprises a main central processing unit and a separate direct memory access central processing unit in a single integrated circuit comprising said microprocessor, said main central processing unit having an arithmetic logic unit, a first push down stack with a top item register and a next item register, connected to provide inputs to said arithmetic logic unit, an output of said arithmetic logic unit being connected to said top item register, said top item register also being connected to provide inputs to an internal data bus, said internal data bus being bidirectionally connected to a loop counter, said loop counter being connected to a decrementer, said internal data bus being bidirectionally connected to a stack pointer, return stack pointer, mode register and instruction register, said stack pointer pointing into said first push down stack, said internal data bus being connected to a memory controller, to a Y register of a return push down stack, an X register and a program counter, said Y register, X register and program counter providing outputs to an internal address bus, said internal address bus providing inputs to said memory controller and to an incrementer, said incrementer being connected to said internal data bus, said direct memory access central processing

[11] *See* Docket No. 458 at 2.

[12] *See id.*

[13] *See id.*

[14] *See id.*

unit providing inputs to said memory controller, said memory controller having an address/data bus and a plurality of control lines for connection to a random access memory.

During reexamination, the patentee added the phrase "said stack pointer pointing into said first push down stack," which did not appear in claim 1.

II. SUMMARY JUDGMENT STANDARDS

Summary judgment is appropriate only if there is "no genuine dispute as to any material fact and the movant is entitled to judgment as a matter of law."[15] The moving party bears the initial burden of production by identifying those portions of the pleadings, discovery, and affidavits which demonstrate the absence of a triable issue of material fact.[16] The standard for summary judgment differs depending on whether the moving party bears the burden of persuasion at trial.[17] If the moving party bears the burden of persuasion at trial, that party must present "credible evidence" showing that he is entitled to a directed verdict.[18] The burden of production then shifts to the non-moving party to produce evidence raising a genuine issue of material fact.[19] On the other hand, if the moving party does not bear the burden of persuasion at trial, he can prevail on a motion for summary judgment in two ways: by proffering "affirmative evidence negating an element of the non-moving party's claim," or by showing the non-moving party has insufficient evidence to establish an "essential element of the non-moving party's claim."[20] If met by the moving party, the burden of production then shifts to the non-moving party, who must then provide

[15] Fed. R. Civ. P. 56(a).

[16] *See* Fed. R. Civ. P. 56(c)(1); *Celotex Corp. v. Catrett*, 477 U.S. 317, 323 (1986).

[17] *See Celotex Corp.*, 477 U.S. at 331.

[18] *Id.*

[19] *See id.*

[20] *Id.*

specific facts showing a genuine issue of material fact for trial.[21] In both instances, the ultimate burden of persuasion remains on the moving party.[22] In reviewing the record, the court must construe the evidence and the inferences to be drawn from the underlying evidence in the light most favorable to the non-moving party.[23]

III. DISCUSSION

A. HTC's Motion for Summary Judgment of Non-Infringement and No Willful Infringement of the '336 Patent

1. Non-Infringement of the '336 Patent

The court first considers HTC's motion for summary judgment of "full" non-infringement of the '336 patent. HTC argues that summary judgment is warranted because when the independent claims of the '336 patent are properly construed, HTC's products do not perform the claimed invention. HTC specifically points to three terms that each appear in two claims: (1) "entire ring oscillator variable speed system clock" (claims 1 and 11), (2) "entire oscillator" (claims 6 and 13), and (3) "an entire variable speed system clock" (claims 10 and 16).

HTC argues as follows. The prosecution history of the '336 patent demonstrates the applicants' repeated and express disclaimer that the claimed timing element – the oscillator or variable speed clock – had any connection to or dependence on a reference signal from an external crystal or other fixed timing piece. To further distinguish the '336 patent, the applicants added the "entire" term to explicitly claim only a timing element that wholly and exclusively appeared with the CPU on the chip. HTC's processors, in contrast, rely on an external crystal timing piece (called

[21] *See id.* at 330; *T.W. Elec. Service, Inc. v. Pac. Elec. Contractors Ass'n*, 809 F.2d 630, 630 (9th Cir. 1987).

[22] *See id.*

[23] *See Anderson v. Liberty Lobby, Inc.*, 477 U.S. 242, 248 (1986); *Matsushita Elec. Indus. Co. v. Zenith Radio Corp.*, 475 U.S. 574, 587 (1986).

a phase-locked loop or "PLL"). Unlike the invention, therefore, the timing elements of HTC's processors do not sit entirely on the chip and do not vary with PVT parameters.

TPL responds that HTC improperly seeks reconsideration of this court's previous claim construction. The court properly construed the "entire variable speed system clock" term and this construction should extend to the other three "entire" terms. HTC's additional limitations are not supported by the specification, which does not speak to whether the oscillator or variable speed system clock also could work with an external crystal. As for any disclaimer, the applicants never disclaimed all reliance or reference to an off-chip crystal. Instead, the disclaimer to avoid the Magar reference was to an off-chip oscillator that generated the on-chip clock. As to the Sheets reference, the applicants distinguished their clock reference by pointing out that it was not an on-chip oscillator but rather an off-chip clock, and that off-chip clock required a command input to change its frequency. The oscillator taught by the '336 patent, in contrast, is self-generating on the chip itself and does not require an outside command to change frequency. As to the variation argument, even by HTC's own admission, the on-chip HTC oscillators vary and the PLLs in fact serve to limit that variation. That the net result may be a minimal change in the frequency of the clock is not enough to take HTC's accused products beyond the claim language.

HTC replies that the on-chip oscillator does not "generate" the CPU clock unless it communicates with the PLL, making the PLL necessary to "generate" the clock – and thereby outside of the claim language (as construed in light of the disclaimers). HTC further replies that frequency control in fact is generation of the clock because the oscillator does not begin to run independently. The PLL controls the oscillator and sets the frequency, which generates the clock. As to the variation issue, HTC argues that a person of ordinary skill in the art would understand the de minimis variation experienced by its products as rendering the timing element essentially fixed.

The court agrees with HTC that the disputed limitations are properly understood to exclude any external clock used to generate a signal.[24] Nevertheless, there remains a factual dispute whether HTC's products contain an on-chip ring oscillator that is self-generating and does not rely on an input control to determine its frequency. While HTC's expert says that the PLLs generate the clock, TPL's expert counters that the ring oscillators generate the clock and the PLLs merely buffer or fix the frequency.[25] This is a classic factual question that requires a trial to answer.

2. Willful Infringement of the '336 Patent

To "establish willful infringement, a patentee must show by clear and convincing evidence that the infringer acted despite an objectively high likelihood that its actions constituted infringement of a valid patent."[26] A patentee therefore must establish two elements. First, the patentee must show the accused infringer acted with "objective recklessness." Objective recklessness remains a question of law "predicated on underlying mixed questions of law and fact."[27] The objective recklessness prong "entails an objective assessment of potential defenses based on the risk presented" by the patent which "may include questions of infringement but also can be expected in almost every case to entail questions of validity that are not necessarily

[24] The patentee's arguments traversing the prior art narrowed the claims. *See Festo Corp. v. Shoketsu Kinzoku Kogyo Kabushiki Co.*, 535 U.S. 722, 740 (2002) ("A patentee's decision to narrow his claims through amendment may be presumed to be a general disclaimer of the territory between the original claim and the amended claim."); *cf. Saeilo Inc. v. Colt's Mfg. Co.*, 26 F. App'x 966, 973 (Fed. Cir. 2002) ("Where an amendment narrows the scope of a claim for a reason related to the statutory requirements for patentability, prosecution history estoppel acts as a complete bar to the application of the doctrine of equivalents to the amended claim element.").

[25] *Compare* Docket No. 457 at 16 ("the oscillators in the accused products indisputably rely on an external crystal or clock generator to clock" the CPU), *with* Docket No. 470 at 14 ("Each HTC product includes a CPU/system clock – a ***ring oscillator*** within a PLL – that ***generates*** a clock signal ***on its own***, as long as it has a power supply.") (emphasis in original).

[26] *In re Seagate Tech., LLC*, 497 F.3d 1360, 1371 (Fed. Cir. 2007) (*en banc*).

[27] *See Bard Peripheral Vascular, Inc. v. W.L. Gore & Assocs., Inc.*, 682 F.3d 1003, 1006-07 (Fed. Cir. 2012) (holding that the objective determination of recklessness, even though predicated on underlying mixed questions of law and fact, is decided by the judge as a question of law subject to de novo review).

dependent on the factual circumstances of the particular party accused of infringement."[28] Second, if the requisite threshold objective recklessness is established, then the patentee must show that the "objectively-defined risk" of infringement determined by the record developed in the infringement proceeding "was either known or so obvious that it should have been known to the accused infringer."[29]

HTC argues that TPL has not presented sufficient evidence to make a prima facie case of willful infringement, in view of its "clear, legitimate, and objectively reasonable defenses" to HTC's claims of infringement.[30] In particular, its proposed constructions have been adopted by other tribunals and the ITC in particular. HTC's non-infringement position at the ITC was "sufficiently compelling and reasonable" that both the ITC staff attorney and Judge Gildea himself agreed with HTC's position.[31]

TPL takes issue with HTC's reference in this case to the ITC litigation. Different theories of infringement and different products are implicated by the two cases. Different claim constructions have issued in the cases. The staff attorney's position and Judge Gildea's conclusions are therefore irrelevant. Separately, TPL's successful licensing of the MMP patent portfolio suggests that HTC could not reasonably or realistically expect its invalidity or

[28] *Id.* at 1006.

[29] *Seagate*, 497 F.3d at 1371.

[30] Looking to Fed. R. Civ. P. 37(c)(1) HTC further points out that TPL failed to substantively respond to its interrogatory about willful infringement. *See* Fed. R. Civ. P. 37(c)(1) ("If a party fails to provide information or identify a witness as required by Rule 26(a) or (e), the party is not allowed to use that information or witness to supply evidence on a motion, at a hearing, or at a trial, unless the failure was substantially justified or is harmless."). But TPL's response raising a host of objections appears substantially justified, even if it is not ultimately persuasive, and in any event HTC does not appear to have taken any steps whatsoever in the intervening four years to compel a more complete response.

[31] Judge Gildea's Initial Determination ("ID") did not issue until September 6, 2013, after the papers for this motion were filed.

non-infringement defenses to succeed in this litigation. Finally, direct pre-suit communication between HTC and TPL establishes that HTC had notice of its allegedly infringing activities.

District courts appear split as to whether current evidence that a party's actions were objectively reasonable is relevant to a willfulness analysis under *Seagate*. In *i4i Ltd. P'ship v. Microsoft Corp.*, Judge Davis held that the correct willfulness analysis "focuses on whether, given the facts and circumstances prior to [the accused infringer's] infringing actions, a reasonable person would have appreciated a high likelihood that acting would infringe a valid patent."[32] The "number of creative defenses that Microsoft is able to muster in an infringement action after years of litigation and substantial discovery is irrelevant to the objective prong of the *Seagate* analysis."[33] Judge Davis then explained that the court should more properly focus on whether defenses would have been objectively reasonable and apparent before Microsoft infringed and was sued.[34] In *Uniloc USA, Inc. v. Microsoft Corp.*, Judge Smith was "not convinced that such a 'before and after' line is so easily drawn, or for that matter appropriate, to measure the objective likelihood (or lack thereof) that a party acted to infringe a valid patent."[35] Judge Smith emphasized that "the inquiry is case-specific" and should focus on an objective view of the record.[36]

The court agrees with HTC that favorable court rulings can support the objective reasonableness of its non-infringement positions. The court cannot help but take note of the analogous issue of the "book of wisdom" when addressing patent damages. The Supreme Court has affirmed that after-arising "[e]xperience . . . is a book of wisdom that courts may not

---

[32] 670 F. Supp. 2d 568, 582 (E.D. Tex. 2009).

[33] *Id.*

[34] *See id.*

[35] 640 F. Supp. 2d 150, 177 n. 33 (D.R.I. 2009).

[36] *Id.*

neglect."[37] Nonetheless, "as the party moving for summary judgment" HTC "must do more than persuade [the court] that its defenses were reasonable."[38] Instead, HTC "must establish that 'there is no genuine dispute as to any material fact' and that [the accused infringer] 'is entitled to judgment as a matter of law'—in other words, that *no reasonable fact-finder* could find willful infringement."[39]

Viewing the evidence in the light most favorable to TPL, the court concludes that a reasonable fact finder could plausibly find facts sufficient to support a conclusion of willful infringement. TPL's burden to show willful infringement by clear and convincing evidence is a steep one. But where factfinding is necessary, trial courts generally reserve willfulness until after a full presentation of the evidence on the record to the jury.[40] The record supports a finding that HTC knew about the patents and TPL's claims of infringement before it began the activities that allegedly infringe and as explained above, here there remains an important issue regarding the role of the external crystal in HTC's products in generating a signal.[41] Under these circumstances summary judgment on the issue of willfulness is not warranted.

B. Partial Summary Judgment of Non-Infringement of the '336 Patent and the '890 Patent and No Willful Infringement of the '890 Patent

HTC next moves for partial summary judgment of non-infringement of the '336 patent and the '890 patent based on the doctrine of absolute intervening rights. By this same motion, HTC also seeks summary judgment of no willful infringement under the '890 patent.

---

[37] *Sinclair Ref. Co. v. Jenkins Petroleum Process Co.*, 289 U.S. 689, 690 (1933).

[38] *Kimberly-Clark Worldwide, Inc. v. First Quality Baby Products, LLC*, Case No. 1:09-cv-1685, 2013 WL 1465403, at *2 (M.D. Pa. Apr. 11, 2013)

[39] *Id.* (citing Fed. R. Civ. P. 56(a)).

[40] *See, e.g. Bard*, 682 F.3d at 1008; *Fujitsu Ltd. v. Belkin Int'l, Inc.*, Case No. 10-cv-03972-LHK, 2012 WL 4497966, at *39 (N.D. Cal. Sept. 28, 2012).

[41] *See* Docket No. 470-1, Ex. A (Nov. 7, 2006 correspondence from Alliacense to HTC); Docket No. 470-1, Ex. B (Nov. 20, 2006 correspondence from Alliacense to HTC).

Under 35 U.S.C § 307(b), a patent owner may not recover for infringement of claims that are invalidated or amended through the reexamination process.[42] The "reexamination statute restricts a patentee's ability to enforce the patent's original claims to those claims that survive reexamination in 'identical' form."[43] "'Identical' does not mean verbatim, but means at most without substantive change."[44] The court must therefore determine whether the scope of the claims are the same, not just whether the same words are used.[45] Section 307 shields "those who deem an adversely held patent to be invalid; if the patentee later cures the infirmity by reissue or reexamination, the making of substantive changes in the claims is treated as an irrebuttable presumption that the original claims were materially flawed."[46] The "statute relieves those who may have infringed the original claims from liability during the period before the claims are validated."[47]

Whether "amendments made to overcome rejections based on prior art are substantive depends on the nature and scope of the amendments, with due consideration to the facts in any given case that justice will be done."[48] "An amendment that clarifies the text of the claim or makes it more definite without affecting its scope is generally viewed as identical."[49] To make its determination under the so-called doctrine of intervening rights, the court must consider "the scope of the original and reexamined claims in light of the specification, with attention to the references

---

[42] *See Fresenius USA, Inc. v. Baxter Intern., Inc.*, 721 F.3d 1330, 1339 (Fed. Cir. 2013).

[43] *Id.* (listing cases).

[44] *Id.*

[45] *See id.*

[46] *Bloom Eng'g Co. v. N. Am. Mfg. Co.*, 129 F.3d 1247, 1249 (Fed. Cir. 1997).

[47] *Id.*

[48] *Id.*

[49] *Id.*

that occasioned the reexamination, as well as the prosecution history and any other relevant information."[50]

1. Non-Infringement of the '336 Patent

As noted earlier the '336 patent issued September 15, 1998, and included ten originally-issued claims.[51] A series of ex parte reexamination requests were filed against the '336 patent between October 2006 and January 2007.[52] When the reexamination proceedings completed, claims 1, 6, and 10 emerged with modified language, and new independent claims 11, 13, and 16 were added. TPL amended claim 1 to further describe the "second clock independent of said ring oscillator" to say that "wherein a clock signal of said clock originates from a source other than said ring oscillator variable speed system clock." Claim 6 was amended to describe the "off-chip external clock" to likewise derive its "clock signal" "from a source other than said oscillator." Claim 10 includes a similar amendment that adds that the "off-chip external clock" has a "clock signal" that "originates form a source other than said variable speed clock." Claims 6 and 10 also added "off-chip" references to the descriptions of the second clocks. Claims 11, 13, and 16 were based on independent claims 1, 6, and 10, but during reexamination TPL added an additional clause to the end of each claim: "wherein said central processing unit operates asynchronously to said input/output interface."

In HTC's view, it should not be held liable for infringement of the '336 patent claims 1, 6, 10, 11, 13, and 16 because those claims were either substantially narrowed or newly-added through reexamination. Any recovery for the '336 patent should be limited to the date of the issuance of the reexamination certificate on December 15, 2009, because the amendments were sufficiently substantive to preclude recovery from before the amendments.

---

[50] *Id.*

[51] *See* Docket No. 458 at 5.

[52] *Id.*

TPL responds that these amendments serve as nothing more than clarification of the claim language and that the scope of the claims have not changed. Several excerpts from the prosecution history of the reexamination demonstrate that the patentee believed the amended claim language only clarified how the second clock was "independent"[53] and that the "external" components were in fact "off-chip"[54].

HTC replies that the original claims differ from the amended claims in scope because the original claims spoke only to the difference in frequency control – and that is what "independence" really references in these claim terms. Because a clock with signal origins from the ring oscillator but with an independent frequency could exist under the original claims but not under the amended claims, the claim is narrower and therefore substantively different. For claims 11, 13, and 16, the "independent" clock signals could have a "readily predictable phase relationship." Because of that possibility, the claims are narrower and thereby substantively different. Further, the court should not credit self-serving testimony from the prosecution history.[55]

On balance, the court finds that the amended claim language added during reexamination did not substantively amend the asserted '336 claims' scope. "Independent" in the disputed claims must be understood to be just that: without dependence of any kind. While HTC offers a more nuanced interpretation that focuses exclusively on frequency control, it cites no intrinsic – or for that matter extrinsic evidence – to support its position. Coupled with the references in the prosecution history indicating that the amendments really were for clarification purposes only, TPL's argument is more persuasive.

---

[53] *See* Docket No. 471-5, Ex. E at 2; Docket No. 471-6, Ex. F at 11, 27; Docket No. 471-7, Ex. G at 8-12, 14.

[54] *See* Docket No. 471-7, Ex. G at 12, 16.

[55] *See Moleculon Research Crop. v. CBS, Inc.*, 793 F.2d 1261, 1270 (Fed. Cir. 1986) (holding that documents submitted by the patentee during prosecution may be considered for claim interpretation purposes, but "might very well contain merely self-serving statements which likely would be accorded no more weight than testimony of an interested witness or argument of counsel. Issues of evidentiary weight are resolved on the circumstances of each case.").

2. Motion for Partial Summary Judgment of Non-Infringement and No Willful Infringement of the '890 Patent

a. Non-Infringement of the '890 Patent

The court next considers HTC's motion for summary judgment of non-infringement of the '890 patent claims 11, 12, 13, 17, and 19. As noted above, claims 12, 13, 17, and 19 all depend on independent claim 11.

HTC again argues the doctrine of absolute intervening rights entitles it to summary judgment of non-infringement. During reexamination, TPL added claim language further defining a stack pointer as "pointing into said first push down stack," after the examiner identified no function for the stack pointer in the original claim language. The examiner noted that the amendment to claim 1 prevented the claim from being anticipated by the prior art under 35 U.S.C. § 102. This change to the '890 patent during reexamination was substantive and that the absolute intervening rights doctrine bars liability arising before the reexamination terminated.

TPL initially responds that HTC's assertion of the absolute intervening rights doctrine is untimely because it did not include the affirmative defense in its answer to TPL's complaint.[56] As to the merits, TPL says that the amendment only clarified the claim scope but did not substantively amend the claim, precluding the absolute intervening rights doctrine. Further, in *Norwood v. Vance* the Ninth Circuit noted that parties may raise affirmative defenses for the first time at summary judgment only if the opposing party is not prejudiced.[57] Allowing HTC to assert the defense – four years into this litigation – would subject it to unfair prejudice.

The court is not persuaded that TPL has established the prejudice necessary to bar HTC's assertion of the absolute intervening rights doctrine at this stage in the litigation. TPL does not, for

[56] The initial declaratory judgment complaint in this case was filed February 8, 2008. *See supra* note 1. The '890 patent did not reissue following reexamination until March 1, 2011. *See supra* note 13.

[57] 591 F.3d 1062, 1075 (9th Cir. 2010).

example, articulate the discovery it might have otherwise taken had HTC promptly moved to amend its answer in 2011.

Turning to the merits, HTC asserts estoppel and argues claim 11 emerged from reexamination substantively different from former claim 1. During reexamination, the examiner found claim 1 invalid. In an August 12, 2010, advisory action the examiner noted that claim 1 failed to provide a function for the "stack pointer" and the claim language only identified the stack pointer as "bidirectionally connected to an internal bus," – an error claim 11 corrected. The examiner also observed that the additional language in claim 11 avoided the May reference, U.S. Patent No. 4,758,948 ("the '948 patent"), that teaches using a push down stack but not expressly a stack pointer performing the function that the amended language defines. Therefore, that the absolute intervening rights doctrine bars infringement liability prior to the issuance of the reexamination certificate.

TPL sees it differently. The change to claim 11 only makes the claim more definite. The examiner's primary concern with claim 1 centered on the discussion in the May patent of an instruction pointer. The instruction pointer identifies the instructions of a process and under the broadest interpretation the stack pointer likewise could be construed to read onto the prior art. No person of ordinary skill in the art would understand a stack pointer could not perform equivalently to an instruction pointer. As described in claim 1, the stack pointer would be understood by a person of ordinary skill in the art to point to only to the first push down stack referenced in claim 1 – and so the additional language only explicitly states what a person of ordinary skill in the art already would understand claim 1 to teach.

HTC replies that TPL's arguments rely on extrinsic evidence and that the intrinsic evidence reveals that absent the added limitation, the stack pointer was impermissibly vague and the amendment substantively narrowed the claim.

The court agrees with HTC. As the examiner's office actions indicated, in the original claim language the stack pointer did nothing except connect to the internal data bus, but TPL's argument that a person of ordinary skill in the art necessarily would color in the ambiguity with an understanding that the stack pointer points only to the first push down stack is not persuasive. As HTC points out, claim 1 (and claim 11) employs the term "comprising," which reveals that the claim is "inclusive or open-ended and does not exclude additional, unrecited elements or method steps."[58] Given that the specification in fact references a second push down stack, the second stack must be presumed to be distinct from the return stack identified in the claim language, other push down stacks potentially could be used and still fall within claim 1. Thus, where the stack pointer points matters. If multiple push down stacks were included in a processor, it is unclear under the language of claim 1 whether the stack pointer points to one of the stacks, all of the stacks, or some multiple in between.

At bottom, the court finds the added language limits the stack pointer to the first push down stack and substantively changes the scope of the claim. Because the added claim language narrows the scope of the claims, any claims of infringement before the date of the issuance of the reexamination certificate must be precluded.

b. Willful Infringement of the '890 Patent

The court finally addresses the issue of willful infringement related to the '890 patent.

HTC asserts that under the objective recklessness prong, the reexamination and amendment of the '890 patent supports HTC's position that it was not objectively reckless. HTC points out that TPL has offered no evidence that it even knew of the '890 patent before the suit. HTC also argues that the failure by TPL to pursue a preliminary injunction suggests that willful infringement is not at issue.

[58] *CollegeNet, Inc. v. ApplyYourself, Inc.*, 418 F.3d 1225, 1235 (Fed. Cir. 2005).

TPL responds that it provided notice to HTC of the patents and of its infringing behavior in 2006. The reexamination process actually cuts against HTC because most of the substance of the patents in fact survived intact with a "second stamp of validity from the PTO."[59] The PTO accepts 92% of reexamination applications, so the PTO's grant of patent reexamination is not enough to undercut willful infringement.[60] A "substantial question of patentability raised by a reexamination request is not dispositive" in a willfulness inquiry.[61]

Although the record at least suggests that HTC was made aware of the patents-in-suit as early as November 2006,[62] as discussed above the reexamined '890 patent bars claims of infringement before the date of the issuance of the certificate because the additional language added to independent claim 11 narrowed the scope of the claim.[63] It follows that because HTC cannot be held liable for infringement before March 1, 2011, willful infringement for this period is precluded.

The court next turns to whether HTC can be found to have willfully infringed the '890 patent following reexamination. Generally, a "patentee who does not attempt to stop an accused infringer's activities [by moving for a preliminary injunction] should not be allowed to accrue

[59] Docket No. 469 at 17.

[60] *See id.* n.11.

[61] *Plumley v. Mockett*, 836 F. Supp. 2d 1053, 1075 (C.D. Cal. 2010); *see also See Lucent Techs., Inc. v. Gateway, Inc.*, Case No. 07–cv–2000–H, 2007 WL 6955272, at *7 (S.D. Cal. Oct. 30, 2007) ("The Court does not assume that a reexamination order will always prevent a plaintiff from meeting their burden on summary judgment regarding willful infringement, but it does consider this as one factor among the totality of the circumstances.").

[62] *See* Docket No. 469-12, Ex. C (correspondence from Alliacense notifying HTC that HTC was infringing the patents contained in the MMP Portfolio, including the '890 patent).

[63] Moreover, at least one district court has noted, albeit in dicta, that "a patentee's willful infringement claim fails as a matter of law where the PTO requires amendments to the patent before issuing a reexamination certificate." *Plumley*, 836 F. Supp. 2d at 1075 (explaining court's opinion in *TGIP, Inc. v. AT & T Corp.*, 527 F. Supp. 2d 561 (E.D. Tex. 2007)).

enhanced damages based solely on the infringer's post-filing conduct."[64] But as TPL happily highlights, HTC conceded in prior litigation "that *Seagate* did not create a *per se* bar to claims for post-filing willful infringement where an injunction was not sought."[65] "Because *Seagate* did not create a *per se* bar, the determination of whether a patentee may pursue a claim for willful infringement based on post-filing conduct without seeking a preliminary injunction 'will depend on the facts of each case.'"[66] Patentees who neither practice the invention nor directly compete with the accused infringer are "excused from *Seagate's* rule that a patentee must seek an injunction to sustain a claim for post-filing willful infringement."[67] There may be circumstances "where an infringer's post-filing conduct was found to be willful" where "some material change that could create an objectively high likelihood of infringing a valid patent, such as a patent surviving a reexamination proceeding without narrowed claims."[68]

Viewing the evidence in the light most favorable to TPL and drawing all reasonable inferences in its favor, especially TPL's successful licensing program related to the patents-in-suit, the court concludes that a reasonable fact finder could plausibly find facts supporting a conclusion of willful infringement following the reexamination of the '890 patent.

---

[64] *Seagate*, 497 F.3d at 1372; *see also Anascape, Ltd. v. Microsoft Corp.*, Case No. 9:06-cv-158, 2008 WL 7182476 (E.D. Tex. Apr. 25, 2008) (patentee who did not move for preliminary injunction was not entitled to benefit from its lack of diligence by obtaining enhanced damages for willfulness during the post-filing period).

[65] *DataQuill Ltd. v. High Tech Computer Corp.*, 887 F. Supp. 2d 999, 1015 (S.D. Cal. 2011).

[66] *Id.* (citing *Seagate* 497 F.3d at 1374).

[67] *Id.*

[68] *LML Holdings, Inc. v. Pac. Coast Distrib. Inc.*, Case No. 11-cv-06173-YGR, 2012 WL 1965878 (N.D. Cal. May 30, 2012) (citing *St. Clair Intellectual Prop. Consultants, Inc. v. Palm, Inc.*, Case No. 04–1436–JJF–LPS, 2009 WL 1649751, at *1 (D. Del. Jun.10, 2009)); *see also Webmap Technologies, LLC v. Google, Inc.*, Case No. 2:09–cv–343–DF–CE, 2010 WL 3768097, at *2-3 (E.D. Tex. Sep. 10, 2010).

IT IS SO ORDERED.

Dated: September 17, 2013

PAUL S. GREWAL
United States Magistrate Judge

IN THE UNITED STATES DISTRICT COURT

FOR THE NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

Acer, Inc.,

Plaintiff,

v.

Technology Properties Ltd, et al.,

Defendants.
_____/

HTC Corp.,

Plaintiff,

v.

Technology Properties Ltd, et al.,

Defendants.
_____/

Barco NV,

Plaintiff,

v.

Technology Properties Ltd, et al.,

Defendants.
_____/

NO. C 08-00877 JW
NO. C 08-00882 JW
NO. C 08-05398 JW

FIRST CLAIM CONSTRUCTION ORDER

I. INTRODUCTION

Technology Properties Limited, Patriot Scientific Corporation and Alliacense, Ltd. (collectively, "Defendants") own a group of five patents known as the Moore Microprocessor

Portfolio patents.[1] Plaintiffs Acer, Inc.,[2] HTC Corp.[3] and Barco, N.V.[4] each filed lawsuits seeking a judicial declaration that the Patents-in-Suit are either invalid or are not infringed. Defendants filed counterclaims for infringement of the Patents-in-Suit. In due course, the actions were related and consolidated.[5]

On January 27, 2012, the Court conducted a hearing in accordance with Markman v. Westview Instruments, Inc.,[6] to construe language of the asserted claims over which there is a dispute. At the hearing, in addition to the normal intrinsic evidence, the parties relied upon a prior

---

[1] The five Patents-in-Suit are U.S. Patent Nos. 5,809,336 ("the '336 Patent"), 5,784,584 ("the '584 Patent"), 5,440,749 ("the '749 Patent"), 6,598,148 ("the '148 Patent") and 5,530,890 ("the '890 Patent").

[2] The first of these now-consolidated actions was filed on February 8, 2008. Acer filed suit against Defendants seeking a judicial declaration that the '336 Patent, the '584 Patent and the '749 Patent are invalid or are not infringed by Acer. (See Docket Item No. 1 in No. C 08-00877 JW.) On November 21, 2008, Defendants counterclaimed for infringement of the '336 Patent and the '749 Patent. (See Docket Item No. 60 in No. C 08-00877 JW.) On February 9, 2009, Acer amended its complaint to add claims pertaining to the '148 Patent and the '890 Patent. (See Docket Item No. 98 in No. C 08-00877 JW.) On February 24, 2009, Defendants counterclaimed with respect to those two patents. (See Docket Item No. 99 in No. C 08-00877 JW.)

[3] On February 8, 2008, HTC also filed suit seeking a judicial declaration that the '336 Patent, the '584 Patent, the '749 Patent and the '148 Patent are invalid or are not infringed by HTC. (See Docket Item No. 1 in No. C 08-00882 JW.) On July 10, 2008, HTC amended its complaint to add claims pertaining to the '890 Patent. (See Docket Item No. 34 in No. C 08-00882 JW.) On November 21, 2008, Defendants counterclaimed with respect to each of those patents except for the '584 Patent. (See Docket Item No. 60 in No. C 08-00882 JW.)

[4] On December 1, 2008, Barco filed suit seeking a judicial declaration that the '584 Patent, the '749 Patent and the '890 Patent are invalid or are not infringed by Barco. (See Docket Item No. 1 in No. C 08-05398 JW.) On February 17, 2009, Defendants counterclaimed for infringement with respect to the '749 Patent, the '890 Patent and the '336 Patent. (See Docket Item No. 27 in No. C 08-05398 JW.)

[5] Judge Fogel ordered the cases related. (See Docket Item No. 21 in No. C 08-00882 JW; Docket Item No. 21 in No. C 08-05398 JW.) On September 1, 2011, this matter was reassigned from Judge Fogel to Chief Judge Ware. (See Docket Item No. 291 in No. C 08-00877 JW.)

[6] 517 U.S. 370 (1996).

claim construction order by Judge T. John Ward[7] and documentary material from reexamination proceedings.[8]

This Claim Construction Order sets forth the Court's construction of disputed words and phrases tendered to the Court for construction.

II. STANDARDS AND PROCEDURES FOR CLAIM CONSTRUCTION

A. General Principles of Claim Construction

Claim construction is a matter of law, to be decided exclusively by the Court. Markman, 517 U.S. at 387. In accordance with the Patent Local Rules of the Northern District, the parties submit their joint selection of the ten disputed terms that are significant in resolving the case as well as their proposed definitions for construction. See Patent L.R. 4-3. After the Markman hearing and upon consideration of the parties' briefs, the Court issues an order construing the meaning of the disputed terms. The Court's construction becomes the legally operative meaning of the disputed terms that governs further proceedings in the case. See Chimie v. PPG Indus., Inc., 402 F.3d 1371, 1377 (Fed.

---

[7] In 2006, Defendants filed a patent infringement suit based upon three of the Patents-in-Suit in this matter–the '336 Patent, the '148 Patent and the '584 Patent–in the Eastern District of Texas. (See Order Denying Motions to Dismiss, to Transfer Venue, and to Stay at 3, Docket Item No. 47 in No. C 08-00877 JW (discussing the Texas action).) Defendants brought that action against unrelated third parties. (See id.) On June 15, 2007, Judge Ward issued a Claim Construction Order in the Texas action in which he construed some of the words and phrases from the three patents at issue in that case. See Tech. Props. Ltd. v. Matsushita Elec. Indus. Co., Ltd., 514 F. Supp. 2d 916 (E.D. Tex. 2007).

[8] As of April 30, 2009, "a total of eleven reexamination proceedings had been initiated against the [Patents-in-Suit] in the United States Patent and Trademark Office ('USPTO')." (Order Granting in part Motion to Stay at 2-3, Docket Item No. 144 in No. C 08-00877 JW.) On June 17, 2009, the Court granted in part motions to stay this action pending reexamination of several of the Patents-in-Suit. (See id.) On February 22, 2010, the Court lifted the stay. (See Docket Item No. 156 in No. C 08-00877 JW.)

The reexamination certificate for the '749 Patent was issued on June 7, 2011. (See Declaration of James C. Otteson in Support of Defendants' Opening Claim Construction Brief for the "Top Ten" Terms, hereafter, "Otteson Decl.," Ex. BB, Ex Parte Reexamination Certificate, Docket Item No. 310-6.) The reexamination of the '749 Patent resulted in amendments to Claim 1, among others. Claim 1 of the '749 Patent–which includes multiple disputed terms–was amended to include the two "wherein" clauses. (See id.)

The reexamination certificate for the '336 Patent was issued on December 15, 2009. (See Otteson Decl., Ex. DD, Ex Parte Reexamination Certificate, Docket Item No. 310-8.) The reexamination of the '336 Patent resulted in amendments to Claims 1, 6 and 10, and the addition of Claim 11, among others. (Id.)

Cir. 2005). Although greater weight should always be given to the intrinsic evidence,[9] claim construction is a fluid process in which the Court may consider a number of extrinsic sources of evidence, so long as they do not contradict the intrinsic evidence. See Vitronics Corp. v. Conceptronic, Inc., 90 F.3d 1576, 1582-83 (Fed. Cir. 1996).

B. Construction from the Viewpoint of an Ordinarily Skilled Artisan

A patent's claims define the scope of the patent: the invention that the patentee may exclude others from practicing. Phillips, 415 F.3d at 1312. The Court generally gives the patent's claims their ordinary and customary meaning. In construing the ordinary and customary meaning of a patent claim, the Court does so from the viewpoint of a person of ordinary skill in the art at the time of the invention, which is considered to be the effective filing date of the patent application. Thus, the Court seeks to construe the patent claim in accordance with what a person of ordinary skill in the art would have understood the claim to have meant at the time the patent application was filed. This inquiry forms an objective baseline from which the Court begins its claim construction. Id. at 1313.

The Court proceeds from that baseline under the premise that a person of ordinary skill in the art would interpret claim language not only in the context of the particular claim in which the language appears, but also in the context of the entire patent specification of which it is a part. Phillips, 415 F.3d at 1313. Additionally, the Court considers that a person of ordinary skill in the art would consult the rest of the intrinsic record, including any surrounding claims, the drawings and the prosecution history, if it is in evidence. Id.; see also Teleflex, Inc. v. Fisosa N. Am. Corp., 299 F.3d 1313, 1324 (Fed. Cir. 2002). In reading the intrinsic evidence, a person of ordinary skill in the art would give consideration to whether the disputed term is a term commonly used in lay language, a technical term, or a term defined by the patentee.

C. Commonly Used Terms

In some cases, disputed claim language involves a commonly understood term that is readily apparent to the Court. In such a case, the Court considers that a person of ordinary skill in the art

---

[9] Phillips v. AWH Corp., 415 F.3d 1303, 1324 (Fed. Cir. 2005).

would give the term its widely accepted meaning, unless a specialized definition is stated in the patent specification or was stated by the patentee during prosecution of the patent. In articulating the widely accepted meaning of such a term, the Court may consult a general purpose dictionary. Phillips, 415 F.3d at 1314.

D. Technical Terms

If a disputed term is a technical term in the field of the invention, the Court considers that one of skill in the art would give the term its ordinary and customary meaning in that technical field, unless a specialized definition is stated in the specification or during prosecution of the patent. Phillips, 415 F.3d at 1314. In arriving at this definition, the Court may consult a technical art-specific dictionary or invite the parties to present testimony from experts in the field on the ordinary and customary definition of the technical term at the time of the invention. Id.

E. Defined Terms

It is well established that a patentee is free to act as his or her own lexicographer. See, e.g., Process Control Corp. v. HydReclaim Corp., 190 F.3d 1350, 1357 (Fed. Cir. 1999). Acting as such, the patentee may use a term differently than a person of ordinary skill in the art would understand it, without the benefit of the patentee's definition. Vitronics Corp., 90 F.3d at 1582. Thus, the Court examines the claims and the intrinsic evidence to determine if the patentee used a term with a specialized meaning.

The Court regards a specialized definition of a term stated in the specification as highly persuasive of the meaning of the term as it is used in a claim. Phillips, 415 F.3d at 1316-17. However, the definition must be stated in clear words which make it apparent to the Court that the term has been defined. See id.; Vitronics Corp., 90 F.3d at 1582. If the definition is not clearly stated or cannot be reasonably inferred, the Court may decline to construe the term pending further proceedings. Statements made by the patentee in the prosecution of the patent application as to the scope of the invention may be considered when deciding the meaning of the claims. Microsoft Corp. v. Multi-Tech Systems, Inc., 357 F.3d 1340, 1349 (Fed. Cir. 2004). Accordingly, the Court

may also examine the prosecution history of the patent when considering whether to construe the claim term as having a specialized definition.

In construing claims, it is for the Court to determine the terms that require construction and those that do not. See U.S. Surgical Corp. v. Ethicon, Inc., 103 F.3d 1554, 1568 (Fed. Cir. 1997). Moreover, the Court is not required to adopt a construction of a term, even if the parties have stipulated to it. Pfizer, Inc. v. Teva Pharm. USA, Inc., 429 F.3d 1364, 1376 (Fed. Cir. 2005). Instead, the Court may arrive at its own constructions of claim terms, which may differ from the constructions proposed by the parties.

III. DISCUSSION

Pursuant to the Patent Local Rules, the parties have tendered ten terms that they have identified as significant to resolving these cases. The parties have asked the Court to consider the tendered words and phrases in a particular order. However, because the sequence in which the patents were issued might influence how a person of ordinary skill in the art would understand the patents, the Court will discuss the words and phrases in the order in which they appear in the Patents-in-Suit.[10]

A. '749 Patent

The '749 Patent is entitled: "High Performance, Low Cost Microprocessor Architecture."

Claim 1 of the '749 Patent, as allowed after reexamination, provides:[11]

> A microprocessor system, comprising a central processing unit integrated circuit, a memory external of said central processing unit integrated circuit, a bus connecting said central processing unit integrated circuit to said memory, and means connected to said bus for fetching instructions for said central processing unit integrated circuit on said bus from said memory, said means for fetching instructions being configured and connected to fetch **multiple sequential instructions** from said memory in parallel and **supply the multiple sequential instructions to said central processing unit integrated circuit during a single memory cycle**, said bus having a width at least equal to a number of bits in each of the instructions times a number of the

[10] Subject to further proceedings, the Court's construction of any particular term is presumed to apply consistently across all claims in the Patents-in-Suit in which the term appears. See, e.g., Paragon Solutions, LLC v. Timex Corp., 566 F.3d 1075, 1087 (Fed. Cir. 2009).

[11] Unless otherwise indicated, all bold typeface is added by the Court for emphasis.

> instructions fetched in parallel, said central processing unit integrated circuit including an arithmetic logic unit and **a first push down stack connected to said arithmetic logic unit**, said first push down stack including means for storing a top item connected to a first input of said arithmetic logic unit to provide the top item to the first input and means for storing a next item connected to a second input of said arithmetic logic unit to provide the next item to the second input, a remainder of said first push down stack being connected to said means for storing a next item to receive the next item from said means for storing a next item when pushed down in said push down stack, said arithmetic logic unit having an output connected to said means for storing a top item;
> wherein
> the microprocessor system comprises an **instruction register** configured to store the multiple sequential instructions and from which instructions are accessed and decoded;
> and wherein
> the means for fetching instructions being configured and connected to fetch multiple sequential instructions from said memory in parallel and supply the multiple sequential instructions to the central processing unit integrated circuit during a single memory cycle comprises supplying the multiple sequential instructions in parallel to said instruction register during the same memory cycle in which the multiple sequential instructions are fetched.

Claim 1 recites a microprocessor system. The parties have tendered for construction a number of words and phrases used in Claim 1.

1. "multiple sequential instructions"

Claim 1 recites that the system comprises, among other components, a "means for fetching"[12] that is configured to fetch "multiple sequential instructions." The parties tender for construction the phrase "multiple sequential instructions."

Upon review, the Court finds that this phrase is composed of commonly used words that have a plain and ordinary meaning. There is nothing in the claim or written description that would lead a person of ordinary skill in the art to conclude that the inventors intended to use the phrase with anything other than its plain and ordinary meaning. In particular, the Court finds that the word "multiple" would have been understood, by a person of ordinary skill in the art, to mean "two or more," while the phrase "sequential instructions" would have been understood to mean "computer

[12] For convenience, the Court will refer to this "means" as the "means for fetching limitation."

instruction in a sequential order." Therefore, at this time, the Court declines to use any different words or phrases to construe the phrase "multiple sequential instructions."

2. ". . . configured and connected to . . . supply multiple sequential instructions to central processing unit integrated circuit during a single memory cycle"

Claim 1 recites that the "means for fetching" is configured and connected to supply multiple sequential instructions to the central processing unit "during a single memory cycle." The parties request the Court to decide what, if any, effect the reexamination proceedings had on the meaning of the phrase "during a single memory cycle."[13] Specifically, the issue tendered to the Court is whether the phrase should be defined as requiring a "prefetch buffer."

During reexamination, the inventors, in referring to the phrase "during a single memory cycle," defended allowance of the claim over a prior art reference known as "Edwards" by stating the following:

> Edwards describes the way the Transputer decodes and executes instructions. As described in Edwards, see, e.g., Fig. 8, below, instructions are supplied to a one-instruction-wide instruction buffer, one at a time, and are there decoded. Fetching multiple instructions into a prefetch buffer and then supplying them one at a time is not sufficient to meet the claim limitation–the supplying of "multiple sequential instructions to a CPU during a single memory cycle."[14]

Upon review, the Court does not find that the cited statements constitute a basis for construing the language of Claim 1 to include the presence or configuration of a prefetch buffer.[15]

---

[13] (See, e.g., Plaintiffs' Consolidated Responsive Claim Construction Brief at 26-28, hereafter, "Plaintiffs' Brief," Docket Item No. 315 in No. C 08-00877 JW.)

[14] (See Declaration of Kyle Chen in Support of Plaintiffs' Consolidated Responsive Claim Construction Brief, hereafter, "Chen Decl.," Ex. 16, Amendment in Response to Non Final Office Action in Ex Parte Reexamination Proceedings at 26, Docket Item No. 316-16.)

[15] Plaintiffs cite to three additional statements made by the inventors that purportedly contain similar disavowals. (See Plaintiffs' Brief at 27-28.) However, the Court finds that none of these cited statements refer to a "prefetch buffer." Further, each cited statement expressly distinguishes the alleged invention from the prior art reference on the same basis, namely, that the instructions are supplied to the CPU "during a single memory cycle." (Id.)

Having disposed of the only issue tendered with respect to this phrase, the Court declines to further construe it.[16]

3. "push down stack connected to said arithmetic logic unit"

Claim 1 recites a central processing unit integrated circuit including an arithmetic logic unit and "a first push down stack connected to said arithmetic logic unit." The parties tender for construction the phrase "push down stack connected to said arithmetic logic unit."

As to this phrase, the Court finds that a person of ordinary skill in the art reading the '749 Patent would understand the phrase "push down stack" to mean a last-in, first-out ("LIFO") data storage structure, in which the last item placed (pushed) onto the stack is the first item removed (popped) from the stack.[17] Further, the Court finds that a person of ordinary skill in the art at the time of the invention would understand that a "push down stack" can be implemented using a dedicated top-of-stack register or a logical stack "pointer" to indicate the "top of the stack" element regardless of its location. For example, the written description discusses stack pointers 102 and 104 in Fig. 2.[18]

Finally, with respect to this phrase, the parties dispute whether the "connected to" language should be construed as "directly connected to" or "physically connected to." The claim requires that the push down stack be "connected" to the arithmetic logic unit. The Court finds that a person of

---

[16] The parties did not request the Court to construe the meaning of the phrase "during a single memory cycle."

[17] See, e.g., MODERN DICTIONARY OF ELECTRONICS 603 (7th ed. 1999) (defining a "pushdown stack" as a "circuit that operates in the reverse of a shift register," and explaining that "[w]hereas[] a shift register is a first-in first-out (FIFO) circuit, pushdown stacks are last-in, first-out (LIFO) memories. When data is requested, the stack will read the last data stored, and all other data will move one step closer to the output. Unless memory is emptied, the first data in will never be retrieved."). The same source alternatively defines a "pushdown stack" as "[e]ssentially a last-in, first-out buffer" in which, "[a]s data is added, the stack moves down with the last item, added [sic] taking the top position. Id. Thus, the "[s]tack height varies with the number of stored items, increasing or decreasing with the entering or retrieving of data. The words push (move down) and pop (retrieve the most recently stoked [sic] item) are used to describe its operation." Id.

[18] Referring to Fig. 2, the specification states: "Stack pointer 102, return stack pointer 104, mode register 106 and instruction register 108 are also connected to the internal data bus 90 by lines 110, 112, 114 and 116, respectively." (See '749 Patent, Col. 6:39-42.)

ordinary skill in the art would understand that the stack might be implemented using "pointers," which negates the need to connect the stack directly or physically to the arithmetic logic unit.[19] Therefore, the Court declines to add as a limitation that the connection must be direct or physical.

Accordingly, the Court construes the phrase "push down stack connected to said arithmetic logic unit" to mean:

a last-in-first-out data storage element connected to the arithmetic logic unit.

4. "instruction register"

Claim 1 contains two "wherein" clauses. With respect to the first "wherein" clause, the parties tender for construction the phrase "wherein the microprocessor system comprises an instruction register."[20]

In computer systems, the phrase "instruction register" has a plain and ordinary meaning, namely, a "register in a central processing unit that holds the address of the next instruction to be executed."[21] A person of ordinary skill in the art reading the written description would understand that the inventors are using the phrase with its plain and ordinary meaning:

> Instruction register 108 receives four 8-bit byte instruction words 1-4 on 32-bit internal data bus 90.

('749 Patent, Col. 7:53-55.)[22]

The parties have drawn the Court's attention to a related term that was construed by Judge Ward and that was subsequently affirmed by the Federal Circuit. Judge Ward's construction related to phrases such as "instruction groups" and "operand" in Claim 29 of the '584 Patent. See Tech.

---

[19] See MODERN DICTIONARY OF ELECTRONICS 603 (7th ed. 1999) ("In actual practice, a hardware-implemented pushdown stack is a collection of registers with a counter that serves as a pointer to indicate the most recently loaded register. Registers are unloaded in the reverse of the sequence in which they were loaded.").

[20] The Court notes that both the body of the claim and the first "wherein" clause disclose a microprocessor system *comprising* recited limitations. However, conventional claim language would have the wherein clause formatted to provide that "the microprocessor system *further* comprises . . ." to avoid any confusion between the wherein clause and the body of the claim.

[21] See MICROSOFT COMPUTER DICTIONARY 276 (5th ed. 2002).

[22] The Court notes that the phrase "8-bit byte" is unusual and appears to be redundant.

Props. Ltd., 514 F. Supp. 2d at 931-34. The claims of the '584 Patent deal specifically with an embodiment that includes "variable width operands." (See '584 Patent, Col. 16:7-26.) This particular embodiment requires all operands to be right justified in the instruction register so that the microprocessor can quickly locate the operands of variable width without the need "to specify the different operand sizes." (See '584 Patent, Col. 16:24-26.) However, unlike Claim 29 of the '584 Patent, Claim 1 of the '749 Patent does not contain such phrases. Thus, the Court does not find Judge Ward's construction pertinent.

Because the Court finds that the language of the claim has been used with its plain and ordinary meaning, the Court declines to further construe it.[23]

B. '890 Patent

Claim 11 of the '890 Patent[24] provides:

> A microprocessor, which comprises a main central processing unit and a **separate direct memory access central processing unit** in a single integrated circuit comprising said microprocessor, said main central processing unit having an arithmetic logic unit, a first push down stack with a top item register and a next item register, connected to provide inputs to said arithmetic logic unit, an output of said arithmetic logic unit being connected to said top item register, said top item register also being connected to provide inputs to an internal data bus, said internal data bus being bidirectionally connected to a loop counter, said loop counter being connected to a decrementer, said internal data bus being bidirectionally connected to a stack pointer, return stack pointer, mode register and instruction register, said stack pointer pointing into said first push down stack, said internal data bus being connected to a memory controller, to a Y register of a return push down stack, an X register and a program counter, said Y register, X register and program counter providing outputs to an internal address bus, said internal address bus providing inputs to said memory controller and to an incrementer, said

[23] The Court notes that in a summary of an in-person interview with the examiner issued on October 25, 1994, the examiner noted with respect to Claim 1: "operand width is variable and right adjusted." (See Chen Decl., Ex. 19, Examiner Interview Summary Record, Docket Item No. 316-20.) The statement appears to have been made in an attempt to distinguish prior art known as "Boufarah," and the Court finds that it may potentially impose a limitation on the type of operands that are to be used and the positioning of the operands in the instruction register. The Court finds that a full understanding of the meaning of this statement and the events that gave rise to it might be relevant to the present analysis. Thus, the Court finds that it would benefit from further briefing as to this issue, as discussed below.

[24] The '890 Patent and the '336 Patent were filed on the same day. However, the '890 Patent was issued earlier than the '336 Patent. (See Chen Decl. ¶¶ 2, 12 (stating that the '890 Patent was issued on June 25, 1996, while the '336 Patent was issued on September 15, 1998).)

incrementer being connected to said internal data bus, said direct memory access central processing unit providing inputs to said memory controller, said memory controller having an address/data bus and a plurality of control lines for connection to a random access memory.

The parties tender for construction the phrase "separate direct memory access central processing unit."

Claim 11 provides two separate central[25] processing units ("CPU"): a "main" CPU and a "direct memory access" ("DMA") CPU. The Court finds that a person of ordinary skill in the art would understand "CPU" to mean a unit of a computing system that fetches, decodes, and executes programmed instructions.[26] In the written description, the inventors use the term CPU consistently with its plain and ordinary meaning.[27]

Further, the written description criticizes "[c]onventional microprocessors" that use "DMA controllers" because "some processing by the main central processing unit (CPU) of the microprocessor is required."[28] With respect to the DMA CPU, the written description states that an object of the invention is to provide a microprocessor "in which DMA does not require use of the main CPU during DMA requests and responses and which provides very rapid DMA response with predictable response times."[29]

---

[25] The parties agree that a person of ordinary skill would understand "central" processing unit to refer to a processing unit, and that the word "central" does not necessarily connote the primary processor in a particular hierarchy.

[26] See, e.g., MODERN DICTIONARY OF ELECTRONICS 107 (7th ed. 1999) (defining a CPU as "[t]hat unit of a computing system that fetches, decodes, and executes programmed instructions and maintains the status of results as the program is executed").

[27] (See, e.g., '890 Patent, Col. 8:22-24 ("The DMA CPU 72 controls itself and has the ability to fetch and execute instructions. It operates as a co-processor to the main CPU 70 (FIG. 2) for time specific processing.").)

[28] ('890 Patent, Col. 1:52-58.)

[29] ('890 Patent, Col. 2:2-5.)

Accordingly, the Court construes the term "separate direct memory access central processing unit" to mean:

> **a central processing unit that accesses memory and that fetches and executes instructions directly, separately, and independently of the main central processing unit.**

C. '336 Patent

1. Claim 1

Claim 1 of the '336 Patent provides:

> A microprocessor system, comprising
> a single integrated circuit including a central processing unit and an **entire ring oscillator variable speed system clock** in said single integrated circuit and connected to said central processing unit for clocking said central processing unit,
> said central processing unit and said ring oscillator variable speed system clock each including a plurality of electronic devices correspondingly constructed of the same process technology with corresponding manufacturing variations,
> a processing frequency capability of said central processing unit and a speed of said ring oscillator variable speed system clock varying together due to said manufacturing variations and due to at least operating voltage and temperature of said single integrated circuit;
> an on-chip input/output interface connected to exchange coupling control signals, addresses and data with said central processing unit; and
> a second clock independent of said ring oscillator variable speed system clock connected to said input/output interface, wherein a clock signal of said second clock originates from a source other than said ring oscillator variable speed system clock.

The parties tender the phrase "ring oscillator" for construction.

Upon review, the Court finds that one of ordinary skill in the art would understand the phrase "ring oscillator" to mean: "interconnected electronic components comprising multiple odd numbers of inverters arranged in a loop."[30] When a voltage is applied, the ring oscillator generates signals that are used by the processing unit to regulate the timing of its operations. In contrast with a circuit

[30] The parties agree that a "ring oscillator" is "an oscillator having a multiple, odd number of inversions arranged in a loop," which is the construction arrived at by Judge Ward in the Texas action, though they disagree about whether additional limitations should be added to Judge Ward's construction of the term. (See Plaintiffs' Brief at 3; Defendants' Opening Claim Construction Brief for the "Top Ten" Terms at 16-17, Docket Item No. 310 in No. C 08-00877 JW.)

that receives its timing signal from an external clock, a person of ordinary skill in the art reading the patent would understand that Claim 1 claims a "single integrated circuit," fabricated so as to include a "ring oscillator."

At issue is whether the phrase "ring oscillator" should be given a specialized meaning based on statements made by the inventors during reexamination of Claims 4 and 8 of the '148 Patent.[31]

Claim 4 of the '148 Patent claims in pertinent part:

> A microprocessor integrated circuit comprising . . . a ring oscillator having a variable output frequency, wherein the ring oscillator provides a system clock to the processing unit, the ring oscillator disposed on said integrated circuit substrate.

Claim 8 of the '148 Patent has a similarly worded limitation.

During reexamination, the examiner reviewed the allowance of Claims 4 and 8 over U.S. Patent No. 4,689,581 ("Talbot"). The Talbot Patent, which is entitled "Integrated Circuit Phase Locked Loop Timing Apparatus," claims:

> an integrated circuit device . . . and a timing apparatus . . . formed on a common single chip, said timing apparatus comprising a phase locked loop [comprising, *inter alia*] a voltage controlled oscillator arranged to be controlled by [a] voltage signal to produce [an] output timing signal at its output.

(Talbot, Col. 10:48-11:9.)

Preliminarily, the examiner rejected Claims 4 and 8 of the '148 Patent as unpatentable over Talbot. During the course of reexamination proceedings, the examiner conducted an interview with the patent owner and discussed whether Claims 4 and 8 were allowable over Talbot.[32] Afterward,

---

[31] Because the '148 Patent shares the same specification with the '336 Patent and is directly related to the other three Patents-in-Suit, the Court finds that any representation regarding similar terms made by the inventors during the prosecution of the '148 Patent is relevant to its consideration and construction of the terms in the '336 Patent. See Microsoft Corp. v. Multi-Tech Sys., Inc., 357 F.3d 1340, 1350 (Fed. Cir. 2004) ("Any statement of the patentee in the prosecution of a related application as to the scope of the invention would be relevant to claim construction.").

[32] (See Otteson Decl., Ex. X, Ex Parte Reexamination Interview Summary, Docket Item No. 310-2.)

the examiner prepared and sent to the patent owner an "Interview Summary."[33] Specifically, with respect to the discussion of Talbot, the examiner wrote:

> Continuing, the patent owner further argued that the reference of Talbot does not teach of a "ring oscillator." The patent owner discussed features of a ring oscillator, such as being **non-controllable**, and being **variable based on the environment**. **The patent owner argued that these features distinguish over what Talbot teaches**. The examiner will reconsider the current rejection based on a forthcoming response, which will include arguments similar to what was discussed.[34]

In its post-interview submission, the patent owner reiterated the contention that the claim should be allowed because Talbot disclosed a "voltage-controlled oscillator" and not the "ring oscillator" disclosed in the claim:

> Further, Talbot does not teach, disclose, or suggest the ring oscillator recited in claim 4. The Examiner cited col. 3, ll. 26-36, and oscillator circuit 12 shown in FIG. 1 of Talbot as teaching the recited ring oscillator. Talbot discusses a voltage-controlled oscillator (VCO) 12, but does not teach or disclose a ring oscillator.[35]

During the course of these claim construction proceedings, the inventors have continued to maintain that Talbot was overcome during reexamination because it does not disclose a "ring oscillator."[36]

---

[33] An examiner's interview summary may serve as a basis for finding a prosecution disclaimer that narrows the claim scope. See, e.g., Rheox, Inc. v. Entact, Inc., 276 F.3d 1319, 1322 (Fed. Cir. 2002); Biovail Corp. Int'l v. Andrx Pharms., Inc., 239 F.3d 1297, 1302-04 (Fed. Cir. 2001).

[34] (See Chen Decl., Ex. 4, Ex Parte Reexamination Interview Summary, Docket Item No. 316-4 (emphasis added).)

[35] (Otteson Decl., Ex. Y, Remarks/Arguments at 11, hereafter, "Remarks," Docket Item No. 310-3.)

[36] For instance, Defendants argued during the Markman hearing that the inventors' written submission distinguished the Talbot reference because Talbot lacked a ring oscillator and never mentioned a requirement of "non-controllability." Further, Defendants also refer to the inventors' written response on February 21, 2008, which states:

> Further, **Talbot does not teach, disclose, or suggest the ring oscillator** recited in claim 4. ... Talbot discusses a voltage-controlled oscillator (VCO) 12, but **does not teach or disclose a ring oscillator**. Talbot provides two different implementations of the VCO 12 in FIGS. 3-4, **neither one of which is a ring oscillator**. Talbot refers to the oscillator of FIG. 3 as a "frequency controlled oscillator" (col. 7, ll. 21-22) and the oscillator of FIG. 4 simply as a "voltage controlled oscillator" (col. 8, ll. 59-65). As the sole inventor of the cited reference,

The Court has examined the Talbot patent. Although the component is, indeed, referred to as a "voltage-controlled oscillator," declarations and other extrinsic materials that have been tendered during the claim construction proceedings call into question the validity of the inventors' contention to the PTO and to this Court that the "ring oscillator" is different from the "voltage-controlled oscillator" disclosed in Talbot. On the one hand, the Court has received extrinsic evidence that the voltage-controlled oscillator disclosed in Talbot *is* a ring oscillator. On the other hand, arguments have been submitted claiming that the voltage-controlled oscillator of Talbot *is not* a ring oscillator.[37]

Under clear Federal Circuit law, a submission made by an inventor during reexamination is regarded as a disavowal only if the court finds that the allegedly disavowing statement is "so clear as to show reasonable clarity and deliberateness, and so unmistakable as to show unambiguous evidence of disclaimer." Omega Eng'g, Inc. v. Raytek Corp., 334 F.3d 1314, 1325 (Fed. Cir. 2003) (citations omitted).

Here, before arriving at a decision on the definition of the phrase "ring oscillator" in the context of the Talbot reference, the Court finds that it would benefit from further briefing. In the supplement briefs, the declarants shall fully articulate the technical basis for their opinions with respect to whether the voltage-controlled oscillator disclosed in Talbot is or is not a ring oscillator. The Court will return to the construction of the phrase "ring oscillator" following the completion of the supplement briefing.

> Talbot presumably possesses at least ordinary skill in the art, yet Talbot did not characterize either of the disclosed oscillators as ring oscillators. Applicants respectfully assert that the reason they were not characterized by Talbot as ring oscillators is because **they are not ring oscillators**. For at least the foregoing reasons, **Talbot does not teach, disclose, or suggest a ring oscillator** as recited in the claims. (Remarks at 11 (emphases added).)

[37] This issue is important to claim construction, because it is relevant to understanding in what manner the ring oscillator is "non-controllable," as distinguished from the voltage-controlled oscillator disclosed in Talbot. Resolving this conflict might affect how the Court approaches issues with respect to the validity of the patent claim at issue.

2. Claim 6

Claim 6 of the '336 Patent provides:

> A microprocessor system comprising:
> a central processing unit disposed upon an integrated circuit substrate, said central processing unit operating at a processing frequency and being constructed of a first plurality of electronic devices;
> an entire oscillator disposed upon said integrated circuit substrate and connected to said central processing unit, said oscillator **clocking said central processing unit** at a clock rate and being constructed of a second plurality of electronic devices, thus varying the processing frequency of said first plurality of electronic devices and the clock rate of said second plurality of electronic devices in the same way **as a function of parameter variation** in one or more fabrication or operational parameters associated with said integrated circuit substrate, thereby enabling said processing frequency to track said clock rate in response to said parameter variation; an on-chip input/output interface, connected between said central processing unit and an off-chip external memory bus, for facilitating exchanging coupling control signals, addresses and data with said central processing unit; and
> an off-chip external clock, independent of said oscillator, connected to said input/output interface wherein said off-chip external clock is operative at a frequency independent of a clock frequency of said oscillator and wherein a clock signal from said off-chip external clock originates from a source other than said oscillator.

a. "clocking said central processing unit"

The parties tender for construction the phrase "clocking said central processing unit."

Upon review, the Court finds that to one of ordinary skill in the art, the plain and ordinary meaning of "clocking said central processing unit" is to provide a clock signal to the central processing unit.

A further issue tendered with respect to this phrase is whether, based on the written description, the construction should include a limitation of the maximum or optimum frequency of the "clocking" function. In the written description of the '336 Patent, the phrase "maximum frequency possible" is used with respect to an embodiment.[38] A description of an embodiment in the specification may not be imposed as a limitation "unless the patentee has demonstrated a clear

[38] (See '336 Patent, Col. 16:67-17:2 (stating that "[b]y deriving system timing from the ring oscillator 430, CPU 70 will always execute at the maximum frequency possible, but never too fast.").)

intention to limit the claim scope using 'words or expressions of manifest exclusion or restriction.'" Innova/Pure Water, Inc. v. Safari Water Filtration Sys., Inc., 381 F.3d 1111, 1117 (Fed. Cir. 2004) (citation omitted). Here, the Court finds that the cited language does not demonstrate "a clear intention to limit the claim scope." Id.

Accordingly, the Court construes "clocking said central processing unit" to mean:

providing a timing signal to said central processing unit.

b. "as a function of parameter variation"

The parties tender for construction the phrase "as a function of parameter variation." The full phrase is: "thus varying the processing frequency of said first plurality of electronic devices and the clock rate of said second plurality of electronic devices in the same way **as a function of parameter variation."**

The disputed issue is whether the phrase requires a mathematical type predetermined functional relationship. Upon review, the Court finds that a person of ordinary skill in the art reading the patent would understand that the phrase "as a function of" is describing a variable that depends on and varies with another.[39] Because neither the written description nor the prosecution history provide a basis for concluding that the phrase should be limited to a narrower definition of an exact mathematical type functional relationship, the Court declines to do so. Having resolved the only dispute tendered with respect to this phrase, the Court declines to construe it further.

3. Claim 10

Claim 10 of the '336 Patent provides:

> In a microprocessor system including a central processing unit, a method for clocking said central processing unit comprising the steps of:
>
> providing said central processing unit upon an integrated circuit substrate, said central processing unit being constructed of a

[39] The Court observes that "function" is a very broad term. See, e.g., MODERN DICTIONARY OF ELECTRONICS 311-12 (7th ed. 1999) (defining "function" as, *inter alia*, a "quantity of value that depends on the value of one or more other quantities" or a "specific purpose of an entity, or its characteristic action," and defining a number of phrases that include the term "function," such as "function codes," "function keys" and a "function table").

> first plurality of transistors and being operative at a processing frequency;
>
> **providing an entire variable speed clock disposed upon said integrated circuit substrate**, said variable speed clock being constructed of a second plurality of transistors;
>
> clocking said central processing unit at a clock rate using said variable speed clock with said central processing unit being clocked by said variable speed clock at a variable frequency dependent upon variation in one or more fabrication or operational parameters associated with said integrated circuit substrate, said processing frequency and said clock rate varying in the same way relative to said variation in said one or more fabrication or operational parameters associated with said integrated circuit substrate;
>
> connecting an on-chip input/output interface between said central processing unit and an off-chip external memory bus, and exchanging coupling control signals, addresses and data between said input/output interface and said central processing unit; and
>
> clocking said input/output interface using an off-chip external clock wherein said off-chip external clock is operative at a frequency independent of a clock frequency of said variable speed clock and wherein a clock signal from said off-chip external clock originates from a source other than said variable speed clock.

The parties have tendered for construction the phrase "providing an entire variable speed clock disposed upon said integrated circuit substrate." There are two issues that are tendered with respect to this language. First, there is a dispute over whether the "variable speed clock" should be defined as limited to a ring oscillator. Here, the Court observes that, in other claims, the inventor discusses a "ring oscillator" as a variable speed system clock. Nonetheless, with respect to this Claim, the Court declines to limit the broader phrase found in Claim 10 to a ring oscillator only.

Second, the parties tender a dispute over the degree of independence between the signal of the "variable speed clock" and any external reference signal. However, upon review the Court finds that this dispute is not pertinent to the construction of the tendered phrase.

Accordingly, the Court construes "providing an entire variable speed clock disposed upon said integrated circuit substrate" to mean:

> **Providing a variable speed clock that is located entirely on the same semiconductor substrate as the central processing unit.**

4. Claim 11

Claim 11 of the '336 Patent provides:

> A microprocessor system, comprising a single integrated circuit including a central processing unit and an entire ring oscillator variable speed system clock in said single integrated circuit and connected to said central processing unit for clocking said central processing unit, said central processing unit and said ring oscillator variable speed system clock each including a plurality of electronic devices correspondingly constructed of the same process technology with corresponding manufacturing variations, a processing frequency capability of said central processing unit and a speed of said ring oscillator variable speed system clock varying together due to said manufacturing variations and due to at least operating voltage and temperature of said single integrated circuit; an on-chip input/output interface connected to exchange coupling control signals, addresses and data with said central processing unit; and a second clock independent of said ring oscillator variable speed system clock connected to said input/output interface, **wherein said central processing unit operates asynchronously to said input/output interface**.

The parties tender for construction the phrase "wherein said central processing unit operates asynchronously to said input/output interface."

Claim 11 discloses a microprocessor system comprising, among others, a central processing unit and an entire ring oscillator variable speed system clock connected to said central processing unit, an on-chip input/output interface, and "a second clock independent of said ring oscillator variable speed system clock" connected to said input/output interface. The subject phrase is contained in a "wherein" clause that describes the relationship between the timing control signal of the central processing unit and the timing signal of the on-chip input/output interface. The claim discloses that the central processing unit operates "asynchronously" to the input/output interface.

The written description is silent as to whether there is or can be *any* timing relationship between the central processing unit and the input/output interface or between their respective clocks.

The inventors first introduced the term "operates asynchronously to" during the re-examination of the '336 Patent in order to "clarify the meaning of 'independent' as recited in the

claims."[40] The examiner had focused on a reference known as "Kato" that purported to show two clock signals that are "in synchronism with each other." (Id. at 19.) The inventors explained that "Kato does not reveal any teaching that any of the components of the data processing circuit operate asynchronously with each other." (Id.) In support of the "independent" and "asynchronous" nature of its clocks, the inventors cited a textbook that describes what an asynchronous system is:

> An *asynchronous* system is one containing two or more independent clock signals. So long as each clock drives independent logic circuitry, such a system is effectively a collection of independent synchronous systems. **The logical combination of signals derived from independent clocks, however, poses difficulty because of the unpredictability of their phase relationship**.[41]

Reading this prosecution history, a person of ordinary skill would understand that the word "asynchronously"[42] means that the timing signal from one clock is independent from and not derived from the other clock such that a phase relationship between the two clocks is not readily predictable.

Accordingly, the Court construes "wherein said central processing unit operates asynchronously to said input/output interface" to mean:

> **the timing control of the central processing unit operates independently of and is not derived from the timing control of the input/output interface such that there is no readily predictable phase relationship between them.**

IV. CONCLUSION

The Court has construed the phrases and terms tendered for construction.

On or before **June 29, 2012**, the parties shall meet and confer and file a Joint Statement addressing the following issues:

---

[40] (See Declaration of Eugene Mar in Support of Defendants' Opening Claim Construction Brief, Ex. G, In re Ex Parte Reexamination of U.S. Patent No. 5,809,336 at 17, Docket Item No. 213-2.)

[41] (Id. (citing STEPHEN A. WARD & ROBERT H. HALSTEAD, JR., COMPUTATION STRUCTURES 93 (1990)) (emphasis added).)

[42] One source provides nine different meanings for the term "asychronous." See MODERN DICTIONARY OF ELECTRONICS 40 (7th ed. 1999) (defining the term, *inter alia*, as a "communication method in which data is sent when it is ready without being referenced to a timing clock, rather than waiting until the receiver signals that it is ready to receive" or as referring to "computer program execution [that is] unexpected or unpredictable with respect to the instruction sequence").

(1) A proposed schedule for supplemental briefs consistent with the terms of this Order;

(2) In light of the Court's impending retirement,[43] the Court proposes to assign this case to Magistrate Judge Grewal. In their Statement, the parties shall state whether they jointly consent to having this case immediately reassigned to Judge Grewal. In the event the parties do not consent to the immediate reassignment, the case will remain with Judge Ware and be subject to reassignment in due course.

Dated: June 12, 2012

JAMES WARE
United States District Chief Judge

[43] On April 28, 2012, Chief Judge Ware announced that he plans to "retire in August 2012 as the terms of his current law clerks come to an end." See Chief Judge Ware Announces Transition, *available at* http://www.cand.uscourts.gov/news/82.

THIS IS TO CERTIFY THAT COPIES OF THIS ORDER HAVE BEEN DELIVERED TO:

Deepak Gupta dgupta@fbm.com
Eugene Y. Mar emar@fbm.com
Harold H. Davis harold.davis@klgates.com
James Carl Otteson jim@agilityiplaw.com
Jas S Dhillon jas.dhillon@klgates.com
Jeffrey M. Fisher jfisher@fbm.com
Jeffrey Michael Ratinoff jeffrey.ratinoff@klgates.com
John L. Cooper jcooper@fbm.com
Kyle Dakai Chen kyle.chen@cooley.com
Mark R. Weinstein mweinstein@cooley.com
Michelle Gail Breit mbreit@agilityiplaw.com
Nan E. Joesten njoesten@fbm.com
Paul A. Alsdorf palsdorf@fbm.com
Samuel Citron O'Rourke eupton@whitecase.com
Stephanie Powers Skaff sskaff@fbm.com
Timothy Paar Walker timothy.walker@klgates.com
William Sloan Coats william.coats@kayescholer.com

Dated: June 12, 2012

Richard W. Wieking, Clerk

By: /s/ JW Chambers
William Noble
Courtroom Deputy

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN JOSE DIVISION

ACER, INC., ACER AMERICA CORPORATION and GATEWAY, INC., Plaintiffs, v. TECHNOLOGY PROPERTIES LTD., PATRIOT SCIENTIFIC CORPORATION, ALLIACENSE LTD., Defendants.	Case No. 5:08-cv-00877 PSG **(Re: Docket Nos. 356, 357, 358, 374)**
HTC CORPORATION, HTC AMERICA, INC., Plaintiffs, v. TECHNOLOGY PROPERTIES LTD., PATRIOT SCIENTIFIC CORPORATION, ALLIACENSE LTD., Defendants.	Case No. 5:08-cv-00882 PSG **(Re: Docket Nos. 385, 387, 388, 403)**

CLAIM CONSTRUCTION ORDER

In this patent infringement suit, Plaintiffs Acer, Inc., Acer America Corp., Gateway, Inc., and Plaintiffs HTC Corp. and HTC America, Inc. (collectively "Plaintiffs") seek a declaratory

judgment that they do not infringe patents owned by Defendants Technology Properties Ltd., Patriot Scientific Corp., and Alliacense Ltd. (collectively "Defendants").[1] Consistent with Pat. L.R. 4-3(c), the parties seek further construction of terms and phrases in claims in the patents-in-suit.[2] Plaintiffs and Defendants each also seek reconsideration of Judge Ware's earlier constructions of certain terms.[3]

As part of those motions for reconsideration, Plaintiffs seek to file a sur-reply on the grounds that Defendants' reply to their motion for reconsideration introduced new arguments and new evidence.[4] The court GRANTS Plaintiffs' motion to file the sur-reply.

In light of this case's long history and the trial date set for June 24, 2013, the court does not wish to add any further delay to the constructions by its preparation of a complete opinion setting forth its reasoning and analysis. To that end, the court at this time will simply issue its constructions without any significant reasoning and analysis:

CLAIM TERM	**CONSTRUCTION**
"instruction register"	Register that receives and holds one or more instructions for supplying to circuits that interpret the instructions
"ring oscillator"	an oscillator having a multiple, odd number of inversions arranged in a loop, wherein the oscillator is variable based on the temperature, voltage and process parameters in the environment
"separate DMA CPU"	a central processing unit that accesses memory and that fetches and executes instructions directly and separately of the main central processing unit
"supply the multiple sequential instructions"	provide the multiple sequential instructions in parallel (as opposed to one-by-one) to said central processing unit integrated circuit during

[1] Unless otherwise noted, the docket citations refer to Case No. 5:08-cv-00882 PSG.

[2] *See* Docket Nos. 387, 394.

[3] *See* Docket Nos. 385, 388.

[4] *See* Docket No. 403.

	a single memory cycle
"clocking said CPU"	Providing a timing signal to said central processing unit

The parties should rest assured that the court arrived at these constructions with a full appreciation of not only the relevant intrinsic and extrinsic evidence, but also the Federal Circuit's teaching in *Phillips v. AWH Corp.*,[5] and its progeny. So that the parties may pursue whatever recourse they believe is necessary, a complete opinion will issue before entry of any judgment.

IT IS SO ORDERED.

Dated: December 4, 2012

PAUL S. GREWAL
United States Magistrate Judge

[5] 415 F.3d 1303, 1312-15 (Fed. Cir. 2005).

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN JOSE DIVISION

ACER, INC., ACER AMERICA CORPORATION and GATEWAY, INC., Plaintiffs, v. TECHNOLOGY PROPERTIES LTD., PATRIOT SCIENTIFIC CORPORATION, ALLIACENSE LTD., Defendants.	Case No. 5:08-cv-00877 PSG **(Re: Docket Nos. 356, 357, 358, 374)**
HTC CORPORATION, HTC AMERICA, INC., Plaintiffs, v. TECHNOLOGY PROPERTIES LTD., PATRIOT SCIENTIFIC CORPORATION, ALLIACENSE LTD., Defendants.	Case No. 5:08-cv-00882 PSG **(Re: Docket Nos. 385, 387, 388, 403)**

CLAIM CONSTRUCTION ORDER

On November 30, 2012, following reassignment of this case to the undersigned with the consent of the parties and in light of the retirement of Chief Judge Ware, and the completion of an

extended *Markman* hearing, the court issued an order from the bench construing five of the parties' disputed terms. The court provided a written summary of its constructions a few days later.[1] The court now explains its reasoning below.

I. BACKGROUND

In this suit, Plaintiffs Acer, Inc., Acer America Corp., Gateway, Inc., HTC Corp., and HTC America, Inc.[2] seek a declaratory judgment that they do not infringe patents owned by Defendants Technology Properties, Patriot Scientific, and Alliacense (collectively "TPL"). All of the patents at issue relate to various aspects of microprocessors.

On November 30, 2012, the court held a claim construction hearing to consider five disputed terms. Prior to the case being reassigned to the undersigned, Judge Ware considered the same five terms.[3] He construed three of them and asked for more briefing on two of them, although he also provided a tentative construction for the two.[4]

The Eastern District of Texas also has considered related terms in another case that TPL filed in 2006 against unrelated third parties. In that case, Judge Ward held a claim construction hearing and issued a decision construing terms based upon patents with the same specification as the patents at issue in this suit.[5] Several terms he construed overlap with terms at issue here. Although the case resolved before proceeding to trial, TPL appealed a portion of the claim construction ruling to the Federal Circuit with respect to one of the three patents in suit; the Federal Circuit affirmed the district court's judgment against TPL.[6]

[1] *See* Docket No. 381.

[2] Barco N.V. was originally a party and was a party to the motions at issue, but is no longer involved in the case.

[3] *See* Docket No. 336.

[4] *See id.*

[5] *See Tech. Properties Ltd. v. Matsushita Elec. Indus. Co., Ltd.*, 514 F. Supp. 2d 916, 927 (E.D. Tex. 2007) *aff'd sub nom.*, 276 F. App'x 1019 (Fed. Cir. 2008). At issue were United States Patent Nos. 5,809,336, 6,598,148, and 5,784,584.

[6] *See Tech. Properties Ltd., Inc. v. Arm, Ltd.*, 276 F. App'x 1019 (Fed. Cir. 2008).

The terms at issue are found in United States Patent No. 5,440,749 (the "'749 Patent") titled "High Performance, Low Cost Microprocessor Architecture,"[7] United States Patent No. 5,809,336 (the "'336 Patent") titled "High Performance Microprocessor Having Variable Speed System Clock,"[8] and United States Patent No. 5,530,890 (the "'890 Patent"), titled "High Performance, Low Cost Microprocessor."[9] All three patents derive from the same original patent application that was subject to a ten-way restriction requirement and eventually resulted in six different patents known as the Moore Microprocessor Portfolio patents, all of which share a common specification.

The '749 Patent claims an invention that accelerates the operation of microprocessors by fetching multiple instructions from memory per memory cycle. Because a CPU can execute instructions faster than it can fetch them from memory, fetching multiple instructions per memory cycle can improve overall performance.

The '336 Patent claims an invention that allows the frequency of a CPU to fluctuate based upon conditions. Traditional microprocessors use fixed frequency clocks to regulate the frequency with which the CPU operates. Fixed clocks generally have to be set lower than the CPU's maximum possible frequency to ensure proper operation under the worst-case conditions. The '336 Patent claims an invention that solves this problem by placing a ring oscillator on the same microchip as the CPU to act as the clock. Because the ring oscillator is on the same microchip and made out of the same components as the CPU, it is subject to the same environmental conditions and thus it will operate at a variable speed based upon conditions allowing the CPU to operate at higher rates during good conditions and lower rates during bad.

The '890 Patent relates to microprocessor architecture and claims a direct memory access mechanism. Most microprocessors have a direct memory access controller that handles the slow operation of reading and writing to memory so that the CPU can execute other instructions while waiting. The patent discloses a direct memory access CPU, which can execute some instructions in addition to reading and writing to memory for the CPU.

---

[7] *See* Docket No. 358-2.

[8] *See* Docket No. 358-6.

[9] *See* Docket No. 368-2.

II. LEGAL STANDARDS

Claim construction is exclusively within the province of the court.[10] "To construe a claim term, the trial court must determine the meaning of any disputed words from the perspective of one of ordinary skill in the pertinent art at the time of filing."[11] This requires a careful review of the intrinsic record, comprised of the claim terms, written description, and prosecution history of the patent.[12] While claim terms "are generally given their ordinary and customary meaning," the claims themselves and the context in which the terms appear "provide substantial guidance as to the meaning of particular claim terms."[13] Indeed, a patent's specification "is always highly relevant to the claim construction analysis."[14] Claims "must be read in view of the specification, of which they are part."[15]

Although the patent's prosecution history "lacks the clarity of the specification and thus is less useful for claim construction purposes," it "can often inform the meaning of the claim language by demonstrating how the inventor understood the invention and whether the inventor limited the invention in the course of prosecution, making the claim scope narrower than it would otherwise be."[16] The court also has the discretion to consider extrinsic evidence, including dictionaries, scientific treatises, and testimony from experts and inventors. Such evidence, however, is "less significant than the intrinsic record in determining the legally operative meaning of claim language."[17]

Judge Ware has already considered all of the terms currently before the court. Although the court granted leave for parties to file motions for reconsideration, it will take as its starting point that

---

[10] *See Markman v. Westview Instruments, Inc.*, 517 U.S. 370, 387 (1996).

[11] *Chamberlain Group, Inc. v. Lear Corp.*, 516 F.3d 1331, 1335 (Fed. Cir. 2008).

[12] *See id.*; *Phillips v. AWH Corp*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (internal citations omitted).

[13] *Phillips*, 415 F.3d at 1312, 1314.

[14] *Id.* at 1312-15.

[15] *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 979 (Fed. Cir. 1995) (en banc), *aff'd*, 517 U.S. 370 (1996); *see also Ultimax Cement Mfg. Corp v. CTS Cement Mfg. Corp.*, 587 F. 3d 1339, 1347 (Fed. Cir. 2009).

[16] *Phillips*, 415 F.3d at 1317 (internal quotations omitted).

[17] *Id.* (internal quotations omitted).

the earlier constructions are correct. Consistent with Local Rule 7-9, absent newly discovered material facts, change in law, or manifest failure to consider material facts or arguments, the court will not alter any earlier constructions.[18]

III. CLAIM CONSTRUCTION

A. "instruction register"

Plaintiffs' Proposed Construction	TPL's Proposed Construction
Register that receives and holds one or more instructions for supplying to circuits that interpret the instructions, in which any operands that are present must be right-justified in the register	Register that receives and holds one or more instructions for supplying to circuits that interpret the instructions

The parties dispute the construction of "instruction register" as used in claim 1 of the '749 Patent. The term "instruction register" was added to a wherein clause in claim 1 of the '749 patent during reexamination. The patent claims a microprocessor system

> wherein the microprocessor system comprises an instruction register configured to store the multiple sequential instructions and from which instructions are accessed and decoded.[19]

Judge Ware tentatively construed "instruction register" in the '749 patent as having its plain and ordinary meaning.[20] Quoting a dictionary, he determined that instruction register meant a "register in a central processing unit that holds the address of the next instruction to be executed."[21] After construing the term, the court noted that the prosecution history might convince the court to limit its construction and requested more briefing.[22]

The parties agree that the term has a slightly different meaning than the one the court previously adopted because the court's previous definition came from a software dictionary and the patents are hardware-related. The parties agree that the meaning of "instruction register" in the

---

[18] *See Therasense, Inc. v. Becton, Dickinson & Co.*, 560 F. Supp. 2d 835, 844 (N.D. Cal. 2008) (following courts in the Northern District of California that "have required a litigant to meet the Civil Local Rule 7-9 standard when requesting reconsideration of a claim construction").

[19] *See* Docket No. 358-2, Reexam. Cert., col.1 ll.55-60.

[20] *See* Docket No. 336 at 11.

[21] *Id.* at 10 (quoting MICROSOFT COMPUTER DICTIONARY 276 (5th ed. 2002)).

[22] *See id.* at 11 n.23.

context of hardware is a "register that receives and holds one or more instructions for supplying to circuits that interpret the instructions." The court takes this construction as its starting point.

TPL urges the court to keep this construction while Plaintiffs argue for a more limited construction requiring that the operands in the register be right-justified. Even though Judge Ware's prior order indicated he was interested in an explanation of the prosecution history, the parties' arguments remain focused on the specification.

Plaintiffs argue that the specification requires the right-justified limitation for the register that it seeks. The Federal Circuit has instructed that "the specification may reveal a special definition given to a claim term by the patentee that differs from the meaning it would otherwise possess" or "reveal an intentional disclaimer."[23] However, only a clear disclaimer can justify narrowing the construction.[24] Where a patent consistently references a certain limitation or a preferred embodiment as the present invention, that also can serve to limit the scope of the invention where no other intrinsic evidence suggests otherwise.[25]

Here, Plaintiffs rely on a section of the patent specification that explains that the patented invention is able to use variable width operands because "operands must be right justified in the instruction register."[26] The specification describes this limitation as necessary to make the "magic" of the patent possible.[27] Plaintiffs argue that this is the equivalent of defining the "present invention," but the intrinsic evidence does not clearly support this limitation.

First, the right justified limitation is not a clear and consistent limitation given the overall context of the patent and the specification. The '749 patent is derived from an application that was subject to a ten-way restriction requirement that eventually resulted in six different patents. The original application, which eventually issued as the '749 patent disclosed all of the inventions in

---

[23] *Phillips*, 415 F.3d at 1316.

[24] *See Voda v. Cordis Corp.*, 536 F.3d 1311, 1320 (Fed. Cir. 2008).

[25] *See Absolute Software, Inc.*, 659 F.3d at 1136.

[26] *See* Docket No. 358-2 at col.18 ll.43-45.

[27] *Id.*

what is now their extensive shared specification.[28] Plaintiffs rely on one small section of the common specification, with the heading "Variable Width Operands," covering about twenty lines of the thirty-three column specification.[29] Although this small section contains strong limiting language, because the specification is common to ten different inventions, it does not necessarily apply to the '749 Patent. In fact, Judge Ware previously held that one of those inventions, disclosed in the '584 patent, deals specifically with variable width operands.[30] But variable width operands are not essential to what is claimed in the '749 Patent. Claim 1 of the '749 Patent, the claim at issue here, does not contain the term operand or require variable width operands. Although parties focus on the '749 patent, the same reasoning applies to the '890 Patent.

Second, the specification actually discloses an embodiment where the operands are not right justified. In one embodiment, the instruction register receives four 8-bit instructions.[31] The specification disclosed two instructions, the "Read-Local-Variable XXXX" and "Write-Local-Variable XXXX," which are fixed width instructions that have a 4-bit opcode and a 4-bit operand.[32] These instructions can go into any of the four 8-bit slots in the instruction register and thus would contain operands that are not right justified.[33] At oral argument, Plaintiffs disputed TPL's characterization of these embodiments, arguing that the "4-bit operands" are not actually operands, but the location in temporary storage where the operand actually exists.[34] Even if the location in temporary storage is not a traditional operand, it acts similarly to one and adds further intrinsic evidence supporting a finding that the right justified limitation does not apply to the '749 and '890 patents.

---

[28] *See generally*, Docket No. 358-2 at col.1-35.

[29] *See* Docket No. 358-2 at col.18 ll.35-56.

[30] *See* Docket No. 336 at 11.

[31] *See* Docket No. 358-2 at col.7 ll.50-58.

[32] *See* Docket No. 358-2 at col.31-32 ll.45-15.

[33] *See generally*, *id.* at col.7 ll.50-58.

[34] *See* Docket No. 382 at 106-07.

Plaintiffs do briefly cite to the prosecution history where, in a handwritten summary of an in-person interview in response to a Patent Office Action rejecting several of the claims of a related patent, the examiner stated "Claim 1: Operand width is variable + right adjusted."[35] Because various claims were withdrawn, however it is unclear to exactly what claim the examiner referred. This is not clear and unmistakable disavowal by the applicant.[36]

The parties agreed upon meaning alone should control. Accordingly, the court construes "instruction register" as the "register that receives and holds one or more instructions for supplying to circuits that interpret the instructions."

B. "ring oscillator"

Plaintiffs' Proposed Construction	TPL's Proposed Construction
an oscillator having a multiple, odd number of inversions arranged in a loop, wherein the oscillator is (1) non-controllable; and (2) variable based on the temperature, voltage and process parameters in the environment	an oscillator having a multiple, odd number of inversions arranged in a loop

The parties ask the court to construe the term "ring oscillator" as it is used in claim 1 of the '336 Patent. Judge Ware held that one of ordinary skill in the art would understand the term to mean "interconnected electronic components comprising multiple odd numbers of inverters arranged in a loop."[37] However, he ordered more briefing as to whether the court should give the terms a specialized meaning based upon the statements of the inventors during reexamination to distinguish their invention from the Talbot Patent.[38]

Once again, the parties agree on the basic meaning of the term, but dispute additional limitations. They agree that the meaning of the term is at least "an oscillator having a multiple, odd

[35] Docket No. 363-19 at 2.

[36] *See Univ. of Pittsburgh of Commonwealth Sys. of Higher Educ. v. Hedrick*, 573 F.3d 1290, 1297 (Fed. Cir. 2009) (finding a "patentee may limit the meaning of a claim term by making a clear and unmistakable disavowal of scope during prosecution," but an examiner's summary of disavowal may only create a "weak inference" of the disavowal); *3M Innovative Properties Co. v. Avery Dennison Corp.*, 350 F.3d 1365, 1373 (Fed. Cir. 2003) (finding that prosecution history "cannot be used to limit the scope of a claim unless the *applicant* took a position before the PTO." (emphasis in the original)).

[37] Docket No. 336 at 13.

[38] *Id.* at 14-16.

number of inversions arranged in a loop." TPL urges the court to adopt meaning alone while the Plaintiffs argue that the term must be further limited to be: (1) non-controllable and (2) variable based on temperature, voltage, and process parameters in the environment. Plaintiffs argue that the prosecution history and specification support their position. As explained below, the prosecution history is too ambiguous to support Plaintiffs' construction in full, but the specification and especially the claim language do support Plaintiffs' second limitation.

1. Prosecution history

A "clear and unmistakable" disavowal by the patentee during prosecution or reexamination can narrow the scope of a claim.[39] However, because the "ongoing negotiations between the inventor and the examiner" can "often produce ambiguities," the doctrine only applies to "unambiguous disavowals."[40]

In the patent examiner's summary of his meeting with the patent owner, he wrote that

> the patent owner further argued that the reference of Talbot does not teach of a 'ring oscillator.' The patent owners discussed features of a ring oscillator, such as being non-controllable and being variable based upon the environment. The patent owner argued that these features distinguish over what Talbot teaches.[41]

The examiner finished his summary noting that he would "reconsider the current rejection based upon a forthcoming response, which will include arguments similar to what was discussed."[42] The subsequent written response argued that the Talbot reference did not teach a ring oscillator generally, and did *not* specifically argue that the ring oscillator was "non-controllable."[43] The examiner accepted this argument and withdrew the rejection.[44]

---

[39] *Grober v. Mako Products, Inc.*, 686 F.3d 1335, 1341 (Fed. Cir. 2012), reh'g denied (Sept. 14, 2012).

[40] *Id.*

[41] Docket No. 357-5 at 5. The interview summary relates to the '148 patent, but it shares the same specification with the '336 patent.

[42] *Id.*

[43] *See id.*

[44] *Id.* at 27.

Plaintiffs argue that the examiner's summary is a clear disavowal that should limit the scope of the claim. The court disagrees. The Federal Circuit has suggested that where, as here, the "disavowal" is only an examiner's summary of a patentee's statement, it only creates a "weak inference" of a disavowal.[45] The subsequent prosecution history does not support Plaintiffs' claim construction because the patent owner appears to have made a different argument in his written reply, simply stating that the Talbot reference did not include a ring oscillator *generally* and not distinguishing the ring oscillator of the '336 Patent based on the examiner's stated exemplary features of ring oscillators.[46]

During prosecution, the patent owner also stated that the "the oscillator or variable speed clock varies in frequency but does not require manual or programmed inputs or external or extra components to do so."[47] This statement is not a disavowal because it only affirms that external inputs are "not required." The statement does not clearly impose a prohibition on all types of control.

2. Specification

Plaintiffs also argue that the specification supports their proposed construction. The specification describes the "ring oscillator" as having its frequency "determined by the parameters of temperature, voltage, and process."[48] Although this portion of the specification appears to disclose the preferred embodiment rather than constitute an express limitation on the claimed invention,[49] Claim 1 of the '336 Patent *claims* that the processing frequency of the CPU and the ring

[45] *See Univ. of Pittsburgh*, 573 F.3d at 1297.

[46] *See generally*, *Innova/Pure Water, Inc. v. Safari Water Filtration Sys., Inc.*, 381 F.3d 1111, 1124 (Fed. Cir. 2004) (describing a series of exchanges between the patent owner and the examiner as the parties "talking past one another" and finding no clear evidence of a disavowal from the confused exchange).

[47] Docket No. 363-4 at 6.

[48] *See* Docket No. 358-6 at col.16 ll.59-60.

[49] *See Brookhill-Wilk 1, LLC. v. Intuitive Surgical, Inc.*, 334 F.3d 1294, 1301-02 (Fed. Cir. 2003) ("statements from the description of the preferred embodiment are simply that-descriptions of a preferred embodiment. . . Absent a clear disclaimer of particular subject matter, the fact that the inventor anticipated that the invention may be used in a particular manner does not limit the scope to that narrow context.")

oscillator vary together due to manufacturing variations, operating voltage, and temperature.[50] The claim itself provides that the "ring oscillator" is "constructed of the same process technology with corresponding manufacturing variations" on the same single integrated circuit so that its performance will fluctuate with the CPU because they are subject to the same "manufacturing variations" and "operating voltage and temperature."[51] During oral argument, TPL admitted that a ring oscillator on the same microprocessor as the CPU will vary based upon voltage, temperature, and process variations.[52] Therefore, based upon the claim language and the specification, the court finds that the disclosed "ring oscillator" varies with voltage, temperature, and process variations.

Even though the claimed "ring oscillator" is "determined by the parameters of temperature, voltage, and process," it does not necessarily follow, as Plaintiffs' argue, that the "ring oscillator" must be non-controllable.[53] The claims do not mention "controllable" or "non-controllable" in relation to the "ring oscillator" and neither does the specification. The term "non-controllable" is only used by the patent examiner in the prosecution history discussed above. Additionally, in the preferred embodiment, the "ring oscillator" is "determined" by temperature, voltage, and process,[54] which suggests at least one embodiment in which the ring oscillator is controlled.

Because of the clear limitation in the claims that temperature, voltage, and process determine the "ring oscillator's" frequency, the court includes those limitations in the construction of the term, but does not find similar support for importing the "non-controllable" limitation. The court therefore construes "ring oscillator" as "an oscillator having a multiple, odd number of inversions arranged in a loop, wherein the oscillator is variable based on the temperature, voltage and process parameters in the environment."

---

[50] *See* Docket No. 358-6, Reexam. Cert. col.2 ll.3-5.

[51] *Id.* at col.1-2 ll.59-05.

[52] *See* Docket No. 382 at 49:3-7.

[53] *See, e.g., Brookhill-Wilk*, 334 F.3d at 1301-02.

[54] *See* Docket No. 358-6 at col.16 ll.59-60.

C. "separate DMA CPU"

Plaintiffs' Proposed Construction	TPL's Proposed Construction
a central processing unit that accesses memory and that fetches and executes instructions directly, separately, and independently of the main central processing unit	Electrical circuit for reading and writing to memory that is separate from a main CPU

Judge Ware previously construed the term "separate direct memory access central processing unit" ("separate DMA CPU") from Claim 11 of the '890 Patent. Claim 11 claims

> A microprocessor, which comprises a main central processing unit and a separate direct memory access [DMA] central processing unit [CPU] in a single integrated circuit comprising said microprocessor . . .

The court construed "separate DMA CPU," consistent with its plain and ordinary meaning as "a central processing unit that accesses memory and that fetches and executes instructions directly, separately, and independently of the main central processing unit."[55] Plaintiffs urge the court to keep this construction while TPL argues that previously unaddressed parts of the prosecution history support a different construction broad enough to include standard DMA controllers, which do not execute instructions.

TPL's primary argument is that the history of the Moore patents supports a broader construction. TPL argues that the DMA CPU that fetches and executes its own instructions was one of the ten categories of inventions derived from the original application, but not the invention that eventually became the patent at issue, the '890 Patent. As explained above, the original patent application for what became the '749 Patent was subject to a ten-way restriction. A restriction indicates that "two or more independent and distinct inventions are claimed in one application."[56] One of these 10 categories of inventions was focused on a "microprocessor system having a DMA for fetching instruction[s] for a CPU and itself."[57] The patentee eventually abandoned this application. The '890 Patent came from a different category of invention "drawn to a microprocessor architecture."[58] TPL argues that because the '890 Patent came from a different

[55] Docket No. 336 at 13.

[56] 35 U.S.C. § 121.

[57] Docket No. 368-7 at 3.

[58] *Id*. *See also* Docket No. 356 at 3-4.

invention category, it should not be read to include the definition of the "DMA CPU" that was the subject of another invention.

The court disagrees. The fact that one abandoned patent focused on a particular subject matter does not necessarily mean that same subject matter cannot be within the scope of another related patent based upon the same specification. First, restriction requirements have little, if any, evidentiary weight.[59] Second, there is nothing in the claims to suggest that "DMA CPU" should have anything other than its plain and ordinary meaning. Third, the specification supports the plain and ordinary meaning. The specification discloses a "DMA CPU" in figures 2 and 9. When describing figure 2, the specification states that the "DMA CPU 72 controls itself and has the ability to fetch and execute instructions. It operates as a co-processor to the main CPU 70."[60] The "DMA CPU 314" in figure 9 is part of another microprocessor that the specification describes as equivalent to the microprocessor in figure 2.[61] A separate passage in a later section of the specification describes another embodiment where the "DMA processor 72 of the microprocessor 50 has been replaced with a more traditional DMA controller 314."[62] The specification goes on to describe the characteristics of a DMA controller. These sections are clear that a DMA controller is distinct from a DMA CPU and the patent refers to each by name where appropriate. Thus where the patent claims a DMA CPU, it means a DMA CPU and not a DMA controller.

TPL also argues that statements made during reexamination by the requester and the examiner support its position. The court disagrees. First, the examiner and the reexamination requester made the cited statements, not the patent owner.[63] Second, regardless of who made the

[59] *See Honeywell Int'l, Inc. v. ITT Indus., Inc.*, 452 F.3d 1312, 1319 (Fed. Cir. 2006); *Rambus Inc. v. Hynix Semiconductor Inc.*, 569 F. Supp. 2d 946, 962 (N.D. Cal. 2008) ("In laying out the details of the original restriction requirement, the court recognizes its limited evidentiary significance.").

[60] *See* Docket No. 368-2 at col.8 ll.22-24.

[61] *See id.* at col.9 ll.5-6.

[62] *Id.* at col.12 ll.62-65.

[63] *See 3M Innovative Properties Co.*, 350 F.3d at 1373 (finding that prosecution history "cannot be used to limit the scope of a claim unless the *applicant* took a position before the PTO."(emphasis in the original)).

statements, they do not clearly show that the term "DMA CPU" was understood to include a DMA controller.[64]

During oral argument, TPL argued that the term "independently" in the original construction is unsupported.[65] The court agrees with this point. Even if the DMA CPU fetches and executes its own instructions, it cannot do so independently. The reason for putting the CPU and DMA CPU on the same chip is so they can work together.[66] Otherwise, the evidence in support of changing the court's prior construction is unpersuasive.

The court construes "separate DMA CPU" as "a central processing unit that accesses memory and that fetches and executes instructions directly and separately of the main central processing unit."

D. "supply the multiple sequential instructions"

Plaintiffs' Proposed Construction	TPL's Proposed Construction
provide the multiple sequential instructions in parallel (as opposed to one-by-one) to said central processing unit integrated circuit during a single memory cycle without using a prefetch buffer or a one-instruction-wide instruction buffer that supplies on instruction at a time	provide the multiple sequential instructions in parallel to said central processing unit integrated circuit during a single memory cycle

The parties ask the court to construe the phrase "supply the multiple sequential instructions to said central processing unit integrated circuit during a single memory cycle," from claim 1 of the '749 patent. Judge Ware previously determined that this phrase was composed of commonly used words that the patentee intended to have their plain and ordinary meaning. Plaintiffs argue for a narrower construction based upon disavowals during reexamination while TPL argues for a broad construction. The parties specifically dispute what limitations the patent places on how the "multiple sequential instructions" are provided to the CPU.

---

[64] *See id.* at 1346-47 ("An applicant's silence in response to an examiner's characterization of a claim does not reflect the applicant's clear and unmistakable acquiescence to that characterization if the claim is eventually allowed on grounds unrelated to the examiner's unrebutted characterization.").

[65] *See* Docket No. 382 at 121-22.

[66] *See* Docket No. 368-2, Reexam. Cert., col.1 ll.22-24; Docket No. 368-2 at col.8 ll.22-24 (the DMA CPU "operates as a co-processor to the main CPU").

During reexamination, TPL unambiguously disavowed that instructions could be provided to the CPU one-by-one. The PTO issued a reexamination rejecting claims in the '749 Patent, including claim 1, based upon the "Edwards" patent[67] and an article by Doug MacGregor.[68] To distinguish the Edwards patent, TPL argued that in the Edwards patent, "instructions are supplied to a one-instruction-wide instruction buffer, one at a time," while for the '749 Patent "[f]etching multiple instructions into a prefetch buffer and then supplying them one at a time is not sufficient to meet the claim limitation—the supplying of 'multiple sequential instructions to a CPU during a single memory cycle.'"[69] Similarly, in distinguishing the invention in MacGregor, TPL wrote that "non-parallel supplying of instructions to the CPU is not supplying them to the CPU during a single memory cycle as required by the claim."[70] By this language, TPL clearly and unambiguously disavowed supplying instructions to the CPU one-by-one.

Plaintiffs also urge the court to find TPL disavowed specific structures or components in the above statements, but these statements as to structures are not clearly disavowals because they are made in the context of describing the prior art. There may be ways of incorporating such structures consistent with not supplying the instructions one-by-one.

Accordingly, the court construes the phrase "supply the multiple sequential instructions to said central processing unit integrated circuit during a single memory cycle" as "provide the multiple sequential instructions in parallel (as opposed to one-by-one) to said central processing unit integrated circuit during a single memory cycle."

E. "clocking said CPU"

Plaintiffs' Proposed Construction	TPL's Proposed Construction
timing the operation of the CPU such that it will always execute at the maximum frequency possible, but never too fast	timing the operation of the CPU

[67] U.S. Patent No. 4,680,698.

[68] Doug MacGregor *et al.*, "The Motorola MC68020," IEEE Micro 101 (August 1984).

[69] Docket No. 358-3 at 27.

[70] *Id.* at 46.

The parties ask the court to construe "clocking said CPU," which appears in claims 1, 6, and 10 of the '336 Patent. Generally speaking, "clocking the CPU" refers to using the system clock to control the speed of the CPU. Judge Ware previously considered "clocking said CPU" and based upon the plain and ordinary meaning of the term, construed it as "providing a timing signal to said central processing unit." The court considered other language in the written description that suggested a more limited construction, but ultimately determined that the patentee had not "demonstrated a clear intention to limit the claim scope."[71] Similarly, Judge Ward construed a longer term[72] from claim 1 containing the term "clocking said CPU" as "an oscillator that generates the signal(s) used for timing the operation of the CPU."[73] In construing the term, Judge Ward similarly did not adopt the type of limiting language that Plaintiffs advocate.

As discussed above and explained in the patent, the disclosed invention uses a variable speed clock—a ring oscillator—that varies with temperature, voltage, and process. The specification states that "[b]y deriving system time from the ring oscillator 430, CPU 70 will always execute at the maximum frequency possible, but never too fast."[74] Plaintiffs argue that this is a clear limitation that should be read into the claims. In general, absent a clear intention to limit the scope of a claim, a description of an embodiment should not limit claim language that otherwise has a broader effect.[75] This rule applies even if the patent only describes a single embodiment.[76] Judge Ware previously considered and rejected Plaintiffs attempt to limit the claim based upon the specification and this court agrees. There is no support in the claim language itself for the requirement that the clock always forces the CPU to operate at its maximum frequency. The court finds that operating at

[71] Docket No. 336 at 17-18 (quoting *Innova/Pure Water*, 381 F.3d at 1117).

[72] Judge Ward construed "an entire ring oscillator variable speed system clock in said single integrated circuit and connected to said central processing unit for clocking said central processing unit."

[73] *Tech. Properties Ltd. v. Matsushita Elec. Indus. Co., Ltd.*, 514 F. Supp. 2d 916, 927 (E.D. Tex. 2007) aff'd sub nom., 276 F. App'x 1019 (Fed. Cir. 2008).

[74] *See* Docket No. 358-6 at col.16-17 ll.63-2.

[75] *See Innova/Pure Water*, 381 F.3d at 1117.

[76] *See id.*

the maximum frequency is merely the preferred embodiment and not the only manner in which the invention can operate.

Plaintiffs also try to introduce evidence from the prosecution history to support their argument. Although Plaintiffs quote a section from the prosecution history where the applicants used the magic words "the present invention," what the applicants disclosed is that the present invention includes a variable speed clock on the same microprocessor as the CPU and thus its speed will vary based upon environmental conditions.[77] This is exactly what is claimed in claim 1. The excerpt goes on to explain that one advantage of the variable speed clock is that it "allows the microprocessor to operate at its fastest safe operating speed,"[78] but again, this is just one embodiment and not necessarily a *requirement* of the invention. Plaintiffs' other citations to the prosecution history are similarly unconvincing.

Because the parties have not convinced the court that the prior construction was in error, the Court declines to change its construction. Accordingly, the court construes "clocking said CPU" as "providing a timing signal to said central processing unit."

IV. CONCLUSION

For the reasons set forth above, the court construes the claims as follows:

CLAIM TERM	CONSTRUCTION
"instruction register"	Register that receives and holds one or more instructions for supplying to circuits that interpret the instructions
"ring oscillator"	an oscillator having a multiple, odd number of inversions arranged in a loop, wherein the oscillator is variable based on the temperature, voltage and process parameters in the environment
"separate DMA CPU"	a central processing unit that accesses memory and that fetches and executes instructions directly and separately of the main central processing unit
"supply the multiple sequential instructions to said central processing unit integrated circuit during a single memory cycle"	provide the multiple sequential instructions in parallel (as opposed to one-by-one) to said central processing unit integrated circuit during

[77] *See* Docket No. 358-9 at 4-5.

[78] *Id.* at 5.

	a single memory cycle
"clocking said CPU"	Providing a timing signal to said central processing unit

Dated: August 21, 2013

PAUL S. GREWAL
United States Magistrate Judge

E-Filed 5/13/2011

IN THE UNITED STATES DISTRICT COURT

FOR THE NORTHERN DISTRICT OF CALIFORNIA

SAN JOSE DIVISION

ACER, INC., ACER AMERICA CORPORATION and GATEWAY, INC.

Plaintiffs,

v.

TECHNOLOGY PROPERTIES LIMITED, PATRIOT SCIENTIFIC CORPORATION, and ALLIACENSE LIMITED,

Defendants.

Case No. 5:08-cv-00877 JF/HRL

HTC CORPORATION, HTC AMERICA, INC.,

Plaintiffs,

v.

TECHNOLOGY PROPERTIES LIMITED, PATRIOT SCIENTIFIC CORPORATION, and ALLIACENSE LIMITED,

Defendants.

Case No. 5:08-cv-00882 JF/HRL

BARCO N.V., a Belgian Corporation Plaintiff, v. TECHNOLOGY PROPERTIES LTD., PATRIOT SCIENTIFIC CORP., ALLIACENSE LTD., Defendants.	Case No. 5:08-cv-05398 JF/HRL **ORDER[1] GRANTING IN PART AND DENYING IN PART DEFENDANTS' MOTIONS TO AMEND INFRINGEMENT CONTENTIONS**

Defendants Technology Properties Ltd., Patriot Scientific Corp., and Alliacense, Ltd. (collectively, "TPL") seek leave to amend their infringement contentions with respect to United States Patent Nos. 5,530,890 ("the '890 patent") and 5,440,749 ("the '749 patent") in each of the above-captioned actions.[2] The Court heard oral argument on April 22, 2011. Because TPL seeks to assert certain claims that it reasonably could not have asserted prior to the reexamination of the patents, the motions will be granted in part and denied in part.

I. BACKGROUND

TPL first sought to amend its preliminary infringement contentions nearly one year ago, after this Court lifted a stay that was imposed pending reexamination of several of the patents-in-suit by the United States Patent and Trademark Office ("USPTO"). Although TPL was permitted to amend its infringement contentions at that time with respect to the '336 patent,[3] the

[1] This disposition is not designated for publication in the official reports

[2] Pursuant to Civil L.R. 3-12, the three actions have been related.

[3] Amendment to the '336 infringement contentions was permitted because all parties agreed that amended invalidity and infringement contentions were needed after the reexamination of the '336 patent. *See* Transcript of Case Management Conference Held on February 12, 2010; Order Following Case Management Conference, filed February 22, 2010.

Court denied TPL's motion to amend its contentions with respect to United States Patent No. 6,598,148 ("the '148 patent") and the '749 and '890 patents, finding that TPL had not been diligent. Order Denying Defendants' Motions to Amend Infringement Contentions, filed September 10, 2010.

TPL now renews its motion to amend its infringement contentions with respect to the '749 and '890 patents based upon subsequent activity by the USPTO.

A. The Reexaminations

In November 2010, the USPTO issued a Notice of Intent to Issue Reexamination Certificate ("NIRC") for the '890 patent. Upon receipt of the NIRC, TPL notified Plaintiffs of its intention to seek leave to assert new claims once the Reexamination Certificate issued. Mar Decl. Ex. B. On March 1, 2011, TPL received the Reexamination Certificate, confirming the patentability of existing claims 5–10 and new claims 11–20. Apart from one clarification to independent claim 11, new claims 11-20 track the patent's original claims 1-10 word-for-word. TPL served the proposed amendments to its infringement contentions the same day. The proposed amended contentions assert claims 7, 9, 11, 12, 13, 17, and 19 against each Plaintiff.

On February 11, 2011, the USPTO issued the NIRC for the '749 patent, confirming the patentability of claims 1-7, 10-20, 21-27, 30, and 34-62. Claim 9 was canceled and replaced by claim 59. TPL expects thirty new claims to be confirmed by the final Reexamination Certificate. Asserting that it wishes to avoid further delay in the instant proceedings, TPL seeks to amend its infringement contentions with respect to the '749 patent before the Certificate issues: it seeks to assert claims 1, 23, 24, 43, 44, 45, 47, 54, 55, and 59 against each Plaintiff. Together, the proposed amendments would add a total of thirteen newly-accused products in the HTC action and one newly-accused product in the Acer action, each of which TPL contends entered or will enter the U.S. market after June 2010, when TPL last attempted to amend its contentions.[4]

II. LEGAL STANDARD

[4] After it filed the instant motions, TPL withdrew its request to add several newly-accused products in the Acer and Barco actions.

An action is governed by the version of the local rules in effect at the time the underlying action is filed. *See Seiko Epson Corp. v. Coretronic Corp.*, No. C 06-06946 MHP, 2008 WL 2563383, at *2 (N.D. Cal. June 23, 2008). Plaintiffs Acer Inc., Acer America Corporation, and Gateway, Inc. (collectively, "Acer") and HTC Corporation and HTC America, Inc. (collectively, "HTC") filed their actions on February 8, 2008. Under the Patent Local Rules in effect at that time, "[a]mendment or modification of the Preliminary or Final Infringement Contentions . . . , other than as expressly permitted in Patent L.R. 3-6, may be made only by order of the Court, which shall be entered only upon a showing of good cause."[5] Patent L.R. 3-7. The Patent Local Rules were amended effective March 1, 2008. Plaintiff Barco, N.V. ("Barco") filed its action on December 1, 2008. The version of Patent Local Rule 3-6 in effect as of that date provides that:

> Amendment of the Infringement Contentions or the Invalidity Contentions may be made only by order of the Court upon a timely showing of good cause. Non-exhaustive examples of circumstances that may, absent undue prejudice to the non-moving party, support a finding of good cause include: (a) a claim construction by the Court different from that proposed by the party seeking amendment; (b) recent discovery of material, prior art despite earlier diligent search; and (c) recent discovery of nonpublic information about the Accused Instrumentality which was not discovered, despite diligent efforts, before the service of the Infringement Contentions.

The Advisory Subcommittee commented that even after March 1, 2008, Patent Local Rule 3-6 would continue to be "regulated by the well-established 'good cause' test." Patent Local Rules Advisory Subcommittee Report at 2. Thus, prior cases discussing the concept of "good cause" remain relevant precedent.

In order to demonstrate good cause, TPL must show first that it was diligent in amending its contentions and then that the non-moving parties will not suffer undue prejudice if the motion to amend is granted. *O2 Micro Int'l Ltd. v. Monolithic Power Sys., Inc.*, 467 F.3d 1355, 1366-68 (Fed. Cir. 2006) (concluding that if a party seeking to amend did not demonstrate diligence, there

---

[5] The applicable version of Patent Local Rule 3-6(a) allows a party alleging infringement to amend its infringement contentions without leave of court if the party believes in good faith that the amendment is required by the court's claim construction ruling or documents produced in connection with the opposing party's invalidity contentions. Here, the Court has not issued a claim construction ruling, nor does TPL allege that it seeks to amend in response to the invalidity contentions served by Plaintiffs.

was "no need to consider the question of prejudice").[6] *See also Johnson v. Mammoth Recreations*, 975 F.2d 604, 609 (9th Cir. 1992) (citation omitted) ("Although the existence or degree of prejudice to the party opposing the modification might supply additional reasons to deny a motion, the focus of the [good cause] inquiry [under Federal Rule of Civil Procedure 16(b)] is upon the moving party's reasons for seeking modification. If that party was not diligent, the inquiry should end."). While the court in *O2 Micro* considered "how quickly the party moves to amend its contentions once a new theory of infringement . . . comes to light," Hon. James Ware & Brian Davy, *The History, Content, Application and Influence of the Northern District of California Patent Local Rules*, 25 SANTA CLARA COMPUTER & HIGH TECH. L.J. 965, 995 (2009), this Court has concluded that "the Court also must address whether the party was diligent in discovering the basis for the proposed amendment." *West v. Jewelry Innovations, Inc.*, No. C07-1812 JF (HRL), 2008 WL 4532558, at *2 (N.D. Cal. Oct. 8, 2008).

The party seeking to amend its contentions bears the burden of establishing diligence. *O2 Micro*, 467 F.3d at 1366-67. "Unlike the liberal policy for amending pleadings, the philosophy behind amending claim charts is decidedly conservative, and designed to prevent the 'shifting sands' approach to claim construction." *LG Elecs. Inc. v. Q-Lity Computer Inc.*, 211 F.R.D. 360, 367 (N.D. Cal. 2002). The rules were "designed to require parties to crystallize their theories of the case early in the litigation and to adhere to those theories once they have been disclosed." *O2 Micro*, 467 F.3d at 1366 n. 12 (quoting *Nova Measuring Instruments Ltd. v. Nanometrics, Inc.*, 417 F. Supp.2d 1121, 1123 (N.D. Cal .2006)). "Nevertheless, judges in this district have recognized that the Patent Local Rules are 'not a straitjacket into which litigants are locked from the moment their contentions are served. There is a modest degree of flexibility, at least near the outset.'" *Halo Electronics, Inc. v. Bel Fuse Inc.*, No. C07-06222 RMW (HRL),

---

[6] Other factors relevant to this inquiry include "the relevance of the newly-discovered prior art, whether the request to amend is motivated by gamesmanship, [and] the difficulty of locating the prior art." *Acco Brands, Inc. v. PC Guardian Anti-Theft Products, Inc.*, No. C 04-03526 SI, 2008 WL 2168379 at *1 (N.D. Cal., May 22, 2008) (citing *Yodlee, Inc. v. CashEdge, Inc.*, No. C 05-01550 SI, 2007 WL 1454259, at *2-3 (N.D. Cal. May 17, 2007)).

2010 WL 3489593, at * 1 (N.D. Cal. Sep. 3, 2010) (citing *Comcast Cable Communications Corp., LLC v. Finisar Corp.*, No. C06-04206, 2007 WL 716131 at *2 (N.D.Cal. Mar.2, 2007).

III. DISCUSSION

After nearly three years of litigation, this case still is in its early stages. Plaintiffs suggest that TPL's current effort to amend its infringement contentions is motivated by gamesmanship, as evidenced by the fact that TPL did not file the instant motions until after claim construction briefing was complete. In response, TPL argues that the timing of its proposed amendments was dictated by the reexamination process, noting that it kept Plaintiffs abreast of developments in that process until it became clear which claims would emerge and in what form.

A. Non-Opposition

HTC and Acer do not object to the substitution of claims 1 and 2 of the '890 patent by replacement claims 11 and 12,[7] nor do they object to the substitution of claim 9 of the '749 patent by replacement claim 59. Each Plaintiff also agrees to permit amendment with respect to claim 1 of the '749 patent in order to address the new limitations that were added to that claim during the reexamination.

B. Diligence

TPL contends that because it was uncertain which claims would survive reexamination, it could not conduct a detailed infringement analysis prior to the issuance of the NIRCs for the '749 and '890 patents. Although TPL brought its previous motion to amend based in part on claims that stood rejected as of June 2010, it certainly was not required to do so. In the present context, diligence does not require that a party awaiting USPTO action assert all potential claims. Instead of promoting an orderly process, such a request would add confusion and uncertainty to the litigation.

After it received the NIRC for the '890 patent, TPL promptly notified Plaintiffs of its intent to amend its '890 contentions (Mar Decl. Ex. B), and it began investigating products that

[7] Barco also does not object to the substitution of claim 1 of the '890 patent by claim 11.

might infringe the pending claims. *See* Mar. 17, 2011 Brataadiredja Decl. ¶ 4. TPL thus was prepared to serve its amended contentions on the same day that the Reexamination Certificate issued. With respect to the '749 patent, TPL offered to assert claims conditionally based on the NIRC, and it completed its investigation of infringing products and served its proposed amended contentions approximately one month after the NIRC issued. *See* Mar. 25, 2011 Brataadiredja Decl. ¶ 3.

HTC points out that as a matter of law, the scope of the claims asserted under the '749 and '890 patents could not have been altered by the reexaminations. 35 U.S.C. § 305 ("No proposed amended or new claim enlarging the scope of a claim of the patent will be permitted in a reexamination proceeding under this chapter."). Accordingly, HTC contends that the universe of potentially infringing products likewise could not have expanded, and TPL did not need to wait for the issuance of the NIRCs before conducting its renewed investigation into infringing products. However, given the fact that TPL did not know which claims would emerge from the reexamination, it was not unreasonable for TPL to investigate potentially infringing products after the NIRCs were issued.

HTC also argues that public information regarding the newly-accused products was available even before TPL sought to amend its contentions in June 2010. However while, information regarding these products may have been available in the form of press releases or other media, it appears that none of the new instrumentalities actually entered the market until after TPL served its amended contentions in May 2010. *See, e.g.*, Chen Decl. Ex. D (press release indicating that the HTC Aria would be available June 20, 2010);[8] *Id.* Ex. F (article indicating that the HTC Desire was released on August 27, 2010). 35 U.S.C. § 271 prohibits the use or sale of infringing products, not the announcement of intent to sell infringing products. Thus, even if public information about these products was available to it prior to June 2010, TPL

[8] Although TPL submitted a corrected version of its proposed amended contentions to HTC on June 22, 2010, two days after the Aria was released, this correction did not alter the substance of its proposed contentions. *See* Corrected Amended Patent Local Rule 3-1 Preliminary Infringement Contentions, June 25, 2010 Mar Decl. Ex. B.

did not have a legal basis for accusing HTC of infringement until the products actually had entered the market.[9] Moreover, as TPL points out, nothing requires parties to bring a motion to amend each time a new product enters the market, as this could cause undue delay in the proceedings and prejudice to all parties involved.

C. Prejudice

Plaintiffs argue that TPL's attempt to assert claims 7 and 9 of the '890 patent in addition to claims 17 and 19 is unnecessary and prejudicial because the latter are mirror images of the former. Acer observes that in order to promote judicial economy courts frequently limit the number of claims that a patentee may assert. *See, e.g., Auto Wax Co. v. Mark V Products*, No. 3:99-CV-0982-M, 2001 WL 292597, at *1 (N.D. Tex. March 14, 2001) (requiring plaintiff to limit number of claims to be tried from 86 to 19); *Fenster Family Patent Holdings, Inc. v. Siemens Medical Solutions USA*, No. 04-0038-JJF, 2005 WL 2304190, at *3 (D. Del. Sept. 20, 2005) (requiring plaintiff to reduce its 90 claims to 10); *Verizon Calif., Inc. v. Ronald A. Katz Tech. Licensing, L.P.*, 326 F.Supp.2d 1060, 1066 (C.D. Cal. 2003) (requiring plaintiff to select a maximum of three representative claims for each patent it contended was infringed). TPL concedes that "[b]ecause the scope of new independent claim 11 is the same as the scope of original independent claim 1, the infringement theories underlying the claim charts for the new claims are the same as those for the original claims." Mar. 17, 2011 Brataadiredja Decl. ¶ 3. Given that assessment, it appears that the assertion of all four claims would be redundant. Because claim 1 has been canceled and all four claims cover the same ground, logic dictates that TPL be limited to the assertion of claims 17 and 19, which depend from surviving claim 11.

Finally, Plaintiffs argue that they will be prejudiced by having to conduct additional prior art research and by having to brief the new claim terms that were added during reexamination. However, any such prejudice is insufficient to outweigh TPL's right to assert new claims and the

---

[9] For this reason, TPL has sought to accuse conditionally the Acer Iconia, which will be released in the United States later this summer. Perhaps TPL could have taken this approach last year with respect to other HTC products it seeks to accuse now. However, it was not required to do so.

Court's interest in resolving the parties' disputes as comprehensively as is possible.

IV. ORDER

Accordingly, TPL's motions will be GRANTED IN PART AND DENIED IN PART. TPL may amend its infringement contentions to: (a) assert claims 11, 12, 13, 17, and 19 of the '890 patent and claims 1, 23, 24, 43, 44, 45, 47, 54, 55, and 59 of the '749 patent against each Plaintiff; (b) include the thirteen newly-accused HTC instrumentalities; and (c) include conditionally the Acer Iconia. A case management conference is hereby scheduled for June 24, 2011 at 10:30 a.m. for the purpose of setting a new date and briefing schedule for a claim construction hearing.

IT IS SO ORDERED.

DATED: May 13, 2011

JEREMY FOGEL
United States District Judge

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN JOSE DIVISION

HTC CORPORATION AND HTC AMERICA, INC., Plaintiffs, v. TECHNOLOGY PROPERTIES LIMITED, et al., Defendants.	Case No. 5:08-cv-00882-PSG **ORDER GRANTING-IN-PART HTC'S MOTION TO CORRECT THE JUDGMENT** **(Re: Docket No. 674)**

Both HTC and TPL agree that the court needs to modify the judgment as it currently stands to incorporate the court's prior order dismissing the '890 patent from this case.[1] Where the parties disagree is what form the modified judgment should take. TPL suggests the court hew closely to the present language of the judgment to which both parties previously agreed.[2] HTC believes it

[1] *See* Docket Nos. 674 and 690.

[2] *See* Docket No. 690 at 3 ("pursuant to the Court's Order dismissing U.S. Patent No. 5,530,890 (the "'890 patent") entered September 19, 2013 (Dkt. No. 594), judgment with respect to the '890 patent is entered as follows:

a) Because Defendants cannot establish entitlement to damages in the present action based on the Court's Summary Judgment Order (issued on September 17, 2013 (Dkt. No. 585)), the Court on September 19, 2013 DISMISSED the Fifth Claim for Relief in HTC's First Amended Complaint (seeking a declaration that HTC does not infringe any valid and enforceable claim of the '890 patent), and Count IV of Defendants' Answer and Counterclaim (alleging infringement of the '890 patent), subject to the conditions of the September 19, 2013 Order (Dkt. No. 594);

would be appropriate to go further by describing the dismissal of the '890 patent as entering judgment in its favor.[3]

The court agrees with TPL that moving well beyond the terms of the court's prior order would be unwarranted in this case. The prior order dismissed the '890 patent because HTC prevailed on its motion for partial summary judgment and was able to avoid a portion of TPL's infringement claims and the potential for money damages. But if the claim had proceeded to trial, broader relief to HTC was available. In particular, HTC may have invalidated the patent altogether. Under such circumstances, language characterizing the dismissal of the'890 patent as a complete victory in favor of HTC is not warranted.

b) The September 19, 2013 Order (id.) shall not affect any other claim or counterclaim asserted in the present action, and shall not impair any rights of Defendants or HTC to challenge on appeal any pretrial ruling by the Court for which an appeal is permissible including, without limitation, any challenge to the Summary Judgment Order's application of the intervening rights doctrine;

c) In the event the Federal Circuit reverses the Summary Judgment Order with respect to application of the intervening rights doctrine to the '890 patent, HTC's declaratory judgment claim and Defendants' counterclaim under the '890 patent will be reinstated and proceed unaffected by the dismissal provided in the September 19, 2013 Order (Dkt. No. 594).).

[3] Docket No. 674 at 3 ("**IT IS FURTHER ORDERED AND ADJUDGED** that pursuant to the Joint Request To Dismiss All Claims Relating to U.S. Patent No. 5,530,890 Under F.R.C.P. 41(a)(2) (Dkt. No. 594), the provisions of which are incorporated herein by reference, judgment is hereby entered in favor of Plaintiffs on Defendants' claim of infringement of U.S. Patent No. 5,530,890.").

In any event, the court finds some modification of the language from the proposed order in this case is warranted. The court adopts the following language:

Pursuant to the court's order dismissing U.S. Patent No. 5,530,890 ("the '890 patent") entered September 19, 2013 (Docket No. 594), judgment with respect to the '890 patent is entered as follows:

a) Because Defendants cannot establish entitlement to damages in the present action based on the court's summary judgment order (issued on September 17, 2013 (Docket No. 585)), the court on September 19, 2013 DISMISSED the Fifth Claim for Relief in HTC's First Amended Complaint (seeking a declaration that HTC does not infringe any valid and enforceable claim of the '890 patent), and Count IV of Defendants' Answer and Counterclaim (alleging infringement of the '890 patent), subject to the conditions of the September 19, 2013 order (Docket No. 594);

b) The September 19, 2013 order (Docket No. 594) shall not affect any other claim or counterclaim asserted in the present action, and shall not impair any rights of Defendants or HTC to challenge on appeal any pretrial ruling by the court for which an appeal is permissible including, without limitation, any challenge to the summary judgment order's application of the intervening rights doctrine;

c) In the event the Federal Circuit reverses the summary judgment order with respect to application of the intervening rights doctrine to the '890 patent, HTC's declaratory judgment claim and Defendants' counterclaim under the '890 patent will be reinstated and proceed unaffected by the dismissal provided in the September 19, 2013 order (Docket No. 594).

A revised judgment consistent with this order will issue.

IT IS SO ORDERED.

Dated: January 21, 2014

PAUL S. GREWAL
United States Magistrate Judge

United States District Court
For the Northern District of California

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN JOSE DIVISION

HTC CORPORATION AND HTC AMERICA, INC., Plaintiffs, v. TECHNOLOGY PROPERTIES LIMITED, et al., Defendants.	Case No. 5:08-cv-00882-PSG **ORDER MODIFYING JUDGMENT** **(Re: Docket No. 674)**

(X) Jury Verdict. This action came before the court for a trial by jury. The issues have been tried and the jury has rendered its verdict.

IT IS SO ORDERED AND ADJUDGED that pursuant to the jury verdict filed October 3, 2013, judgment is entered in favor of Defendants.

IT IS FURTHER ORDERED pursuant to the court's order dismissing U.S. Patent No. 5,530,890 ("the '890 patent") entered September 19, 2013 (Docket No. 594), judgment with respect to the '890 patent is entered as follows:

a) Because Defendants cannot establish entitlement to damages in the present action based on the court's summary judgment order (issued on September 17, 2013 (Docket No. 585)), the court on September 19, 2013 DISMISSED the Fifth Claim for Relief in HTC's First Amended Complaint (seeking a declaration that HTC does not infringe any valid and enforceable claim of the '890 patent), and Count IV of Defendants' Answer and

Counterclaim (alleging infringement of the '890 patent), subject to the conditions of the September 19, 2013 order (Docket No. 594);

b) The September 19, 2013 order (Docket No. 594) shall not affect any other claim or counterclaim asserted in the present action, and shall not impair any rights of Defendants or HTC to challenge on appeal any pretrial ruling by the court for which an appeal is permissible including, without limitation, any challenge to the summary judgment order's application of the intervening rights doctrine;

c) In the event the Federal Circuit reverses the summary judgment order with respect to application of the intervening rights doctrine to the '890 patent, HTC's declaratory judgment claim and Defendants' counterclaim under the '890 patent will be reinstated and proceed unaffected by the dismissal provided in the September 19, 2013 order (Docket No. 594).

IT IS SO ORDERED.

Dated: January 21, 2014

PAUL S. GREWAL
United States Magistrate Judge

US005530890A

United States Patent [19]

Moore et al.

[11] Patent Number: 5,530,890

[45] Date of Patent: Jun. 25, 1996

[54] **HIGH PERFORMANCE, LOW COST MICROPROCESSOR**

[75] Inventors: **Charles H. Moore**, Woodside; **Russell H. Fish, III**, Mt. View, both of Calif.

[73] Assignee: **Nanotronics Corporation**, Eagle Point, Oreg.

[21] Appl. No.: **480,206**

[22] Filed: **Jun. 7, 1995**

Related U.S. Application Data

[62] Division of Ser. No. 389,334, Aug. 3, 1989, Pat. No. 5,440,749.

[51] **Int. Cl.**[6] **G06F 9/22**

[52] **U.S. Cl.** **395/800**; 364/931; 364/925.6; 364/937.1; 364/965.4; 364/232.8; 364/244.3

[58] **Field of Search** 395/375, 500, 395/775, 800

[56] **References Cited**

U.S. PATENT DOCUMENTS

3,603,934	9/1971	Heath	395/181
4,003,033	1/1977	O'Keefe et al.	395/287
4,037,090	7/1977	Raymond	364/706
4,042,972	8/1977	Grunes et al.	395/375
4,050,058	9/1977	Garlic	395/800
4,067,058	1/1978	Derchak	395/740
4,079,455	3/1978	Ozga	395/800
4,110,822	8/1978	Porter	395/375
4,125,871	11/1978	Martin	395/550
4,128,873	12/1978	Lamiaux	395/183.06
4,253,785	3/1981	Chamberlin	375/375
4,354,228	10/1982	Moore et al.	395/800
4,376,977	3/1983	Brunshorst	395/375
4,382,279	5/1983	Mgon	395/800
4,403,303	9/1983	Howes et al.	395/500
4,450,519	5/1984	Guttag et al.	395/800
4,463,421	7/1984	Laws	395/325
4,538,239	8/1985	Magar	364/759

(List continued on next page.)

OTHER PUBLICATIONS

C. Whitby–Strevans, "The transputer", *The 12th Annual International Symposium on Computer Architecture, Conference Proceeings,* Jun. 17–19, 1985, pp. 292–300.

D. W. Best et al., "An Advanced–Architecture CMOS/SOS Microprocessor", *IEEE Micro,* vol. 2, No. 3, Aug. 1982, pp. 11–25.

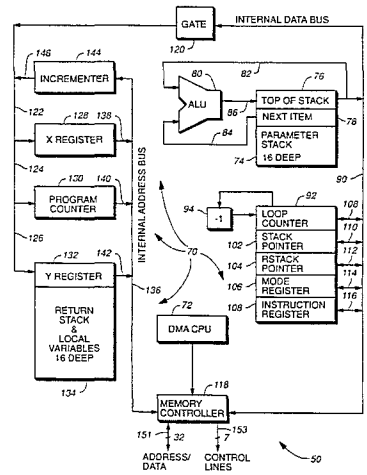
Primary Examiner—David Y. Eng

Attorney, Agent, or Firm—Cooley Godward Castro Huddleson & Tatum

[57] **ABSTRACT**

A microprocessor (**50**) includes a main central processing unit (CPU) (**70**) and a separate direct memory access (DMA) CPU (**72**) in a single integrated circuit making up the microprocessor (**50**). The main CPU (**70**) has a first 16 deep push down stack (**74**), which has a top item register (**76**) and a next item register (**78**), respectively connected to provide inputs to an arithmetic logic unit (ALU) (**80**) by lines (**82**) and (**84**). An output of the ALU (**80**) is connected to the top item register (**76**) by line (**86**). The output of the top item register at (**82**) is also connected by line (**88**) to an internal data bus (**90**). A loop counter (**92**) is connected to a decrementer (**94**) by lines (**96**) and (**98**). The loop counter (**92**) is bidirectionally connected to the internal data bus (**90**) by line (**100**). Stack pointer (**102**), return stack pointer (**104**), mode register (**106**) and instruction register (**108**) are also connected to the internal data bus (**90**) by lines (**110**), (**112**), (**114**) and (**116**), respectively. The internal data bus (**90**) is connected to memory controller (**118**) and to gate (**120**). The gate (**120**) provides inputs on lines (**122**), (**124**), and (**126**) to X register (**128**), program counter (**130**) and Y register (**132**) of return push down stack (**134**). The X register (**128**), program counter (**130**) and Y register (**132**) provide outputs to internal address bus (**136**) on lines (**138**), (**140**) and (**142**). The internal address bus provides inputs to the memory controller (**118**) and to an incrementer (**144**). The incrementer (**144**) provides inputs to the X register, program counter and Y register via lines (**146**), (**122**), (**124**) and (**126**). The DMA CPU (**72**) provides inputs to the memory controller (**118**) on line (**148**). The memory controller (**118**) is connected to a RAM by address/data bus (**150**) and control lines (**152**).

10 Claims, 19 Drawing Sheets



U.S. PATENT DOCUMENTS

4,541,045	9/1985	Kromer	395/375
4,562,537	12/1985	Barnett et al.	395/375
4,577,282	3/1986	Candel et al.	395/800
4,607,332	8/1986	Goldberg	395/375
4,626,988	12/1986	George et al.	395/375
4,649,471	3/1987	Briggs	395/325
4,665,495	5/1987	Thaden	345/185
4,709,329	11/1987	Hecker	395/275
4,713,749	12/1987	Magar et al.	395/375
4,714,994	12/1987	Oklobdzija et al.	395/375
4,720,812	1/1988	Kao et al.	395/700
4,772,888	9/1988	Kimura	340/825.5
4,777,591	10/1988	Chang et al.	395/800
4,787,032	11/1988	Culley et al.	395/725
4,803,621	2/1989	Kelly	395/400
4,860,198	8/1989	Takenaka	395/307
4,870,562	9/1989	Kimoto	395/550
4,931,986	6/1990	Daniel et al.	395/550
5,036,460	7/1991	Takahira	395/425
5,070,451	12/1991	Moore et al.	395/375
5,127,091	6/1992	Bonfarah	395/375

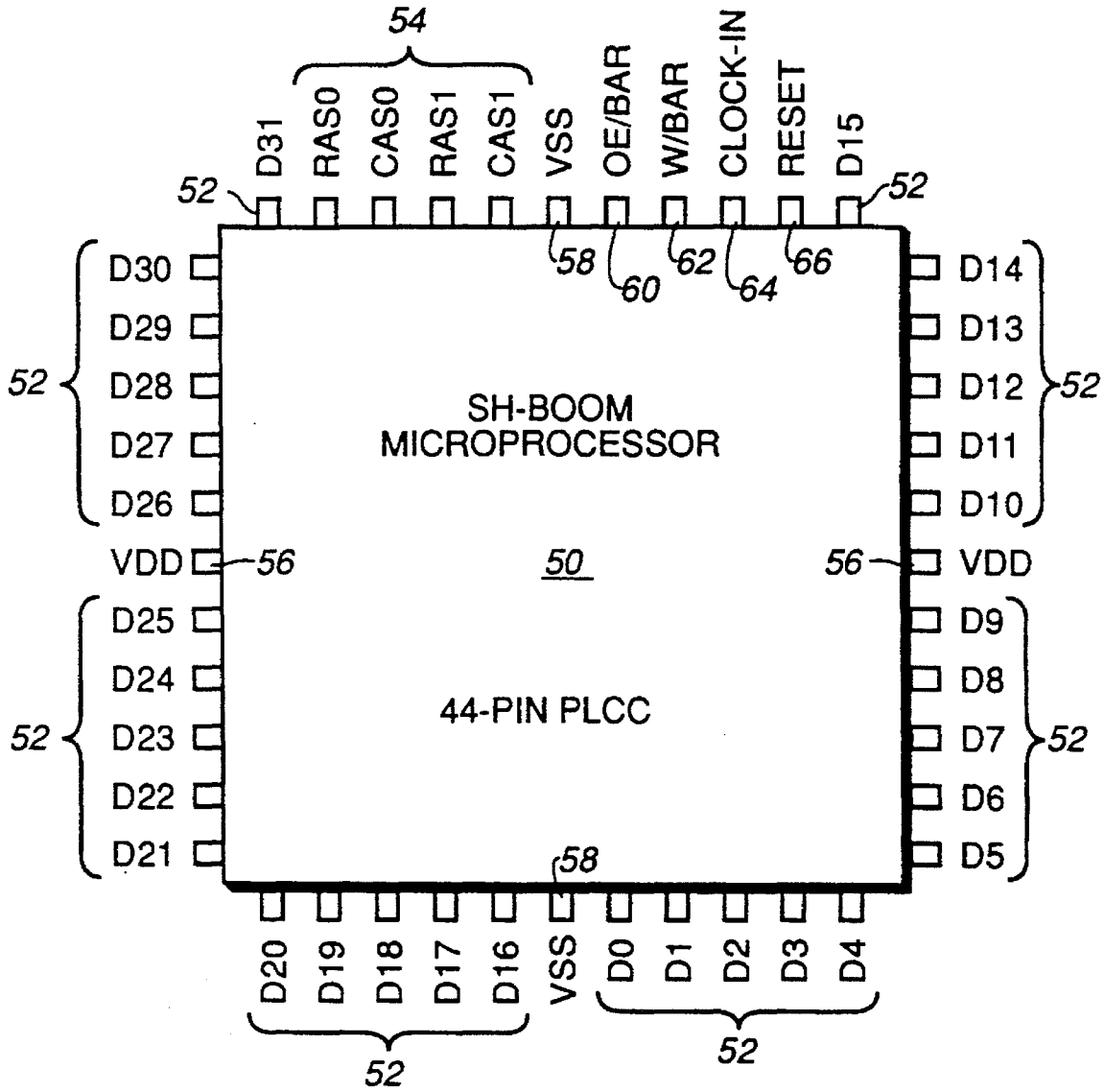

54
D31
RAS0
CAS0
RAS1
CAS1
VSS
OE/BAR
W/BAR
CLOCK-IN
RESET
D15
52
58
60
62
64
66
D30
D29
D28
D27
D26
VDD
D25
D24
D23
D22
D21
D14
D13
D12
D11
D10
VDD
D9
D8
D7
D6
D5
SH-BOOM
MICROPROCESSOR
50
44-PIN PLCC
56
D20
D19
D18
D17
D16
D0
D1
D2
D3
D4

FIG._1

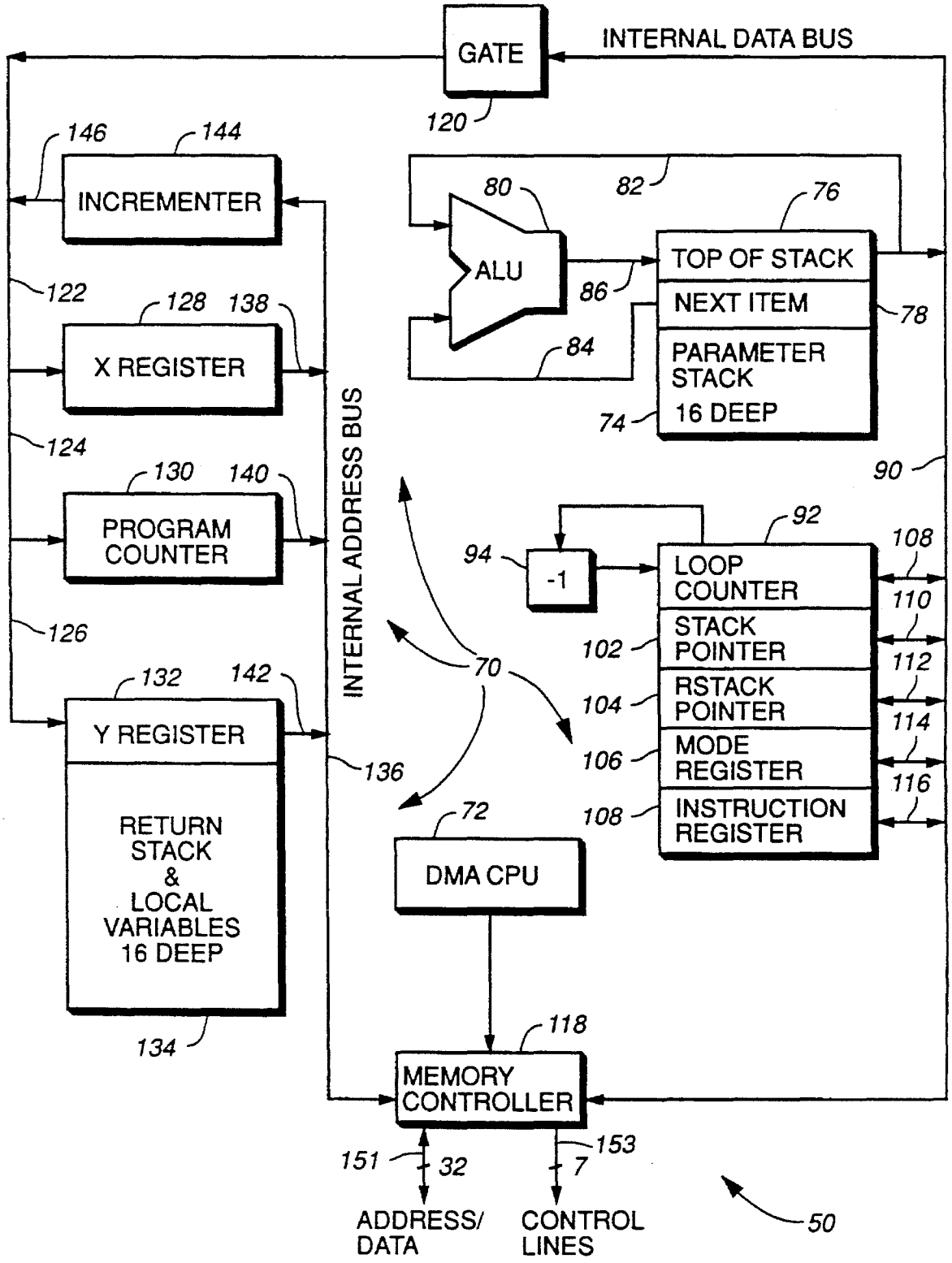

GATE
INTERNAL DATA BUS
120
146
144
INCREMENTER
80
82
76
ALU
TOP OF STACK
NEXT ITEM
PARAMETER STACK
16 DEEP
86
84
78
74
122
128
138
X REGISTER
124
130
140
PROGRAM COUNTER
126
INTERNAL ADDRESS BUS
90
92
94
-1
LOOP COUNTER
STACK POINTER
RSTACK POINTER
MODE REGISTER
INSTRUCTION REGISTER
108
110
112
114
116
102
104
106
108
70
132
142
Y REGISTER
136
RETURN STACK & LOCAL VARIABLES 16 DEEP
134
72
DMA CPU
118
MEMORY CONTROLLER
151
32
153
7
ADDRESS/ DATA
CONTROL LINES
50

FIG._2

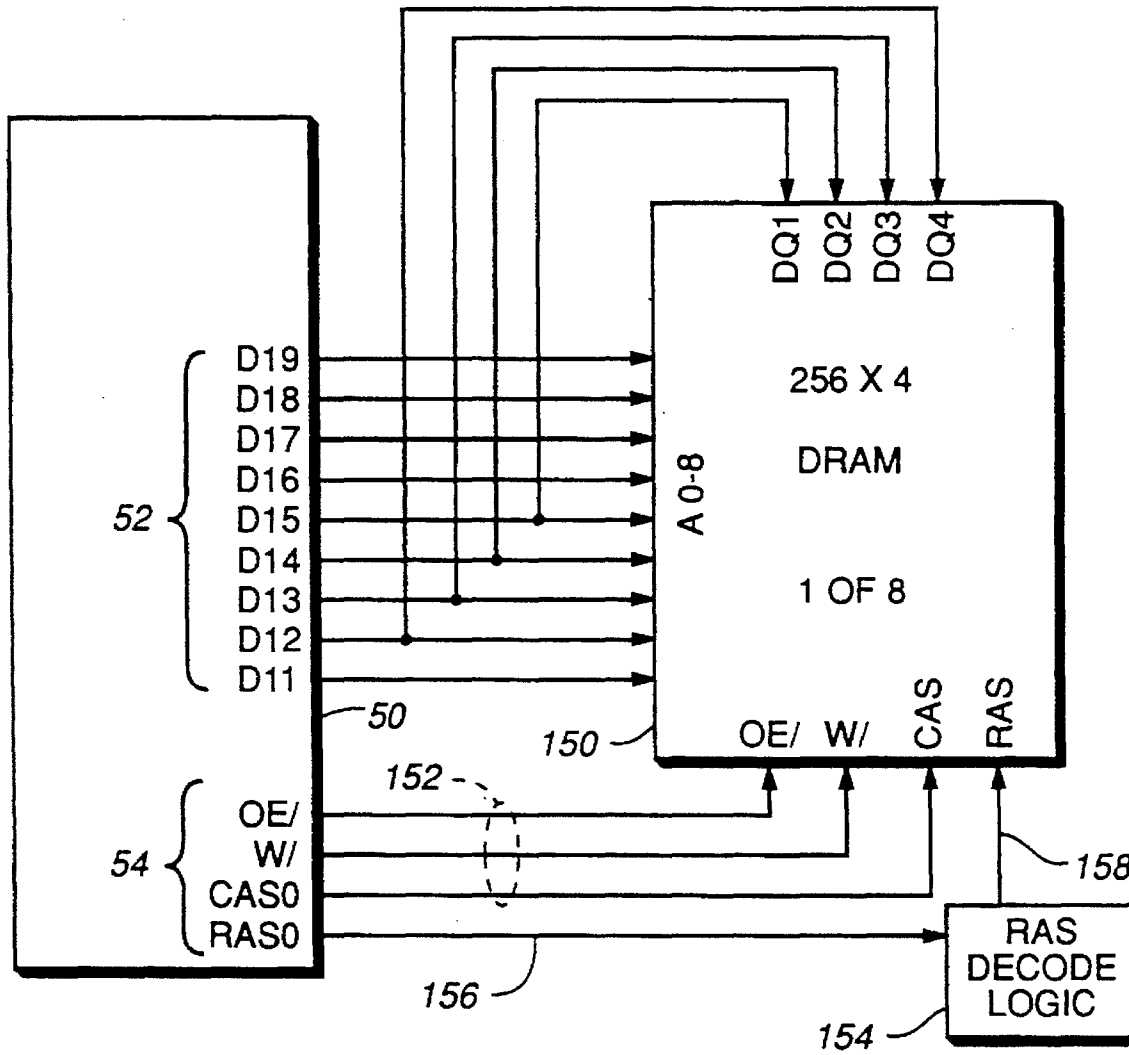

DQ1
DQ2
DQ3
DQ4
256 X 4
DRAM
1 OF 8
A 0-8
OE/
W/
CAS
RAS
D19
D18
D17
D16
D15
D14
D13
D12
D11
52
50
150
152
OE/
W/
CAS0
RAS0
54
158
156
RAS
DECODE
LOGIC
154

FIG._3

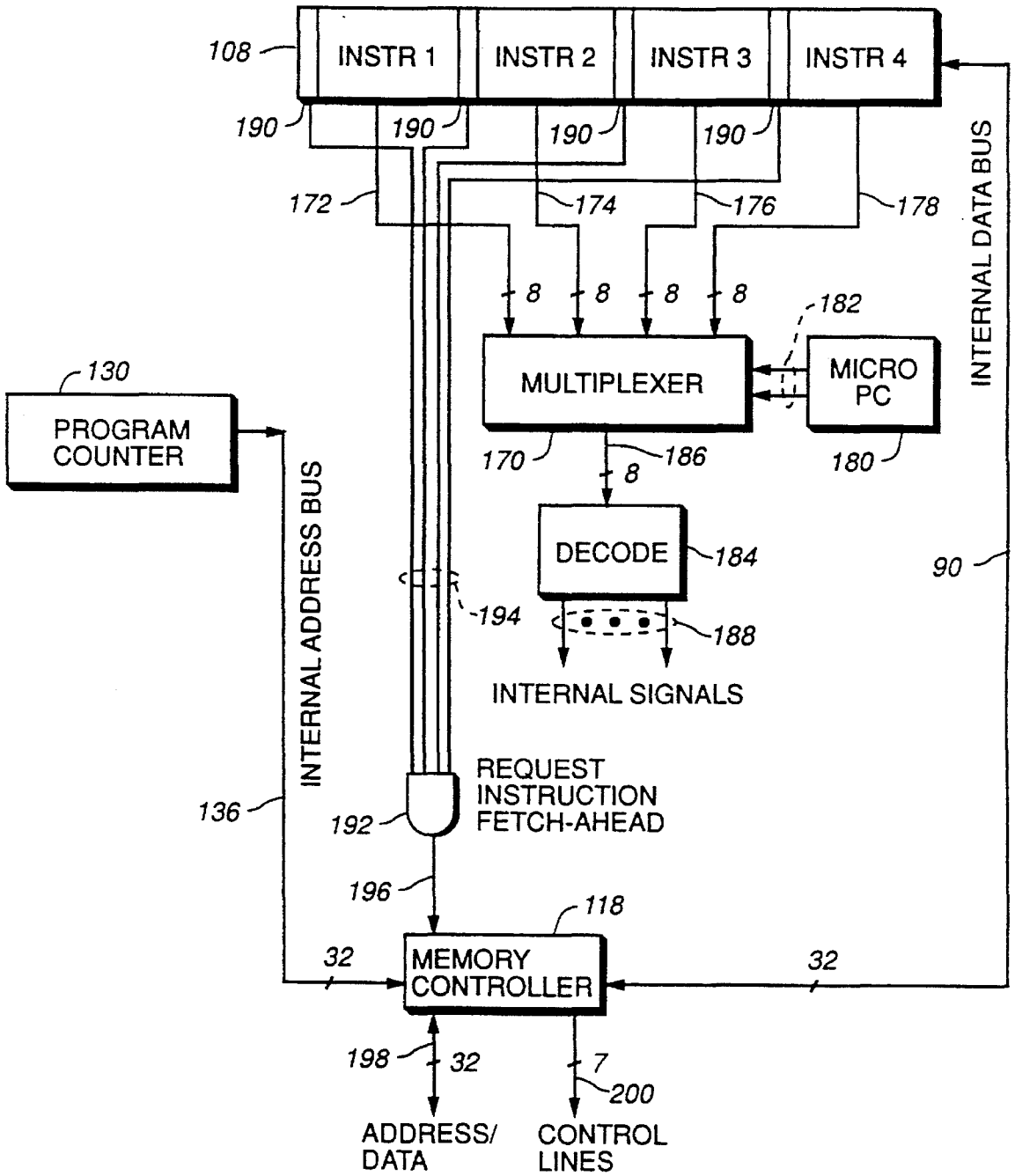

108
INSTR 1
INSTR 2
INSTR 3
INSTR 4
190
190
190
190
172
174
176
178
8
8
8
8
182
MULTIPLEXER
MICRO PC
180
170
186
8
DECODE
184
194
188
INTERNAL SIGNALS
INTERNAL DATA BUS
90
130
PROGRAM COUNTER
INTERNAL ADDRESS BUS
136
192
REQUEST INSTRUCTION FETCH-AHEAD
196
118
32
MEMORY CONTROLLER
32
198
32
7
200
ADDRESS/ DATA
CONTROL LINES

FIG._4

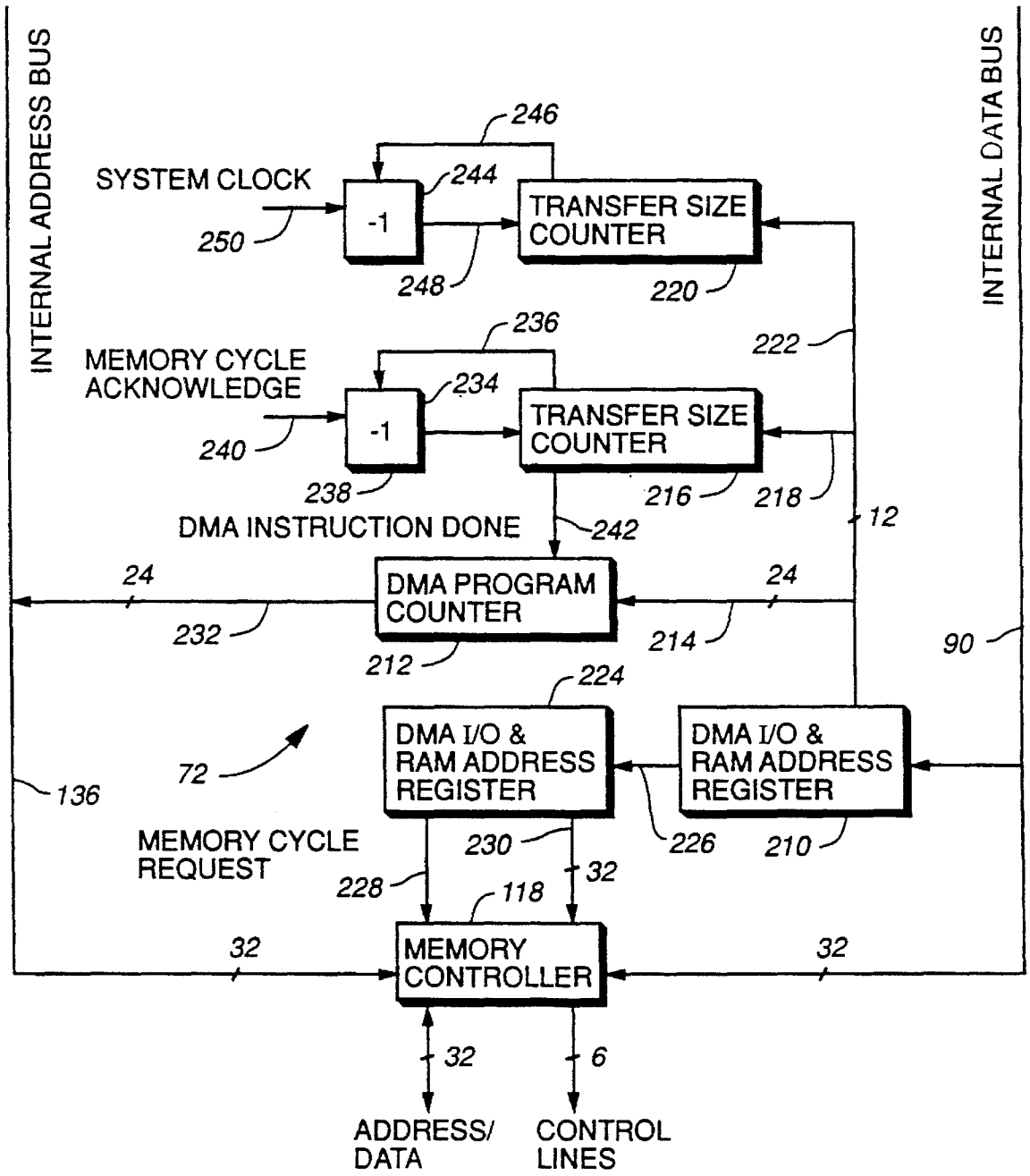

INTERNAL ADDRESS BUS
INTERNAL DATA BUS
SYSTEM CLOCK
250
-1
246
244
248
TRANSFER SIZE COUNTER
220
MEMORY CYCLE ACKNOWLEDGE
240
-1
236
234
238
TRANSFER SIZE COUNTER
216
218
222
12
DMA INSTRUCTION DONE
242
24
232
DMA PROGRAM COUNTER
212
24
214
90
224
72
136
DMA I/O & RAM ADDRESS REGISTER
DMA I/O & RAM ADDRESS REGISTER
226
210
MEMORY CYCLE REQUEST
230
228
32
118
32
MEMORY CONTROLLER
32
32
6
ADDRESS/ DATA
CONTROL LINES

FIG._5

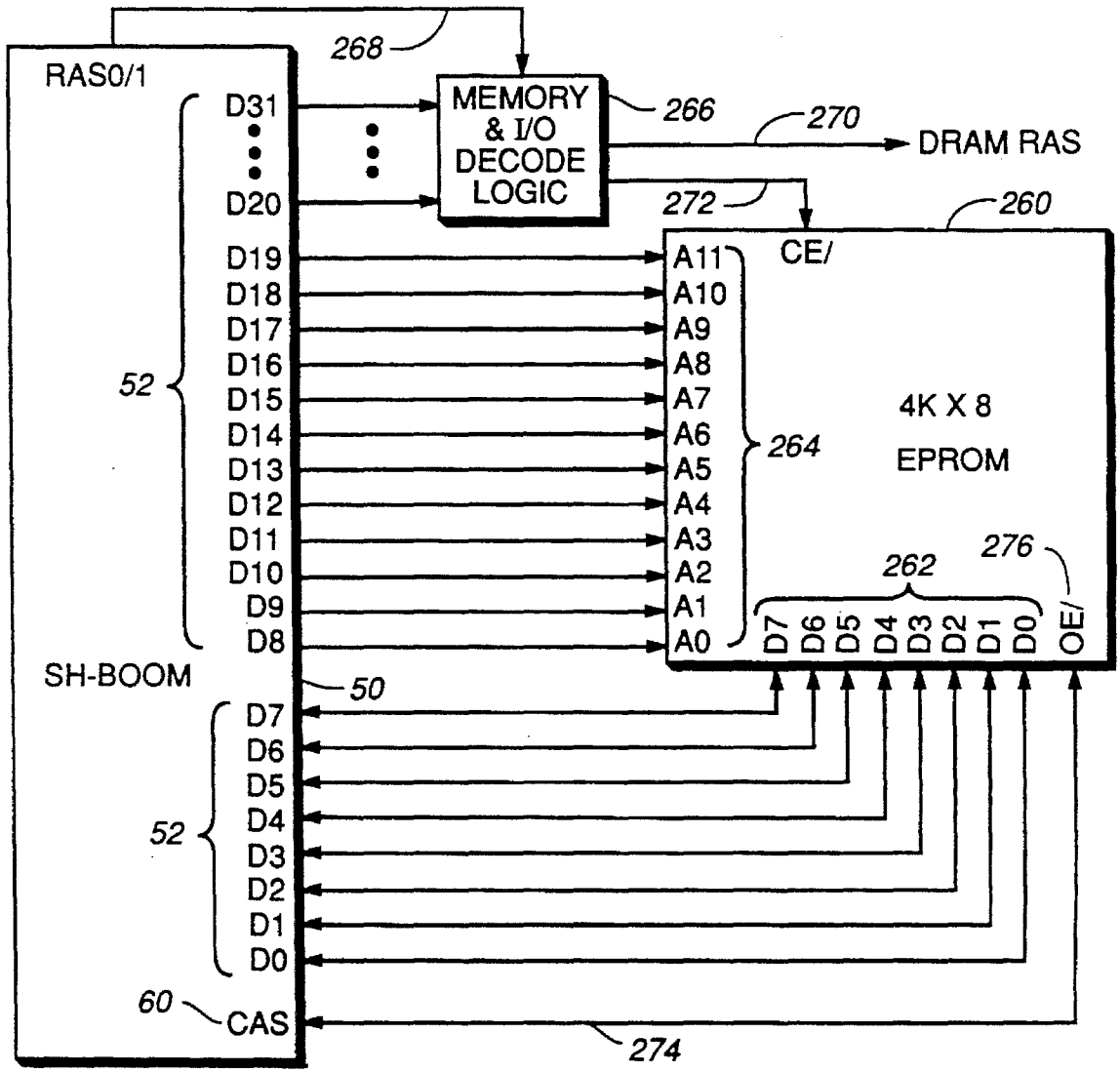

268
RAS0/1
D31
D20
MEMORY & I/O DECODE LOGIC
266
270
DRAM RAS
272
260
CE/
D19
D18
D17
D16
D15
D14
D13
D12
D11
D10
D9
D8
52
A11
A10
A9
A8
A7
A6
A5
A4
A3
A2
A1
A0
264
4K X 8
EPROM
276
262
D7
D6
D5
D4
D3
D2
D1
D0
OE/
SH-BOOM
50
52
60
CAS
274

FIG._6

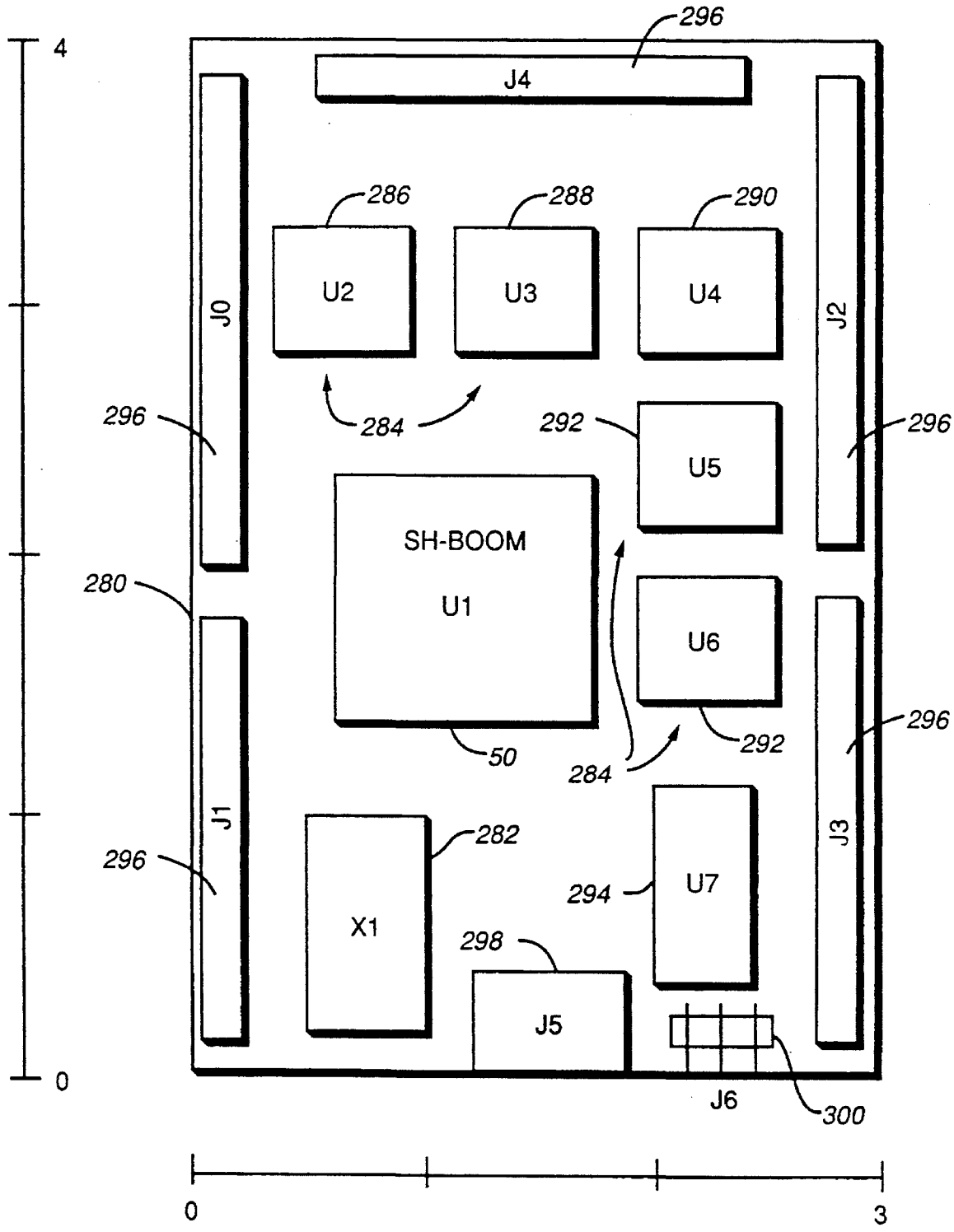

296
J4
286
288
290
U2
U3
U4
J0
J2
284
296
292
296
U5
SH-BOOM
U1
280
U6
296
50
284
292
J1
J3
296
282
294
U7
X1
298
J5
J6
300
4
0
0
3

FIG._7

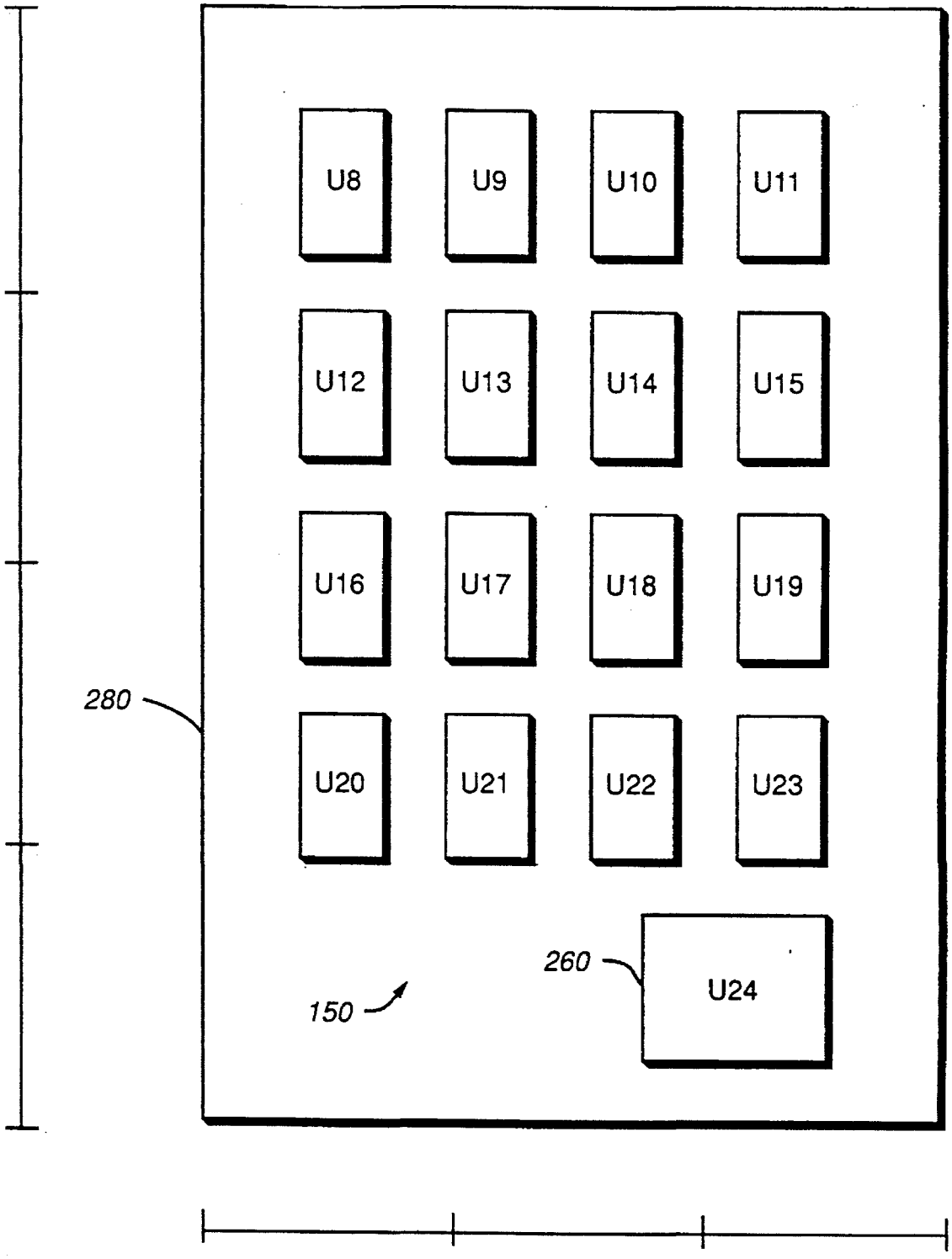

U8
U9
U10
U11
U12
U13
U14
U15
U16
U17
U18
U19
280
U20
U21
U22
U23
260
U24
150

FIG._8

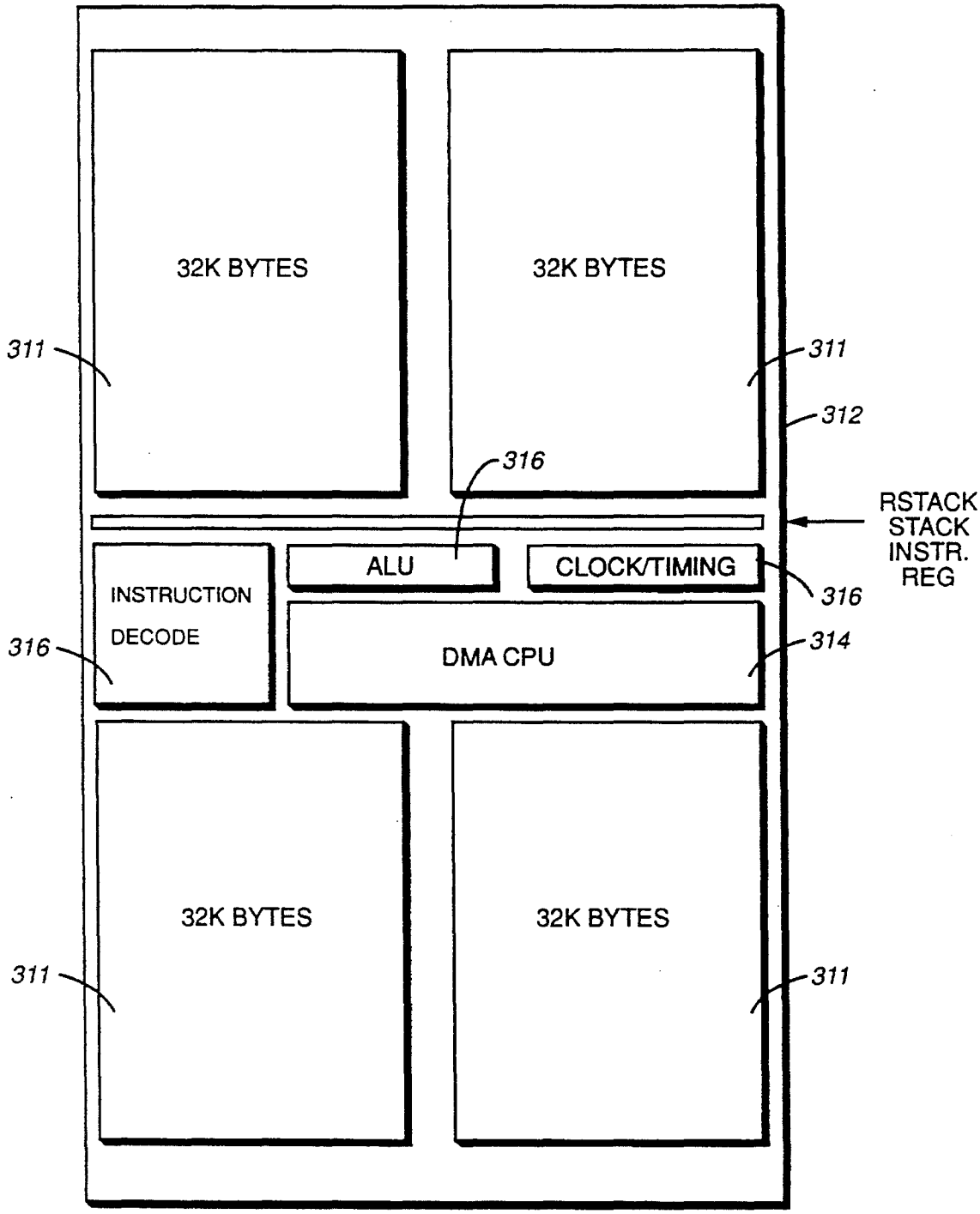

32K BYTES
32K BYTES
311
311
312
316
RSTACK
STACK
INSTR.
REG
ALU
CLOCK/TIMING
316
INSTRUCTION
DECODE
316
DMA CPU
314
32K BYTES
32K BYTES
311
311

FIG._9

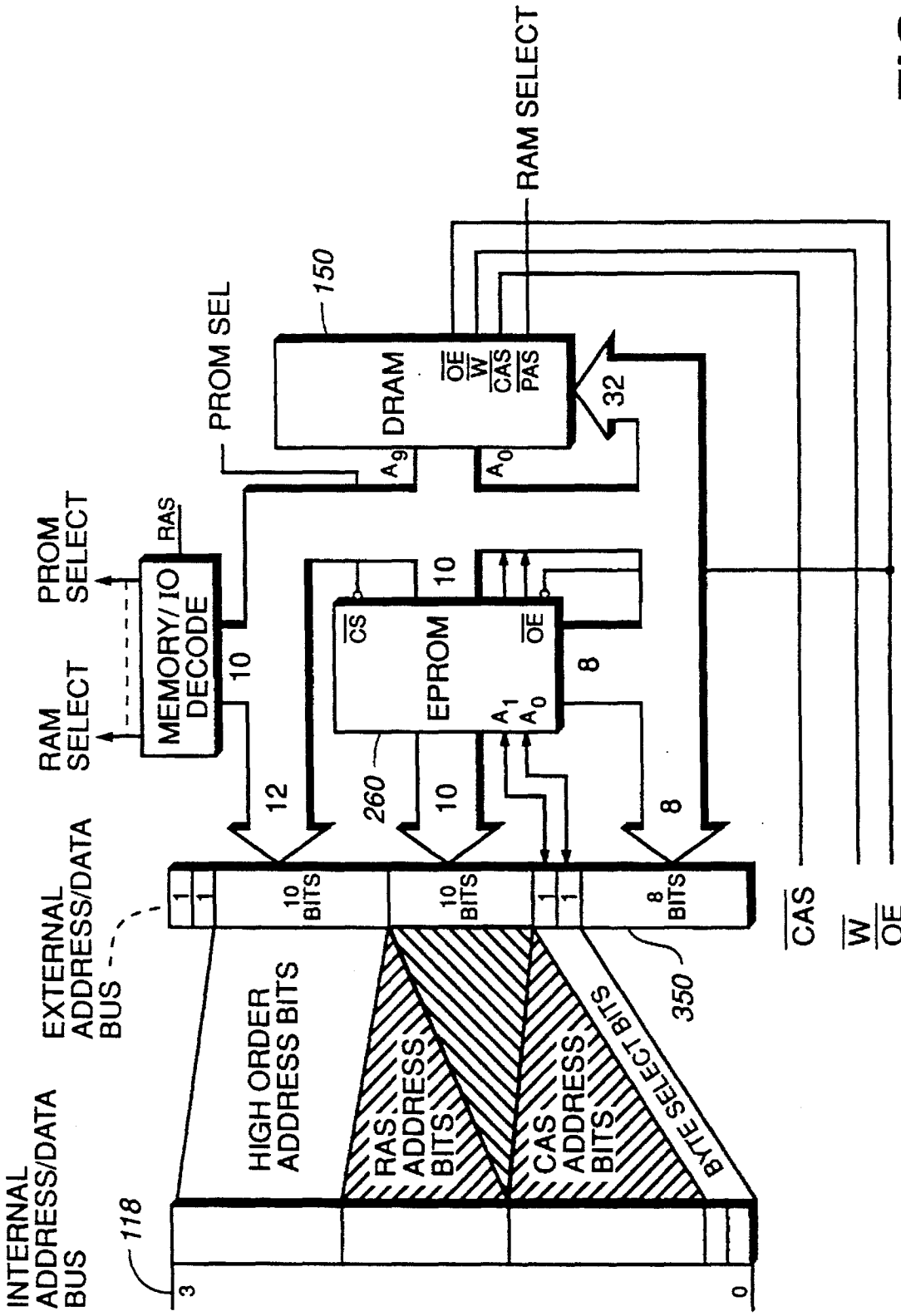
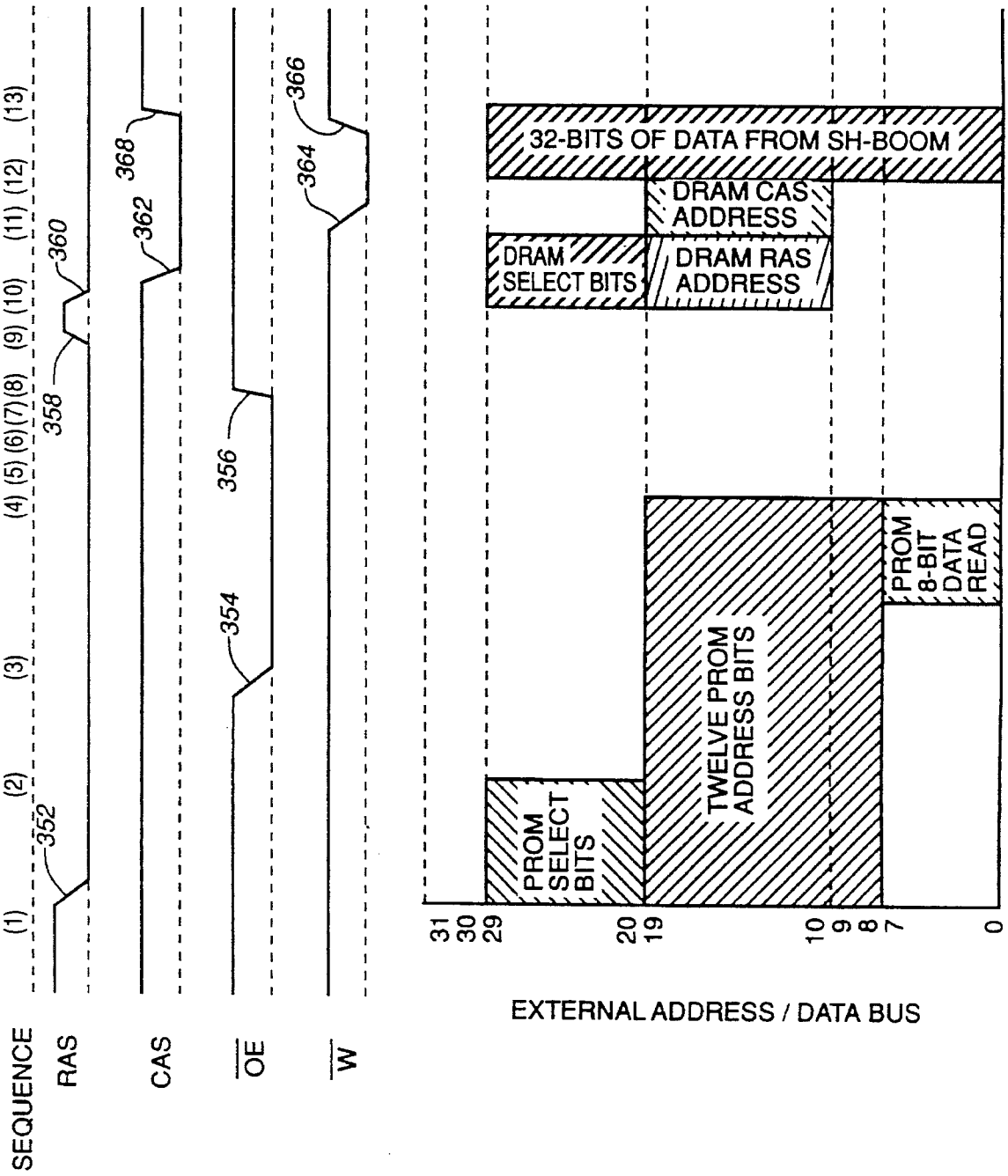

RAM SELECT
150
PROM SEL
DRAM
OE
W
CAS
RAS
32
A9
A0
PROM SELECT
RAS
MEMORY/ IO DECODE
10
RAM SELECT
CS
EPROM
OE
A1
A0
10
8
12
260
10
8
EXTERNAL ADDRESS/DATA BUS
1
1
10 BITS
10 BITS
1
1
8 BITS
CAS
W
OE
350
HIGH ORDER ADDRESS BITS
RAS ADDRESS BITS
CAS ADDRESS BITS
BYTE SELECT BITS
INTERNAL ADDRESS/DATA BUS
118
3
0

FIG._10

FIG._11


SEQUENCE
RAS
CAS
OE
W
(1)
(2)
(3)
(4) (5) (6) (7) (8)
(9) (10)
(11) (12)
(13)
352
354
356
358
360
362
364
366
368
32-BITS OF DATA FROM SH-BOOM
DRAM CAS ADDRESS
DRAM SELECT BITS
DRAM RAS ADDRESS
PROM SELECT BITS
TWELVE PROM ADDRESS BITS
PROM 8-BIT DATA READ
31 30 29 20 19 10 9 8 7 0
EXTERNAL ADDRESS / DATA BUS

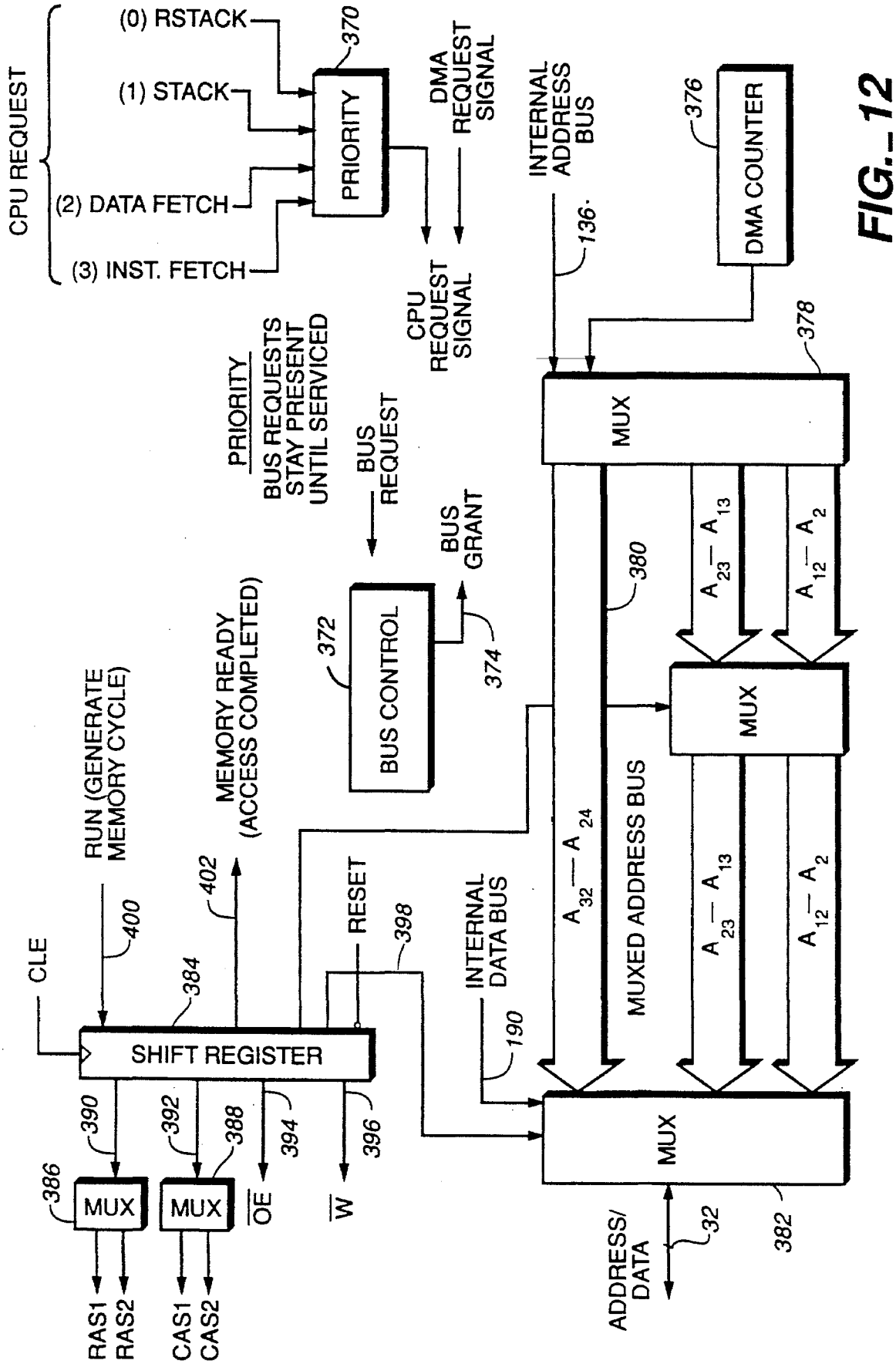

CPU REQUEST
(0) RSTACK
(1) STACK
(2) DATA FETCH
(3) INST. FETCH
370
PRIORITY
DMA REQUEST SIGNAL
CPU REQUEST SIGNAL
INTERNAL ADDRESS BUS
136
376
DMA COUNTER
PRIORITY
BUS REQUESTS STAY PRESENT UNTIL SERVICED
BUS REQUEST
BUS GRANT
372
BUS CONTROL
374
378
MUX
380
A23 — A13
A12 — A2
MUX
A32 — A24
MUXED ADDRESS BUS
MEMORY READY (ACCESS COMPLETED)
RUN (GENERATE MEMORY CYCLE)
400
402
RESET
398
CLE
384
INTERNAL DATA BUS
190
SHIFT REGISTER
390
392
388
394
396
386
MUX
MUX
OE
W
MUX
382
32
ADDRESS/ DATA
RAS1
RAS2
CAS1
CAS2

FIG._12

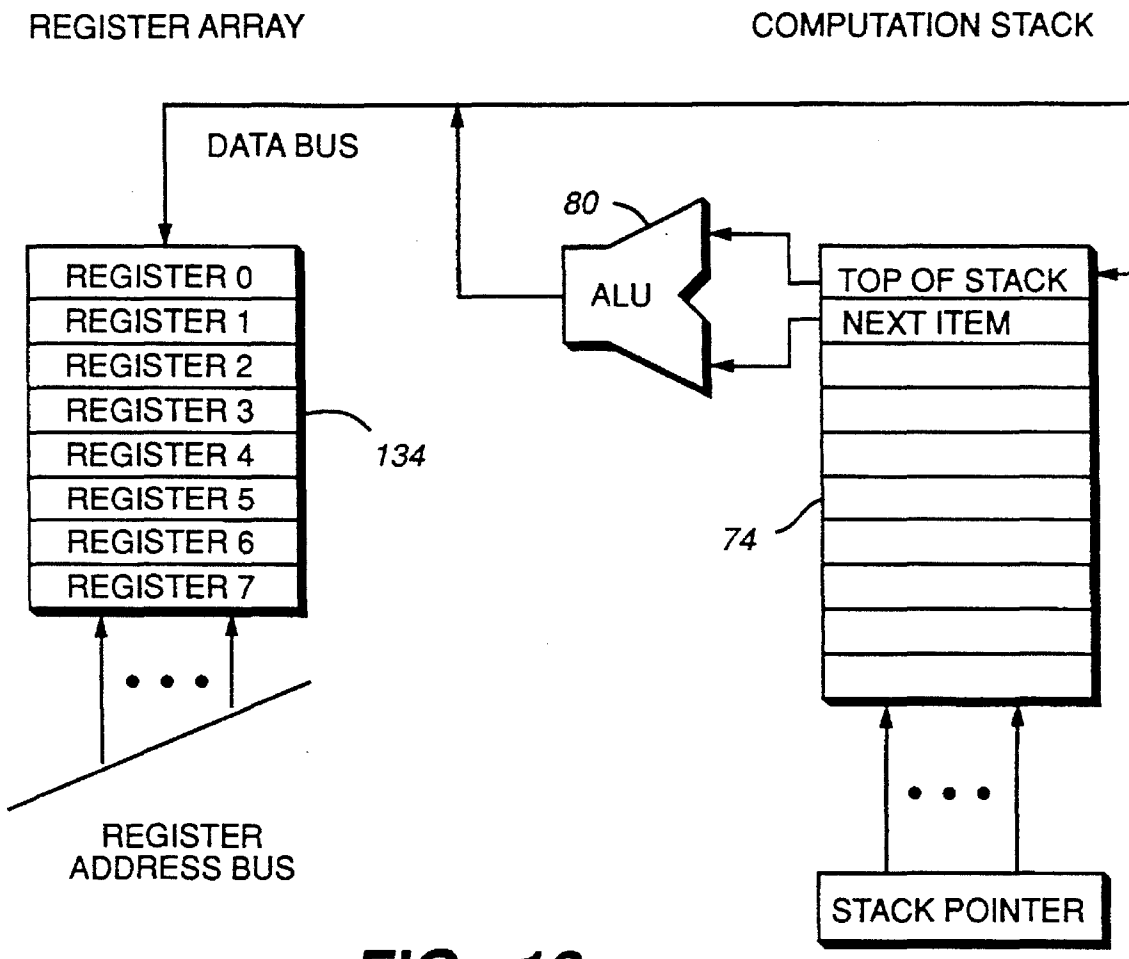

REGISTER ARRAY
COMPUTATION STACK
DATA BUS
80
ALU
REGISTER 0
REGISTER 1
REGISTER 2
REGISTER 3
REGISTER 4
REGISTER 5
REGISTER 6
REGISTER 7
134
TOP OF STACK
NEXT ITEM
74
REGISTER
ADDRESS BUS
STACK POINTER

FIG._13

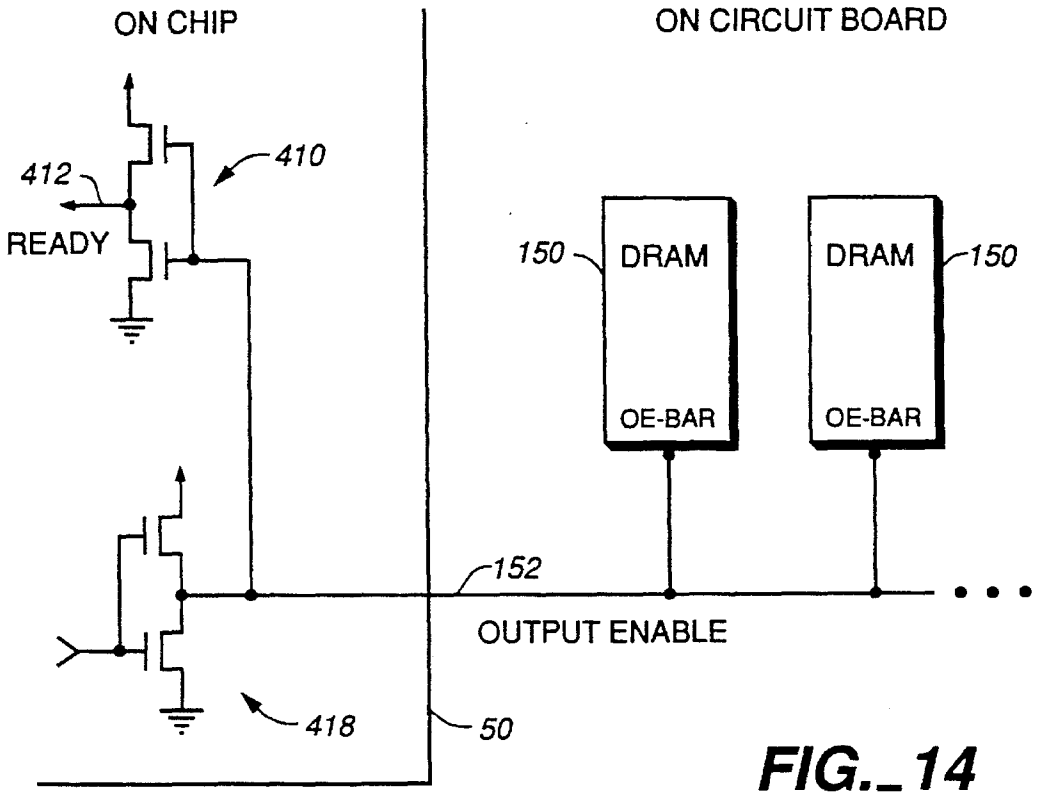

ON CHIP
ON CIRCUIT BOARD
412
410
READY
150
DRAM
OE-BAR
DRAM
OE-BAR
150
152
OUTPUT ENABLE
418
50

FIG._14

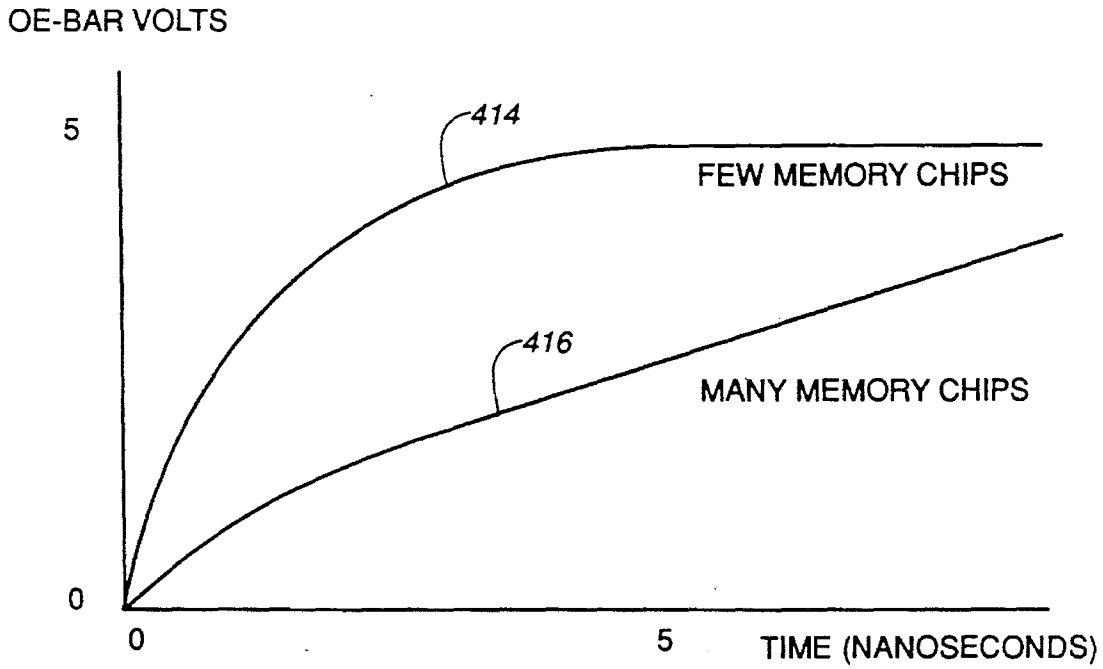

OE-BAR VOLTS
5
0
414
FEW MEMORY CHIPS
416
MANY MEMORY CHIPS
0
5
TIME (NANOSECONDS)

FIG._15

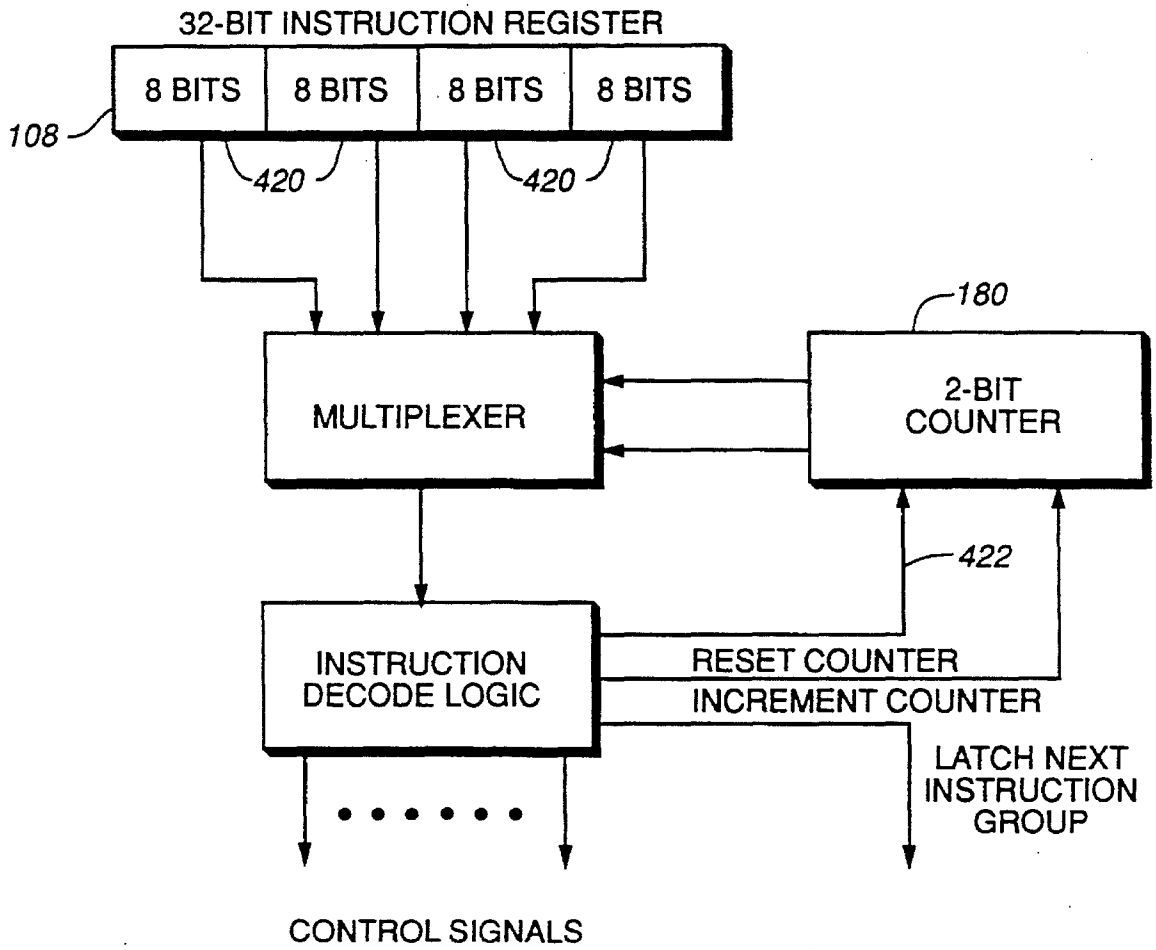

32-BIT INSTRUCTION REGISTER
8 BITS
8 BITS
8 BITS
8 BITS
108
420
420
180
MULTIPLEXER
2-BIT COUNTER
422
INSTRUCTION DECODE LOGIC
RESET COUNTER
INCREMENT COUNTER
LATCH NEXT INSTRUCTION GROUP
CONTROL SIGNALS

FIG._16

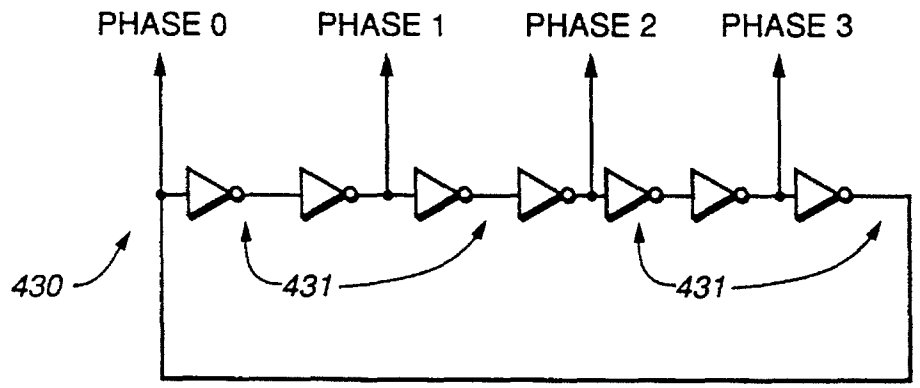

PHASE 0
PHASE 1
PHASE 2
PHASE 3
430
431
431

FIG._18

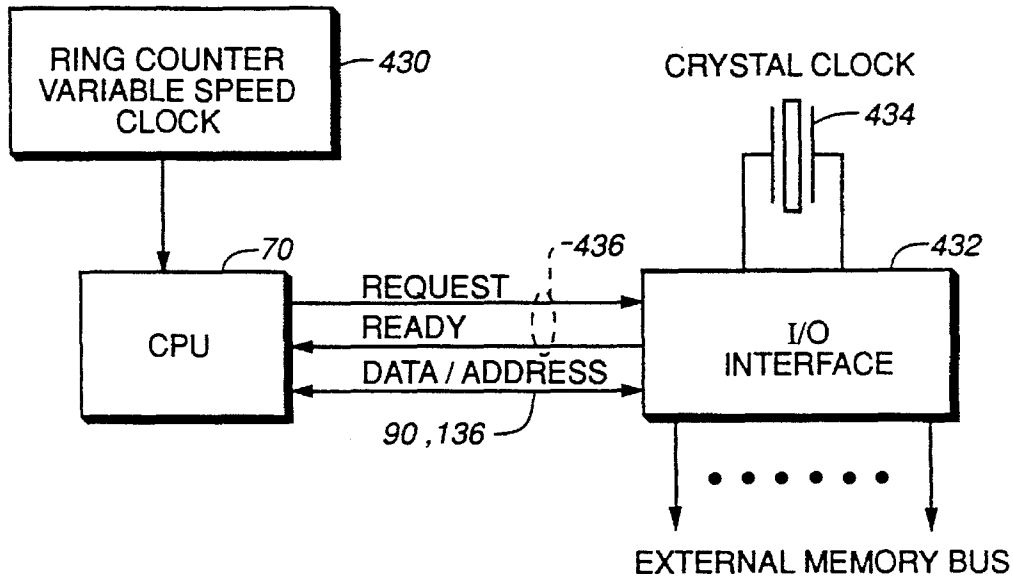

RING COUNTER VARIABLE SPEED CLOCK
430
CRYSTAL CLOCK
434
70
436
432
CPU
REQUEST
READY
DATA / ADDRESS
I/O INTERFACE
90 , 136
EXTERNAL MEMORY BUS

FIG._17

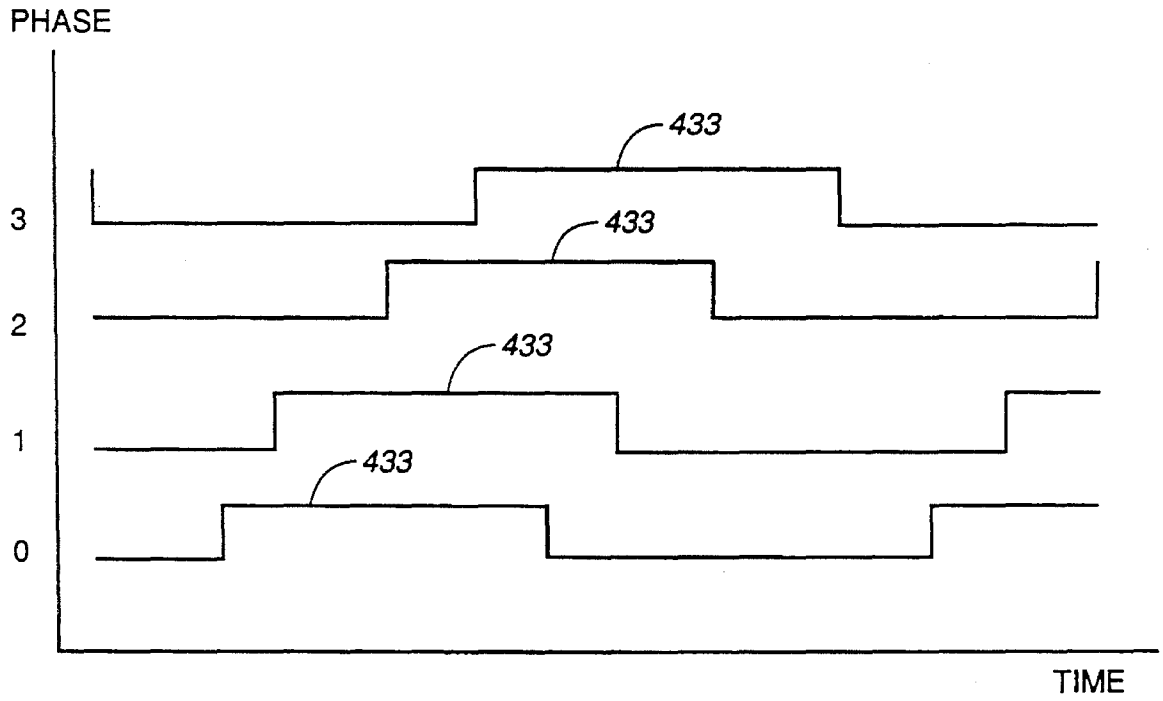

PHASE
433
433
433
433
3
2
1
0
TIME

FIG._19

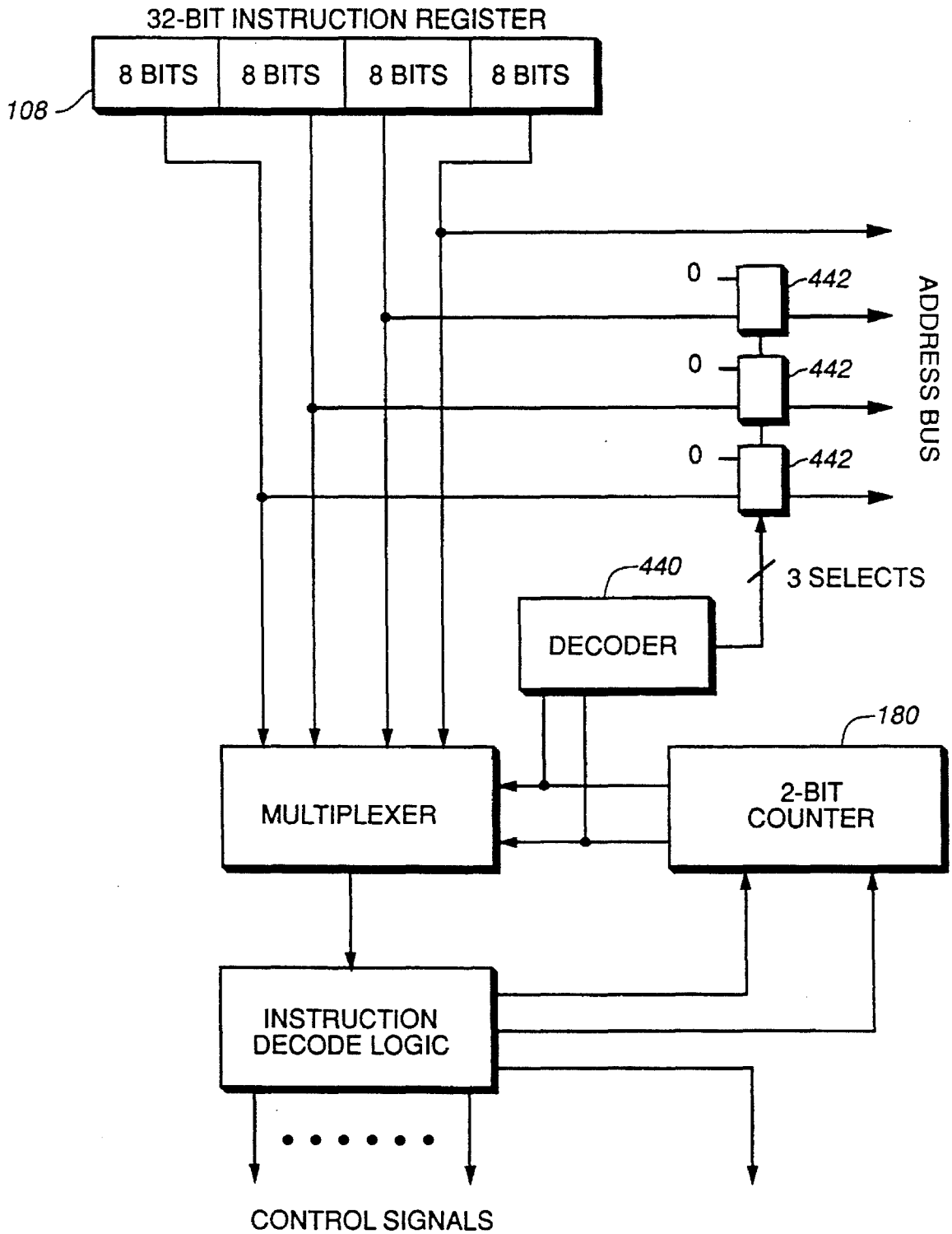

32-BIT INSTRUCTION REGISTER
8 BITS
8 BITS
8 BITS
8 BITS
108
0
442
0
442
0
442
ADDRESS BUS
440
3 SELECTS
DECODER
180
MULTIPLEXER
2-BIT COUNTER
INSTRUCTION DECODE LOGIC
CONTROL SIGNALS

FIG._20

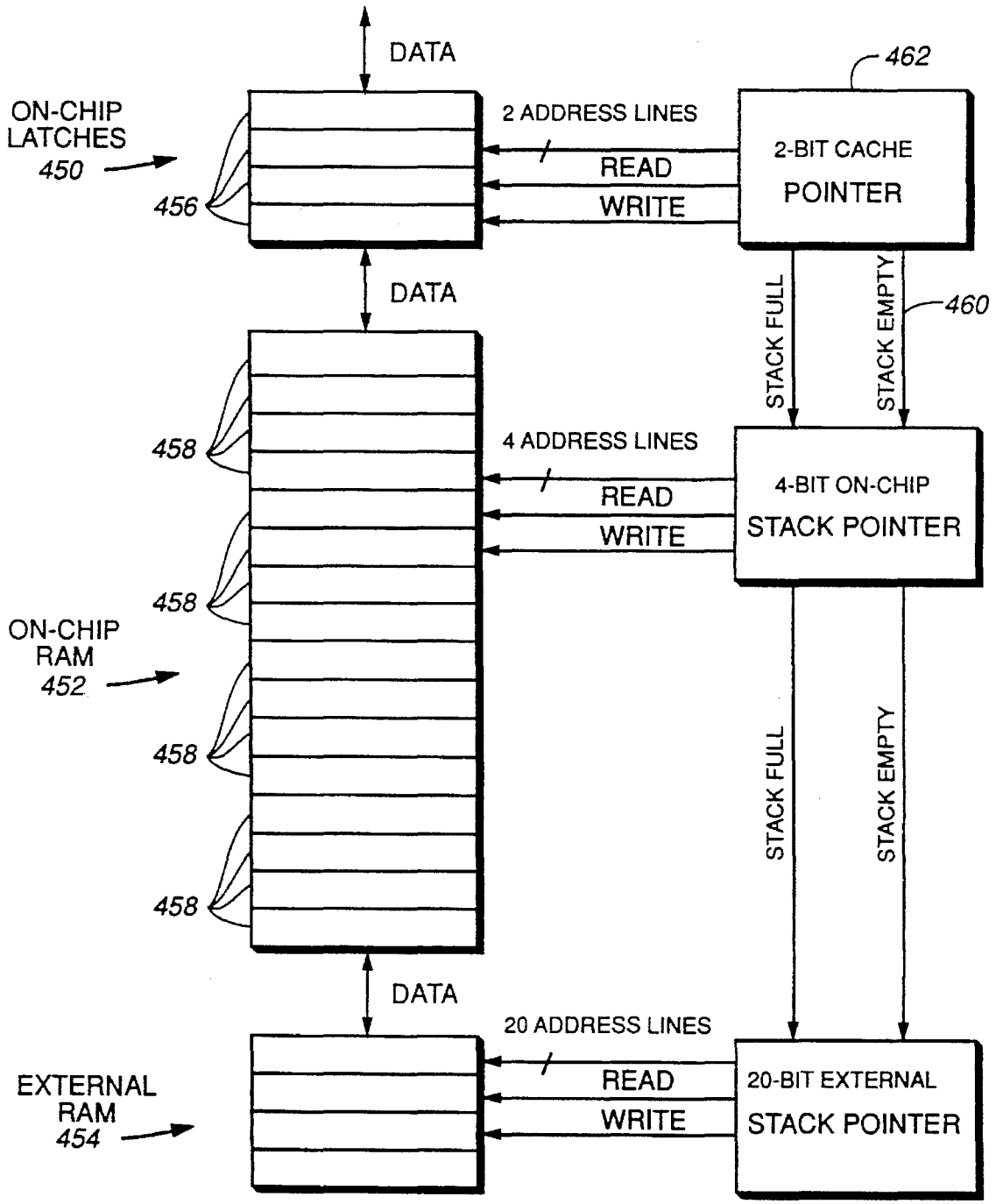

DATA
462
ON-CHIP LATCHES 450
2 ADDRESS LINES
READ
WRITE
2-BIT CACHE POINTER
456
DATA
STACK FULL
STACK EMPTY
460
458
4 ADDRESS LINES
READ
WRITE
4-BIT ON-CHIP STACK POINTER
458
ON-CHIP RAM 452
458
STACK FULL
STACK EMPTY
458
DATA
20 ADDRESS LINES
READ
WRITE
20-BIT EXTERNAL STACK POINTER
EXTERNAL RAM 454

FIG._21

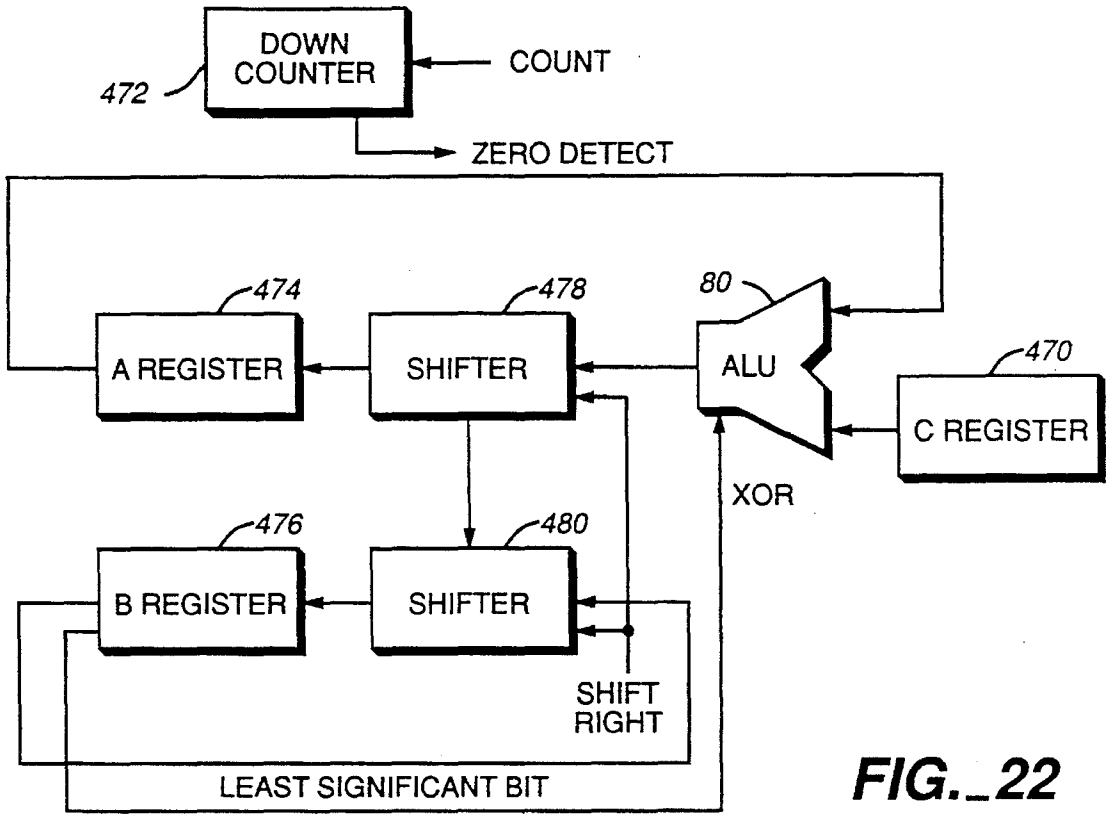

DOWN COUNTER
472
COUNT
ZERO DETECT
474
A REGISTER
478
SHIFTER
80
ALU
470
C REGISTER
XOR
476
B REGISTER
480
SHIFTER
SHIFT RIGHT
LEAST SIGNIFICANT BIT

FIG._22

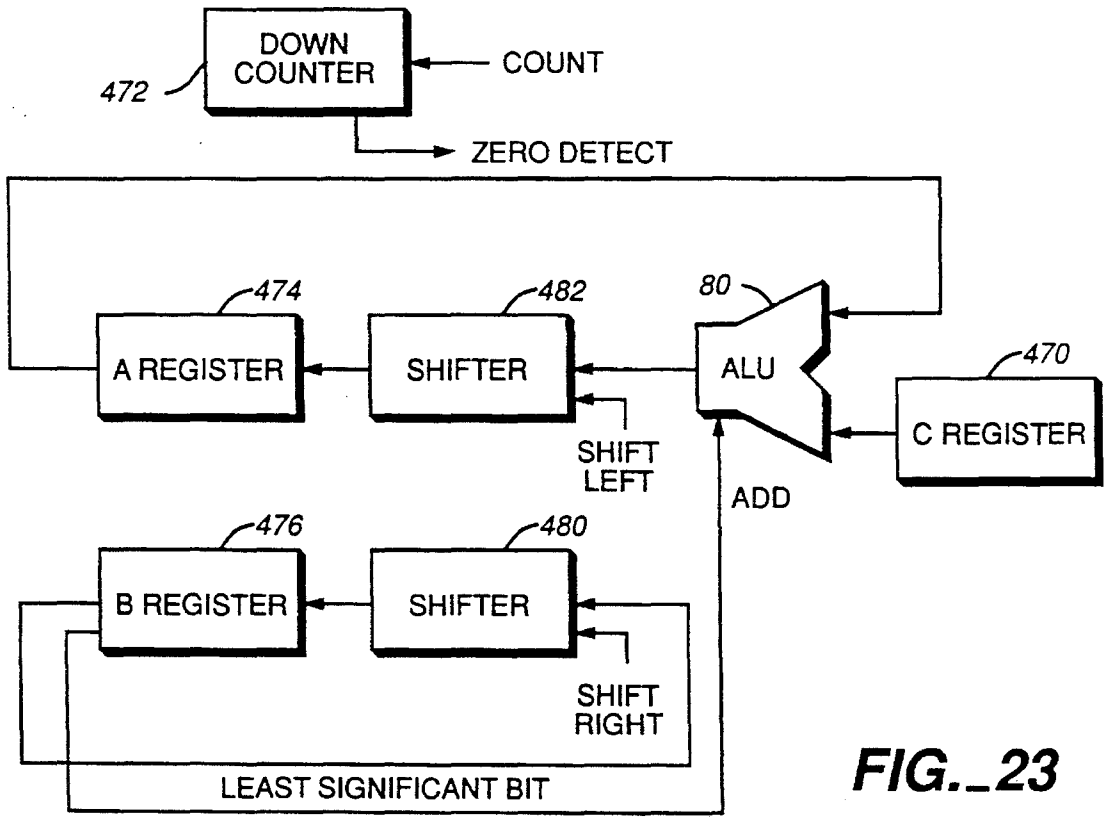

DOWN COUNTER
472
COUNT
ZERO DETECT
474
A REGISTER
482
SHIFTER
80
ALU
470
C REGISTER
SHIFT LEFT
ADD
476
B REGISTER
480
SHIFTER
SHIFT RIGHT
LEAST SIGNIFICANT BIT

FIG._23

HIGH PERFORMANCE, LOW COST MICROPROCESSOR

CROSS REFERENCE TO RELATED APPLICATIONS

This application is a division of U.S. application Ser. No. 07/389,334, filed Aug. 3, 1989, now U.S. Pat. No. 5,440,749.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to a simplified, reduced instruction set computer (RISC) microprocessor. More particularly, it relates to such a microprocessor which is capable of performance levels of, for example, 20 million instructions per second (MIPS) at a price of, for example, 20 dollars.

2. Description of the Prior Art

since the invention of the microprocessor, improvements in its design have taken two different approaches. In the first approach, a brute force gain in performance has been achieved through the provision of greater numbers of faster transistors in the microprocessor integrated circuit and an instruction set of increased complexity. This approach is exemplified by the Motorola 68000 and Intel 80X86 microprocessor families. The trend in this approach is to larger die sizes and packages, with hundreds of pinouts.

More recently, it has been perceived that performance gains can be achieved through comparative simplicity, both in the microprocessor integrated circuit itself and in its instruction set. This second approach provides RISC microprocessors, and is exemplified by the Sun SPARC and the Intel 8960 microprocessors. However, even with this approach as conventionally practiced, the packages for the microprocessor are large, in order to accommodate the large number of pinouts that continue to be employed. A need therefore remains for further simplification of high performance microprocessors.

With conventional high performance microprocessors, fast static memories are required for direct connection to the microprocessors in order to allow memory accesses that are fast enough to keep up with the microprocessors. Slower dynamic random access memories (DRAMs) are used with such microprocessors only in a hierarchical memory arrangement, with the static memories acting as a buffer between the microprocessors and the DRAMs. The necessity to use static memories increases cost of the resulting systems.

Conventional microprocessors provide direct memory accesses (DMA) for system peripheral units through DMA controllers, which may be located on the microprocessor integrated circuit, or provided separately. Such DMA controllers can provide routine handling of DMA requests and responses, but some processing by the main central processing unit (CPU) of the microprocessor is required.

SUMMARY OF THE INVENTION

Accordingly, it is an object of this invention to provide a microprocessor with a reduced pin count and cost compared to conventional microprocessors.

It is another object of the invention to provide a high performance microprocessor that can be directly connected to DRAMs without sacrificing microprocessor speed.

It is a further object of the invention to provide a high performance microprocessor in which DMA does not require use of the main CPU during DMA requests and responses and which provides very rapid DMA response with predictable response times.

The attainment of these and related objects may be achieved through use of the novel high performance, low cost microprocessor herein disclosed. In accordance with one aspect of the invention, a microprocessor system in accordance with this invention has a central processing unit, a dynamic random access memory and a bus connecting the central processing unit to the dynamic random access memory. There is a multiplexing means on the bus between the central processing unit and the dynamic random access memory. The multiplexing means is connected and configured to provide row addresses, column addresses and data on the bus.

In accordance with another aspect of the invention, the microprocessor system has a means connected to the bus for fetching instructions for the central processing unit on the bus. The means for fetching instructions is configured to fetch multiple sequential instructions in a single memory cycle. In a variation of this aspect of the invention, a programmable read only memory containing instructions for the central processing unit is connected to the bus. The means for fetching instructions includes means for assembling a plurality of instructions from the programmable read only memory and storing the plurality of instructions in the dynamic random access memory.

In another aspect of the invention, the microprocessor system includes a central processing unit, a direct memory access processing unit and a memory connected by a bus. The direct memory access processing unit includes means for fetching instructions for the central processing unit and for fetching instructions for the direct memory access processing unit on the bus.

In a further aspect of the invention, the microprocessor system, including the memory, is contained in an integrated circuit. The memory is a dynamic random access memory, and the means for fetching multiple instructions includes a column latch for receiving the multiple instructions.

In still another aspect of the invention, the microprocessor system additionally includes an instruction register for the multiple instructions connected to the means for fetching instructions. A means is connected to the instruction register for supplying the multiple instructions in succession from the instruction register. A counter is connected to control the means for supplying the multiple instructions to supply the multiple instructions in succession. A means for decoding the multiple instructions is connected to receive the multiple instructions in succession from the means for supplying the multiple instructions. The counter is connected to said means for decoding to receive incrementing and reset control signals from the means for decoding. The means for decoding is configured to supply the reset control signal to the counter and to supply a control signal to the means for fetching instructions in response to a SKIP instruction in the multiple instructions. In a modification of this aspect of the invention, the microprocessor system additionally has a loop counter connected to receive a decrement control signal from the means for decoding. The means for decoding is configured to supply the reset control signal to the counter and the decrement control signal to the loop counter in response to a MICROLOOP instruction in the multiple instructions. In a further modification to this aspect of the invention, the means for decoding is configured to control

the counter in response to an instruction utilizing a variable width operand. A means is connected to the counter to select the variable width operand in response to the counter.

In a still further aspect of the invention, the microprocessor system includes an arithmetic logic unit. A first push down stack is connected to the arithmetic logic unit. The first push down stack includes means for storing a top item connected to a first input of the arithmetic logic unit and means for storing a next item connected to a second input of the arithmetic logic unit. The arithmetic logic unit has an output connected to the means for storing a top item. The means for storing a top item is connected to provide an input to a register file. The register file desirably is a second push down stack, and the means for storing a top item and the register file are bidirectionally connected.

In another aspect of the invention, a data processing system has a microprocessor including a sensing circuit and a driver circuit, a memory, and an output enable line connected between the memory, the sensing circuit and the driver circuit. The sensing circuit is configured to provide a ready signal when the output enable line reaches a predetermined electrical level, such as a voltage. The microprocessor is configured so that the driver circuit provides an enabling signal on the output enable line responsive to the ready signal.

In a further aspect of the invention, the microprocessor system has a ring counter variable speed system clock connected to the central processing unit. The central processing unit and the ring counter variable speed system clock are provided in a single integrated circuit. An input/output interface is connected to exchange coupling control signals, addresses and data with the input/output interface. A second clock independent of the ring counter variable speed system clock is connected to the input/output interface.

In yet another aspect of the invention, a push down stack is connected to the arithmetic logic unit. The push down stack includes means for storing a top item connected to a first input of the arithmetic logic unit and means for storing a next item connected to a second input of the arithmetic logic unit. The arithmetic logic unit has an output connected to the means for storing a top item. The push down stack has a first plurality of stack elements configured as latches and a second plurality of stack elements configured as a random access memory. The first and second plurality of stack elements and the central processing unit are provided in a single integrated circuit. A third plurality of stack elements is configured as a random access memory external to the single integrated circuit. In this aspect of the invention, desirably a first pointer is connected to the first plurality of stack elements, a second pointer connected to the second plurality of stack elements, and a third pointer is connected to the third plurality of stack elements. The central processing unit is connected to pop items from the first plurality of stack elements. The first stack pointer is connected to the second stack pointer to pop a first plurality of items from the second plurality of stack elements when the first plurality of stack elements are empty from successive pop operations by the central processing unit. The second stack pointer is connected to the third stack pointer to pop a second plurality of items from the third plurality of stack elements when the second plurality of stack elements are empty from successive pop operations by the central processing unit.

In another aspect of the invention, a first register is connected to supply a first input to the arithmetic logic unit. A first shifter is connected between an output of the arithmetic logic unit and the first register. A second register is

connected to receive a starting polynomial value. An output of the second register is connected to a second shifter. A least significant bit of the second register is connected to The arithmetic logic unit. A third register is connected to supply feedback terms of a polynomial to the arithmetic logic unit. A down counter, for counting down a number corresponding to digits of a polynomial to be generated, is connected to the arithmetic logic unit. The arithmetic logic unit is responsive to a polynomial instruction to carry out an exclusive OR of the contents of the first register with the contents of the third register if the least significant bit of the second register is a "ONE" and to pass the contents of the first register unaltered if the least significant bit of the second register is a "ZERO", until the down counter completes a count The polynomial to be generated results in said first register.

In still another aspect of the invention, a result register is connected to supply a first input to the arithmetic logic unit. A first, left shifting shifter is connected between an output of the arithmetic logic unit and the result register. A multiplier register is connected to receive a multiplier in bit reversed form. An output of the multiplier register is connected to a second, right shifting shifter. A least significant bit of the multiplier register is connected to the arithmetic logic unit. A third register is connected to supply a multiplicand to said arithmetic logic unit. A down counter, for counting down a number corresponding to one less than the number of digits of the multiplier, is connected to the arithmetic logic unit. The arithmetic logic unit is responsive to a multiply instruction to add the contents of the result register with the contents of the third register, when the least significant bit of the multiplier register is a "ONE" and to pass the contents of the result register unaltered, until the down counter completes a count. The product results in the result register.

The attainment of the foregoing and related objects, advantages and features of the invention should be more readily apparent to those skilled in the art, after review of the following more detailed description of the invention, taken together with the drawings, in which:

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is an external, plan view of an integrated circuit package incorporating a microprocessor in accordance with the invention.

FIG. **2** is a block diagram of a microprocessor in accordance with the invention.

FIG. **3** is a block diagram of a portion of a data processing system incorporating the microprocessor of FIGS. **1** and **2**.

FIG. **4** is a more detailed block diagram of a portion of the microprocessor shown in FIG. **2**.

FIG. **5** is a more detailed block diagram of another portion of the microprocessor shown in FIG. **2**.

FIG. **6** is a block diagram of another portion of the data processing system shown in part in FIG. **3** and incorporating the microprocessor of FIGS. **1–2** and **4–5**.

FIGS. **7** and **8** are layout diagrams for the data processing system shown in part in FIGS. **3** and **6**.

FIG. **9** is a layout diagram of a second embodiment of a microprocessor in accordance with the invention in a data processing system on a single integrated circuit.

FIG. **10** is a more detailed block diagram of a portion of the data processing system of FIGS. **7** and **8**.

FIG. **11** is a timing diagram useful for understanding operation of the system portion shown in FIG. **12**.

FIG. **12** is another more detailed block diagram of a further portion of the data processing system of FIGS. **7** and **8**.

FIG. **13** is a more detailed block diagram of a portion of the microprocessor shown in FIG. **2**.

FIG. **14** is a more detailed block and schematic diagram of a portion of the system shown in FIGS. **3** and **7–8**.

FIG. **15** is a graph useful for understanding operation of the system portion shown in FIG. **14**.

FIG. **16** is a more detailed block diagram showing part of the system portion shown in FIG. **4**.

FIG. **17** is a more detailed block diagram of a portion of the microprocessor shown in FIG. **2**.

FIG. **18** is a more detailed block diagram of part of the microprocessor portion shown in FIG. **17**.

FIG. **19** is a set of waveform diagrams useful for understanding operation of the part of the microprocessor portion shown in FIG. **18**.

FIG. **20** is a more detailed block diagram showing another part of the system portion shown in FIG. **4**.

FIG. **21** is a more detailed block diagram showing another part of the system portion shown in FIG. **4**.

FIGS. **22** and **23** are more detailed block diagrams showing another part of the system portion shown in FIG. **4**.

DETAILED DESCRIPTION OF THE INVENTION

OVERVIEW

The microprocessor of this invention is desirably implemented as a 32-bit microprocessor optimized for:

HIGH EXECUTION SPEED, and

LOW SYSTEM COST.

In this embodiment, the microprocessor can be thought of as 20 MIPS for 20 dollars. Important distinguishing features of the microprocessor are:

Uses low-cost commodity DYNAMIC RAMS to run 20 MIPS

4 instruction fetch per memory cycle

On-chip fast page-mode memory management

Runs fast without external cache

Requires few interfacing chips

Crams 32-bit CPU in 44 pin SOJ package

The instruction set is organized so that most operations can be specified with 8-bit instructions. Two positive products of this philosophy are:

Programs are smaller,

Programs can execute much faster.

The bottleneck in most computer systems is the memory bus. The bus is used to fetch instructions and fetch and store data. The ability to fetch four instructions in a single memory bus cycle significantly increases the bus availability to handle data.

Turning now to the drawings, more particularly to FIG. **1**, there is shown a packaged 32-bit microprocessor **50** in a 44-pin plastic leadless chip carrier, shown approximately 100 times its actual size of about 0.8 inch on a side. The fact that the microprocessor **50** is provided as a 44-pin package represents a substantial departure from typical microprocessor packages, which usually have about 200 input/output (I/O) pins. The microprocessor **50** is rated at 20 million instructions per second (MIPS). Address and data lines **52**, also labelled **D0–D31**, are shared for addresses and data without speed penalty as a result of the manner in which the microprocessor **50** operates, as will be explained below.

DYNAMIC RAM

In addition to the low cost 44-pin package, another unusual aspect of the high performance microprocessor **50** is that it operates directly with dynamic random access memories (DRAMs), as shown by row address strobe (RAS) and column address strobe (CAS) I/O pins **54**. The other I/O pins for the microprocessor **50** include V_{DD} pins **56**, V_{SS} pins **58**, output enable pin **60**, write pin **62**, clock pin **64** and reset pin **66**.

All high speed computers require high speed and expensive memory to keep up. The highest speed static RAM memories cost as much as ten times as much as slower dynamic RAMs. This microprocessor has been optimized to use low-cost dynamic RAM in high-speed page-mode. Page-mode dynamic RAMs offer static RAM performance without the cost penalty. For example, low-cost 85 nsec. dynamic RAMs access at 25 nsec when operated in fast page-mode. Integrated fast page-mode control on the microprocessor chip simplifies system interfacing and results in a faster system.

Details of the microprocessor **50** are shown in FIG. **2**. The microprocessor **50** includes a main central processing unit (CPU) **70** and a separate direct memory access (DMA) CPU **72** in a single integrated circuit making up the microprocessor **50**. The main CPU **70** has a first 16 deep push down stack **74**, which has a top item register **76** and a next item register **78**, respectively connected to provide inputs to an arithmetic logic unit (ALU) **80** by lines **82** and **84**. An output of the ALU **80** is connected to the top item register **76** by line **86**. The output of the top item register at **82** is also connected by line **88** to an internal data bus **90**.

A loop counter **92** is connected to a decrementer **94** by lines **96** and **98**. The loop counter **92** is bidirectionally connected to the internal data bus **90** by line **100**. Stack pointer **102**, return stack pointer **104**, mode register **106** and instruction register **108** are also connected to the internal data bus **90** by lines **110**, **112**, **114** and **116**, respectively. The internal data bus **90** is connected to memory controller **118** and to gate **120**. The gate **120** provides inputs on lines **122**, **124**, and **126** to X register **128**, program counter **130** and Y register **132** of return push down stack **134**. The X register **128**, program counter **130** and Y register **132** provide outputs to internal address bus **136** on lines **138**, **140** and **142**. The internal address bus provides inputs to the memory controller **118** and to an incrementer **144**. The incrementer **144** provides inputs to the X register, program counter and Y register via lines **146**, **122**, **124** and **126**. The DMA CPU **72** provides inputs to the memory controller **118** on line **148**. The memory controller **118** is connected to a RAM (not shown) by address/data bus **150** and control lines **152**.

FIG. **2** shows that the microprocessor **50** has a simple architecture. Prior art RISC microprocessors are substantially more complex in design. For example, the SPARC RISC microprocessor has three times the gates of the microprocessor **50**, and the Intel 8960 RISC microprocessor has 20 times the gates of the microprocessor **50**. The speed of this microprocessor is in substantial part due to this simplicity. The architecture incorporates push down stacks and register write to achieve this simplicity.

The microprocessor **50** incorporates an I/O that has been tuned to make heavy use of resources provided on the integrated circuit chip. On chip latches allow use of the same I/O circuits to handle three different things: column addressing, row addressing and data, with a slight to non-existent speed penalty. This triple bus multiplexing results in fewer buffers to expand, fewer interconnection lines, fewer I/O pins and fewer internal buffers.

The provision of on-chip DRAM control gives a performance equal to that obtained with the use of static RAMs. As a result, memory is provided at ¼ the system cost of static RAM used in most RISC systems.

The microprocessor **50** fetches 4 instructions per memory cycle; the instructions are in an 8-bit format, and this is a 32-bit microprocessor. System speed is therefore 4 times the memory bus bandwidth. This ability enables the microprocessor to break the Von Neumann bottleneck of the speed of getting the next instruction. This mode of operation is possible because of the use of a push down stack and register array. The push down stack allows the use of implied addresses, rather than the prior art technique of explicit addresses for two sources and a destination.

Most instructions execute in 20 nanoseconds in the microprocessor **50**. The microprocessor can therefore execute instructions at 50 peak MIPS without pipeline delays. This is a function of the small number of gates in the microprocessor **50** and the high degree of parallelism in the architecture of the microprocessor.

FIG. **3** shows how column and row addresses are multiplexed on lines D8–D14 of the microprocessor **50** for addressing DRAM **150** from I/O pins **52**. The DRAM **150** is one of eight, but only one DRAM **150** has been shown for clarity. As shown, the lines D11–D18 are respectively connected to row address inputs A0–A8 of the DRAM **150**. Additionally, lines D12–D15 are connected to the data inputs DQ1–DQ4 of the DRAM **150**. The output enable, write and column address strobe pins **54** are respectively connected to the output enable, write and column address strobe inputs of the DRAM **150** by lines **152**. The row address strobe pin **54** is connected through row address strobe decode logic **154** to the row address strobe input of the DRAM **150** by lines **156** and **158**.

D0–D7 pins **52** (FIG. **1**) are idle when the microprocessor **50** is outputting multiplexed row and column addresses on D11–D18 pins **52**. The D0–D7 pins **52** can therefore simultaneously be used for I/O when right justified I/O is desired. Simultaneous addressing and I/O can therefore be carried out.

FIG. **4** shows how the microprocessor **50** is able to achieve performance equal to the use of static RAMS with DRAMs through multiple instruction fetch in a single clock cycle and instruction fetch-ahead. Instruction register **108** receives four 8-bit byte instruction words **1–4** on 32-bit internal data bus **90**. The four instruction byte **1–4** locations of the instruction register **108** are connected to multiplexer **170** by busses **172**, **174**, **176** and **178**, respectively. A microprogram counter **180** is connected to the multiplexer **170** by lines **182**. The multiplexer **170** is connected to decoder **184** by bus **186**. The decoder **184** provides internal signals to the rest of the microprocessor **50** on lines **188**.

Most significant bits **190** of each instruction byte **1–4** location are connected to a 4-input decoder **192** by lines **194**. The output of decoder **192** is connected to memory controller **118** by line **196**. Program counter **130** is connected to memory controller **118** by internal address bus **136**, and the instruction register **108** is connected to the memory controller **118** by the internal data bus **90**. Address/data bus **198** and control bus **200** are connected to the DRAMS **150** (FIG. **3**).

In operation, when the most significant bits **190** of remaining instructions **1–4** are "1" in a clock cycle of the microprocessor **50**, there are no memory reference instructions in the queue. The output of decoder **192** on line **196** requests an instruction fetch ahead by memory controller **118** without interference with other accesses. While the current instructions in instruction register **108** are executing, the memory controller **118** obtains the address of the next set of four instructions from program counter **130** and obtains that set of instructions. By the time the current set of instructions has completed execution, the next set of instructions is ready for loading into the instruction register.

Details of the DMA CPU **72** are provided in FIG. **5**. Internal data bus **90** is connected to memory controller **118** and to DMA instruction register **210**. The DMA instruction register **210** is connected to DMA program counter **212** by bus **214**, to transfer size counter **216** by bus **218** and to timed transfer interval counter **220** by bus **222**. The DMA instruction register **210** is also connected to DMA I/O and RAM address register **224** by line **226**. The DMA I/O and RAM address register **224** is connected to the memory controller **118** by memory cycle request line **228** and bus **230**. The DMA program counter **212** is connected to the internal address bus **136** by bus **232**. The transfer size counter **216** is connected to a DMA instruction done decrementer **234** by lines **236** and **238**. The decrementer **234** receives a control input on memory cycle acknowledge line **240**. When transfer size counter **216** has completed its count, it provides a control signal to DMA program counter **212** on line **242**. Timed transfer interval counter **220** is connected to decrementer **244** by lines **246** and **248**. The decrementer **244** receives a control input from a microprocessor system clock on line **250**.

The DMA CPU **72** controls itself and has the ability to fetch and execute instructions. It operates as a co-processor to the main CPU **70** (FIG. **2**) for time specific processing.

FIG. **6** shows how the microprocessor **50** is connected to an electrically programmable read only memory (EPROM) **260** by reconfiguring the data lines **52** so that some of the data lines **52** are input lines and some of them are output lines. Data lines **52** D0–D7 provide data to and from corresponding data terminals **262** of the EPROM **260**. Data lines **52** D9–D18 provide addresses to address terminals **264** of the EPROM **260**. Data lines **52** D19–D31 provide inputs from the microprocessor **50** to memory and I/O decode logic **266**. RAS **0/1** control line **268** provides a control signal for determining whether the memory and I/O decode logic provides a DRAM RAS output on line **270** or a column enable output for the EPROM **260** on line **272**. Column address strobe terminal **60** of the microprocessor **50** provides an output enable signal on line **274** to the corresponding terminal **276** of the EPROM **260**.

FIGS. **7** and **8** show the front and back of a one card data processing system **280** incorporating the microprocessor **50**, MSM514258-10 type DRAMs **150** totalling 2 megabytes, a Motorola 50 MegaHertz crystal oscillator clock **282**, I/O circuits **284** and a 27256 type EPROM **260**. The I/O circuits **284** include a 74HC04 type high speed hex inverter circuit **286**, an IDT39C828 type 10-bit inverting buffer circuit **288**, an IDT39C822 type 10-bit inverting register circuit **290**, and two IDT39C823 type 9-bit non-inverting register circuits **292**. The card **280** is completed with a MAX12V type DC—DC converter circuit **294**, 34-pin dual AMP type headers **296**, a coaxial female power connector **298**, and a 3-pin AMP right angle header **300**. The card **280** is a low cost, imbeddable product that can be incorporated in larger systems or used as an internal development tool.

The microprocessor **50** is a very high performance (50 MHz) RISC influenced 32-bit CPU designed to work closely with dynamic RAM. Clock for clock, the microprocessor **50** approaches the theoretical performance limits possible with a single CPU configuration. Eventually, the microprocessor **50** and any other processor is limited by the bus bandwidth and the number of bus paths. The critical conduit is between the CPU and memory.

One solution to the bus bandwidth/bus path problem is to integrate a CPU directly onto the memory chips, giving every memory a direct bus the CPU. FIG. **9** shows another microprocessor **310** that is provided integrally with 1 mega-

bit of DRAM **311** in a single integrated circuit **312**. Until the present invention, this solution has not been practical, because most high performance CPUs require from 500,000 to 1,000,000 transistors and enormous die sizes just by themselves. The microprocessor **310** is equivalent to the microprocessor **50** in FIGS. **1–8**. The microprocessors **50** and **310** are the most transistor efficient high performance CPUs in existence, requiring fewer than 50,000 transistors for dual processors **70** and **72** (FIG. **2**) or **314** and **316** (less memory). The very high speed of the microprocessors **50** and **310** is to a certain extent a function of the small number of active devices. In essence, the less silicon gets in the way, the faster the electrons can get where they are going.

The microprocessor **310** is therefore the only CPU suitable for integration on the memory chip die **312**. Some simple modifications to the basic microprocessor **50** to take advantage of the proximity to the DRAM array **311** can also increase the microprocessor **50** clock speed by 50 percent, and probably more.

The microprocessor **310** core on board the DRAM die **312** provides most of the speed and functionality required for a large group of applications from automotive to peripheral control. However, the integrated CPU **310**/DRAM **311** concept has the potential to redefine significantly the way multiprocessor solutions can solve a spectrum of very compute intensive problems. The CPU **310**/DRAM **311** combination eliminates the Von Neumann bottleneck by distributing it across numerous CPU/DRAM chips **312**. The microprocessor **310** is a particularly good core for multiprocessing, since it was designed with the SDI targeting array in mind, and provisions were made for efficient interprocessor communications.

Traditional multiprocessor implementations have been very expensive in addition to being unable to exploit fully the available CPU horsepower. Multiprocessor systems have typically been built up from numerous board level or box level computers. The result is usually an immense amount of hardware with corresponding wiring, power consumption and communications problems. By the time the systems are interconnected, as much as 50 percent of the bus speed has been utilized just getting through the interfaces.

In addition, multiprocessor system software has been scarce. A multiprocessor system can easily be crippled by an inadequate load-sharing algorithm in the system software, which allows one CPU to do a great deal of work and the others to be idle. Great strides have been made recently in systems software, and even UNIX V.4 may be enhanced to support multiprocessing. Several commercial products from such manufacturers as DUAL Systems and UNISOFT do a credible job on 68030 type microprocessor systems now.

The microprocessor **310** architecture eliminates most of the interface friction, since up to 64 CPU **310**/RAM **311** processors should be able to intercommunicate without buffers or latches. Each chip **312** has about 40 MIPS raw speed, because placing the DRAM **311** next to the CPU **310** allows the microprocessor **310** instruction cycle to be cut in half, compared to the microprocessor **50**. A 64 chip array of these chips **312** is more powerful than any other existing computer. Such an array fits on a 3×5 card, cost less than a FAX machine, and draw about the same power as a small television.

Dramatic changes in price/performance always reshape existing applications and almost always create new ones. The introduction of microprocessors in the mid 1970s created video games, personal computers, automotive computers, electronically controlled appliances, and low cost computer peripherals.

The integrated circuit **312** will find applications in all of the above areas, plus create some new ones. A common generic parallel processing algorithm handles convolution/Fast Fourier Transform (FFT)/pattern recognition. Interesting product possibilities using the integrated circuit **312** include high speed reading machines, real-time speech recognition, spoken language translation, real-time robot vision, a product to identify people by their faces, and an automotive or aviation collision avoidance system.

A real time processor for enhancing high density television (HDTV) images, or compressing the HDTV information into a smaller bandwidth, would be very feasible. The load sharing in HDTV could be very straightforward. Splitting up the task according to color and frame would require 6, 9 or 12 processors. Practical implementation might require 4 meg RAMs integrated with the microprocessor **310**.

The microprocessor **310** has the following specifications:

CONTROL LINES
4 - POWER/GROUND
1 - CLOCK
32 - DATA I/O
4 - SYSTEM CONTROL
EXTERNAL MEMORY FETCH
EXTERNAL MEMORY FETCH AUTOINCREMENT X
EXTERNAL MEMORY FETCH AUTOINCREMENT Y
EXTERNAL MEMORY WRITE
EXTERNAL MEMORY WRITE AUTOINCREMENT X
EXTERNAL MEMORY WRITE AUTOINCREMENT Y
EXTERNAL PROM FETCH
LOAD ALL X REGISTERS
LOAD ALL Y REGISTERS
LOAD ALL PC REGISTERS
EXCHANGE X AND Y
INSTRUCTION FETCH
ADD TO PC
ADD TO X
WRITE MAPPING REGISTER
READ MAPPING REGISTER
REGISTER CONFIGURATION
MICROPROCESSOR **310** CPU **316** CORE
COLUMN LATCH1 (1024 BITS) 32×32 MUX
STACK POINTER (16 BITS)
COLUMN LATCH2 (1024 BITS) 32×32 MUX
RSTACK POINTER (16 BITS)
PROGRAM COUNTER 32 BITS
X0 REGISTER 32 BITS (ACTIVATED ONLY FOR ON-CHIP ACCESSES)
Y0 REGISTER 32 BITS (ACTIVATED ONLY FOR ON-CHIP ACCESSES)
LOOP COUNTER 32 BITS
DMA CPU **314** CORE
DMA PROGRAM COUNTER 24 BITS
INSTRUCTION REGISTER 32 BITS
I/O & RAM ADDRESS REGISTER 32 BITS
TRANSFER SIZE COUNTER 12 BITS
INTERVAL COUNTER 12 BITS

To offer memory expansion for the basic chip **312**, an intelligent DRAM can be produced. This chip will be optimized for high speed operation with the integrated circuit **312** by having three on-chip address registers: Program Counter, X Register and Y register. As a result, to access the intelligent DRAM, no address is required, and a total access cycle could be as short as 10 nsec. Each expansion DRAM would maintain its own copy of the three registers and would be identified by a code specifying its memory address. Incrementing and adding to the three

registers will actually take place on the memory chips. A maximum of 64 intelligent DRAM peripherals would allow a large system to be created without sacrificing speed by introducing multiplexers or buffers.

There are certain differences between the microprocessor **310** and the microprocessor **50** that arise from providing the microprocessor **310** on the same die **312** with the DRAM **311**. Integrating the DRAM **311** allows architectural changes in the microprocessor **310** logic to take advantage of existing on-chip DRAM **311** circuitry. Row and column design is inherent in memory architecture. The DRAMs **311** access random bits in a memory array by first selecting a row of 1024 bits, storing them into a column latch, and then selecting one of the bits as the data to be read or written.

The time required to access the data is split between the row access and the column access. Selecting data already stored in a column latch is faster than selecting a random bit by at least a factor of six. The microprocessor **310** takes advantage of this high speed by creating a number of column latches and using them as caches and shift registers. Selecting a new row of information may be thought of as performing a 1024-bit read or write with the resulting immense bus bandwidth.

1. The microprocessor **50** treats its 32-bit instruction register **108** (see FIGS. **2** and **4**) as a cache for four 8-bit instructions. Since the DRAM **311** maintains a 1024-bit latch for the column bits, the microprocessor **310** treats the column latch as a cache for 128 8-bit instructions. Therefore, the next instruction will almost always be already present in the cache. Long loops within the cache are also possible and more useful than the 4 instruction loops in the microprocessor **50**.

2. The microprocessor **50** uses two 16×32-bit deep register arrays **74** and **134** (FIG. **2**) for the parameter stack and the return stack. The microprocessor **310** creates two other 1024-bit column latches to provide the equivalent of two 32×32-bit arrays, which can be accessed twice as fast as a register array.

3. The microprocessor **50** has a DMA capability which can be used for I/O to a video shift register. The microprocessor **310** uses yet another 1024-bit column latch as a long video shift register to drive a CRT display directly. For color displays, three on-chip shift registers could also be used. These shift registers can transfer pixels at a maximum of 100 MHz.

4. The microprocessor **50** accesses memory via an external 32-bit bus. Most of the memory **311** for the microprocessor **310** is on the same die **312**. External access to more memory is made using an 8-bit bus. The result is a smaller die, smaller package and lower power consumption than the microprocessor **50**.

5. The microprocessor **50** consumes about a third of its operating power charging and discharging the I/O pins and associated capacitances. The DRAMs **150** (FIG. **8**) connected to the microprocessor **50** dissipate most of their power in the I/O drivers. A microprocessor **310** system will consume about one-tenth the power of a microprocessor **50** system, since having the DRAM **311** next to the processor **310** eliminates most of the external capacitances to be charged and discharged.

6. Multiprocessing means splitting a computing task between numerous processors in order to speed up the solution. The popularity of multiprocessing is limited by the expense of current individual processors as well as the limited interprocessor communications ability. The microprocessor **310** is an excellent multiprocessor candidate, since the chip **312** is a monolithic computer complete with memory, rendering it low-cost and physically compact.

The shift registers implemented with the microprocessor **310** to perform video output can also be configured as interprocessor communication links. The INMOS transputer attempted a similar strategy, but at much lower speed and without the performance benefits inherent in the microprocessor **310** column latch architecture. Serial I/O is a prerequisite for many multiprocessor topologies because of the many neighbor processors which communicate. A cube has 6 neighbors. Each neighbor communicates using these lines:

DATA IN
CLOCK IN
READY FOR DATA
DATA OUT
DATA READY?
CLOCK OUT

A special start up sequence is used to initialize the on-chip DRAM **311** in each of the processors.

The microprocessor **310** column latch architecture allows neighbor processors to deliver information directly to internal registers or even instruction caches of other chips **312**. This technique is not used with existing processors, because it only improves performance in a tightly coupled DRAM system.

7. The microprocessor **50** architecture offers two types of looping structures: LOOP-IF-DONE and MICRO-LOOP. The former takes an 8-bit to 24-bit operand to describe the entry point to the loop address. The latter performs a loop entirely within the 4 instruction queue and the loop entry point is implied as the first instruction in the queue. Loops entirely within the queue run without external instruction fetches and execute up to three times as fast as the long loop construct. The microprocessor **310** retains both constructs with a few differences. The microprocessor **310** microloop functions in the same fashion as the microprocessor **50** operation, except the queue is 1024-bits or 128 8-bit instructions long. The microprocessor **310** microloop can therefore contain jumps, branches, calls and immediate operations not possible in the 4 8-bit instruction microprocessor **50** queue.

Microloops in the microprocessor **50** can only perform simple block move and compare functions. The larger microprocessor **310** queue allows entire digital signal processing or floating point algorithms to loop at high speed in the queue.

The microprocessor **50** offers four instructions to redirect execution:

CALL
BRANCH
BRANCH-IF-ZERO
LOOP-IF-NOT-DONE

These instructions take a variable length address operand 8, 16 or 24 bits long. The microprocessor **50** next address logic treats the three operands similarly by adding or subtracting them to the current program counter. For the microprocessor **310**, the 16 and 24-bit operands function in the same manner as the 16 and 24-bit operands in the microprocessor **50**. The 8-bit class operands are reserved to operate entirely within the instruction queue. Next address decisions can therefore be made quickly, because only 10 bits of addresses are affected, rather than 32. There is no carry or borrow generated past the 10 bits.

8. The microprocessor **310** CPU **316** resides on an already crowded DRAM die **312**. To keep chip size as small as possible, the DMA processor **72** of the microprocessor **50** has been replaced with a more traditional DMA controller **314**. DMA is used with the microprocessor **310** to perform the following functions:

Video output to a CRT

Multiprocessor serial communications
8-bit parallel I/O

The DMA controller **314** can maintain both serial and parallel transfers simultaneously. The following DMA sources and destinations are supported by the microprocessor **310**:

DESCRIPTION	I/O	LINES
1. Video shift register	OUTPUT	1 to 3
2. Multiprocessor serial	BOTH	6 lines/channel
3. 8-bit parallel	BOTH	8 data, 4 control

The three sources use separate 1024-bit buffers and separate I/O pins. Therefore, all three may be active simultaneously without interference.

The microprocessor **310** can be implemented with either a single multiprocessor serial buffer or separate receive and sending buffers for each channel, allowing simultaneous bidirectional communications with six neighbors simultaneously.

FIGS. **10** and **11** provide details of the PROM DMA used in the microprocessor **50**. The microprocessor **50** executes faster than all but the fastest PROMs. PROMS are used in a microprocessor **50** system to store program segments and perhaps entire programs. The microprocessor **50** provides a feature on power-up to allow programs to be loaded from low-cost, slow speed PROMs into high speed DRAM for execution. The logic which performs this function is part of the DMA memory controller **118**. The operation is similar to DMA, but not identical, since four 8-bit bytes must be assembled on the microprocessor **50** chip, then written to the DRAM **150**.

The microprocessor **50** directly interfaces to DRAM **150** over a triple multiplexed data and address bus **350**, which carries RAS addresses, CAS addresses and data. The EPROM **260**, on the other hand, is read with non-multiplexed busses. The microprocessor **50** therefore has a special mode which unmultiplexes the data and address lines to read 8 bits of EPROM data. Four 8-bit bytes are read in this fashion. The multiplexed bus **350** is turned back on, and the data is written to the DRAM **150**.

When the microprocessor **50** detects a RESET condition, the processor stops the main CPU **70** and forces a mode **0** (PROM LOAD) instruction into the DMA CPU **72** instruction register. The DMA instruction directs the memory controller to read the EPROM **260** data at 8 times the normal access time for memory. Assuming a 50 MHz microprocessor **50**, this means an access time of 320 nsec. The instruction also indicates:

The selection address of the EPROM **260** to be loaded,
The number of 32-bit words to transfer,
The DRAM **150** address to transfer into.

The sequence of activities to transfer one 32-bit word from EPROM **260** to DRAM **150** are:

1. RAS goes low at **352**, latching the EPROM **260** select information from the high order address bits. The EPROM **260** is selected.
2. Twelve address bits (consisting of what is normally DRAM CAS addresses plus two byte select bits are placed on the bus **350** going to the EPROM **260** address pins. These signals will remain on the lines until the data from the EPROM **260** has been read into the microprocessor **50**. For the first byte, the byte select bits will be binary 00.
3. CAS goes low at **354**, enabling the EPROM **260** data onto the lower 8 bits of the external address/data bus **350**. NOTE: It is important to recognize that, during this part of the cycle, the lower 8 bits of the external data/address bus are functioning as inputs, but the rest of the bus is still acting as outputs.
4. The microprocessor **50** latches these eight least significant bits internally and shifts them 8 bits left to shift them to the next significant byte position.
5. Steps **2**, **3** and **4** are repeated with byte address **01**.
6. Steps **2**, **3** and **4** are repeated with byte address **10**.
7. Steps **2**, **3** and **4** are repeated with byte address **11**.
8. CAS goes high at **356**, taking the EPROM **260** off the data bus.
9. RAS goes high at **358**, indicating the end of the EPROM **260** access.
10. RAS goes low at **360**, latching the DRAM select information from the high order address bits. At the same time, the RAS address bits are latched into the DRAM **150**. The DRAM **150** is selected.
11. CAS goes low at **362**, latching the DRAM **150** CAS addresses.
12. The microprocessor **50** places the previously latched EPROM **260** 32-bit data onto the external address/data bus **350**. W goes low at **364**, writing the 32 bits into the DRAM **150**.
13. W goes high at **366**. CAS goes high at **368**. The process continues with the next word.

FIG. **12** shows details of the microprocessor **50** memory controller **118**. In operation, bus requests stay present until they are serviced. CPU **70** requests are prioritized at **370** in the order of: 1, Parameter Stack; 2, Return Stack; 3, Data Fetch; 4, Instruction Fetch. The resulting CPU request signal and a DMA request signal are supplied as bus requests to bus control **372**, which provides a bus grant signal at **374**. Internal address bus **136** and a DMA counter **376** provide inputs to a multiplexer **378**. Either a row address or a column address are provided as an output to multiplexed address bus **380** as an output from the multiplexer **378**. The multiplexed address bus **380** and the internal data bus **90** provide address and data inputs, respectively, to multiplexer **382**. Shift register **384** supplies row address strobe (RAS) **1** and **2** control signals to multiplexer **386** and column address strobe (CAS) **1** and **2** control signals to multiplexer **388** on lines **390** and **392**. The shift register **384** also supplies output enable (OE) and write (W) signals on lines **394** and **396** and a control signal on line **398** to multiplexer **382**. The shift register **384** receives a RUN signal on line **400** to generate a memory cycle and supplies a MEMORY READY signal on line **402** when an access is complete.

STACK/REGISTER ARCHITECTURE

Most microprocessors use on-chip registers for temporary storage of variables. The on-chip registers access data faster than off-chip RAM. A few microprocessors use an on-chip push down stack for temporary storage.

A stack has the advantage of faster operation compared to on-chip registers by avoiding the necessity to select source and destination registers. (A math or logic operation always uses the top two stack items as source and the top of stack as destination.) The stack's disadvantage is that it makes some operations clumsy. Some compiler activities in particular require on-chip registers for efficiency.

As shown in FIG. **13**, the microprocessor **50** provides both on-chip registers **134** and a stack **74** and reaps the benefits of both.

BENEFITS:

1. Stack math and logic is twice as fast as those available on an equivalent register only machine. Most program-

mers and optimizing compilers can take advantage of this feature.

2. Sixteen registers are available for on-chip storage of local variables which can transfer to the stack for computation. The accessing of variables is three to four times as fast as available on a strictly stack machine.

The combined stack **74**/register **134** architecture has not been used previously due to inadequate understanding by computer designers of optimizing compilers and the mix of transfer versus math/logic instructions.

ADAPTIVE MEMORY CONTROLLER

A microprocessor must be designed to work with small or large memory configurations. As more memory loads are added to the data, address, and control lines, the switching speed of the signals slows down. The microprocessor **50** multiplexes the address/data bus three ways, so timing between the phases is critical. A traditional approach to the problem allocates a wide margin of time between bus phases so that systems will work with small or large numbers of memory chips connected. A speed compromise of as much as 50% is required.

As shown in FIG. **14**, the microprocessor **50** uses a feedback technique to allow the processor to adjust memory bus timing to be fast with small loads and slower with large ones. The OUTPUT ENABLE (OE) line **152** from the microprocessor **50** is connected to all memories **150** on the circuit board. The loading on the output enable line **152** to the microprocessor **50** is directly related to the number of memories **150** connected. By monitoring how rapidly OE **152** goes high after a read, the microprocessor **50** is able to determine when the data hold time has been satisfied and place the next address on the bus.

The level of the OE line **152** is monitored by CMOS input buffer **410** which generates an internal READY signal on line **412** to the microprocessor's memory controller. Curves **414** and **416** of the FIG. **15** graph show the difference in rise time likely to be encountered from a lightly to heavily loaded memory system. When the OE line **152** has reached a predetermined level to generate the READY signal, driver **418** generates an OUTPUT ENABLE signal on OE line **152**.

SKIP WITHIN THE INSTRUCTION CACHE

The microprocessor **50** fetches four 8-bit instructions each memory cycle and stores them in a 32-bit instruction register **108**, as shown in FIG. **16**. A class of "test and skip" instructions can very rapidly execute a very fast jump operation within the four instruction cache.

SKIP CONDITIONS:

Always
ACC non-zero
ACC negative
Carry flag equal logic one
Never
ACC equal zero
ACC positive
Carry flag equal logic zero

The SKIP instruction can be located in any of the four byte positions **420** in the 32-bit instruction register **108**. If the test is successful, SKIP will jump over the remaining one, two, or three 8-bit instructions in the instruction register **108** and cause the next four-instruction group to be loaded into the register **108**. As shown, the SKIP operation is implemented by resetting the 2-bit microinstruction counter **180** to zero on line **422** and simultaneously latching the next instruction group into the register **108**. Any instructions following the SKIP in the instruction register are overwritten by the new instructions and not executed.

The advantage of SKIP is that optimizing compilers and smart programmers can often use it in place of the longer conditional JUMP instruction. SKIP also makes possible microloops which exit when the loop counts down or when the SKIP jumps to the next instruction group. The result is very fast code.

Other machines (such as the PDP-**8** and Data General NOVA) provide the ability to skip a single instruction. The microprocessor **50** provides the ability to skip up to three instructions.

MICROLOOP IN THE INSTRUCTION CACHE

The microprocessor **50** provides the MICROLOOP instruction to execute repetitively from one to three instructions residing in the instruction register **108**. The microloop instruction works in conjunction with the LOOP COUNTER **92** (FIG. **2**) connected to the internal data bus **90**. To execute a microloop, the program stores a count in LOOP COUNTER **92**. MICROLOOP may be placed in the first, second, third, or last byte **420** of the instruction register **108**. If placed in the first position, execution will just create a delay equal to the number stored in LOOP COUNTER **92** times the machine cycle. If placed in the second, third, or last byte **420**, when the microloop instruction is executed, it will test the LOOP COUNT for zero. If zero, execution will continue with the next instruction. If not zero, the LOOP COUNTER **92** is decremented and the 2-bit microinstruction counter is cleared, causing the preceding instructions in the instruction register to be executed again.

Microloop is useful for block move and search operations. By executing a block move completely out of the instruction register **108**, the speed of the move is doubled, since all memory cycles are used by the move rather than being shared with instruction fetching. Such a hardware implementation of microloops is much faster than conventional software implementation of a comparable function.

OPTIMAL CPU CLOCK SCHEME

The designer of a high speed microprocessor must produce a product which operate over wide temperature ranges, wide voltage swings, and wide variations in semiconductor processing. Temperature, voltage, and process all affect transistor propagation delays. Traditional CPU designs are done so that with the worse case of the three parameters, the circuit will function at the rated clock speed. The result are designs that must be clocked a factor of two slower than their maximum theoretical performance, so they will operate properly in worse case conditions.

The microprocessor **50** uses the technique shown in FIGS. **17–19** to generate the system clock and its required phases. Clock circuit **430** is the familiar "ring oscillator" used to test process performance. The clock is fabricated on the same silicon chip as the rest of the microprocessor **50**.

The ring oscillator frequency is determined by the parameters of temperature, voltage, and process. At room temperature, the frequency will be in the neighborhood of 100 MHZ. At 70 degrees Centigrade, the speed will be 50 MHZ. The ring oscillator **430** is useful as a system clock, with its stages **431** producing phase 0-phase **3** outputs **433** shown in FIG. **19**, because its performance tracks the parameters which similarly affect all other transistors on the same silicon die. By deriving system timing from the ring oscillator **430**, CPU **70** will always execute at the maximum frequency possible, but never too fast. For example, if the processing of a particular die is not good resulting in slow transistors, the latches and gates on the microprocessor **50** will operate slower than normal. Since the microprocessor **50** ring oscillator clock **430** is made from the same transistors on the same die as the latches and gates, it too will

operate slower (oscillating at a lower frequency), providing compensation which allows the rest of the chip's logic to operate properly.

ASYNCHRONOUS/SYNCHRONOUS CPU

Most microprocessors derive all system timing from a single clock. The disadvantage is that different parts of the system can slow all operations. The microprocessor **50** provides a dual-clock scheme as shown in FIG. **17**, with the CPU **70** operating asynchronously to I/O interface **432** forming part of memory controller **118** (FIG. **2**) and the I/O interface **432** operating synchronously with the external world of memory and I/O devices. The CPU **70** executes at the fastest speed possible using the adaptive ring oscillator clock **430**. Speed may vary by a factor of four depending upon temperature, voltage, and process. The external world must be synchronized to the microprocessor **50** for operations such as video display updating and disc drive reading and writing. This synchronization is performed by the I/O interface **432**, speed of which is controlled by a conventional crystal clock **434**. The interface **432** processes requests for memory accesses from the microprocessor **50** and acknowledges the presence of I/O data. The microprocessor **50** fetches up to four instructions in a single memory cycle and can perform much useful work before requiring another memory access. By decoupling the variable speed of the CPU **70** from the fixed speed of the I/O interface **432**, optimum performance can be achieved by each. Recoupling between the CPU **70** and the interface **432** is accomplished with hand shake signals on lines **436**, with data/addresses passing on bus **90**, **136**.

ASYNCHRONOUS/SYNCHRONOUS CPU IMBEDDED ON A DRAM CHIP

System performance is enhanced even more when the DRAM **311** and CPU **314** (FIG. **9**) are located on the same die. The proximity of the transistors means that DRAM **311** and CPU **314** parameters will closely follow each other. At room temperature, not only would the CPU **314** execute at 100 MHZ, but the DRAM **311** would access fast enough to keep up. The synchronization performed by the I/O interface **432** would be for DMA and reading and writing I/O ports. In some systems (such as calculators) no I/O synchronization at all would be required, and the I/O clock would be tied to the ring counter clock.

VARIABLE WIDTH OPERANDS

Many microprocessors provide variable width operands. The microprocessor **50** handles operands of 8, 16, or 24 bits using the same op-code. FIG. **20** shows the 32-bit instruction register **108** and the 2-bit microinstruction register **180** which selects the 8-bit instruction. Two classes of microprocessor **50** instructions can be greater than 8-bits, JUMP class and IMMEDIATE. A JUMP or IMMEDIATE op-code is 8-bits, but the operand can be 8, 16, or 24 bits long. This magic is possible because operands must be right justified in the instruction register. This means that the least significant bit of the operand is always located in the least significant bit of the instruction register. The microinstruction counter **180** selects which 8-bit instruction to execute. If a JUMP or IMMEDIATE instruction is decoded, the state of the 2-bit microinstruction counter selects the required 8, 16, or 24 bit operand onto the address or data bus. The unselected 8-bit bytes are loaded with zeros by operation of decoder **440** and gates **442**. The advantage of this technique is the saving of a number of op-codes required to specify the different operand sizes in other microprocessors.

TRIPLE STACK CACHE

Computer performance is directly related to the system memory bandwidth. The faster the memories, the faster the computer. Fast memories are expensive, so techniques have been developed to move a small amount of high-speed memory around to the memory addresses where it is needed. A large amount of slow memory is constantly updated by the fast memory, giving the appearance of a large fast memory array. A common implementation of the technique is known as a high-speed memory cache. The cache may be thought of as fast acting shock absorber smoothing out the bumps in memory access. When more memory is required than the shock can absorb, it bottoms out and slow speed memory is accessed. Most memory operations can be handled by the shock absorber itself.

The microprocessor **50** architecture has the ALU **80** (FIG. **2**) directly coupled to the top two stack locations **76** and **78**. The access time of the stack **74** therefore directly affects the execution speed of the processor. The microprocessor **50** stack architecture is particularly suitable to a triple cache technique, shown in FIG. **21** which offers the appearance of a large stack memory operating at the speed of on-chip latches **450**. Latches **450** are the fastest form of memory device built on the chip, delivering data in as little as 3 nsec. However latches **450** require large numbers of transistors to construct. On-chip RAM **452** requires fewer transistors than latches, but is slower by a factor of five (15 nsec access). Off-chip RAM **150** is the slowest storage of all. The microprocessor **50** organizes the stack memory hierarchy as three interconnected stacks **450**, **452** and **454**. The latch stack **450** is the fastest and most frequently used. The on-chip RAM stack **452** is next. The off-chip RAM stack **454** is slowest. The stack modulation determines the effective access time of the stack. If a group of stack operations never push or pull more than four consecutive items on the stack, operations will be entirely performed in the 3 nsec latch stack. When the four latches **456** are filled, the data in the bottom of the latch stack **450** is written to the top of the on-chip RAM stack **452**. When the sixteen locations **458** in the on-chip RAM stack **452** are filled, the data in the bottom of the on-chip RAM stack **452** is written to the top of the off-chip RAM stack **454**. When popping data off a full stack **450**, four pops will be performed before stack empty line **460** from the latch stack pointer **462** transfers data from the on-chip RAM stack **452**. By waiting for the latch stack **450** to empty before performing the slower on-chip RAM access, the high effective speed of the latches **456** are made available to the processor. The same approach is employed with the on-chip RAM stack **452** and the off-chip RAM stack **454**.

POLYNOMIAL GENERATION INSTRUCTION

Polynomials are useful for error correction, encryption, data compression, and fractal generation. A polynomial is generated by a sequence of shift and exclusive OR operations. Special chips are provided for this purpose in the prior art.

The microprocessor **50** is able to generate polynomials at high speed without external hardware by slightly modifying how the ALU **80** works. As shown in FIG. **21**, a polynomial is generated by loading the "order" (also known as the feedback terms) into C Register **470**. The value thirty one (resulting in 32 iterations) is loaded into DOWN COUNTER **472**. A register **474** is loaded with zero. B register **476** is loaded with the starting polynomial value. When the POLY instruction executes, C register **470** is exclusively ORed with A register **474** if the least significant bit of B register **476** is a one. Otherwise, the contents of the A register **474** passes through the ALU **80** unaltered. The combination of A and B is then shifted right (divided by 2) with shifters **478** and **480**. The operation automatically repeats the specified number of iterations, and the resulting polynomial is left in A register **474**.

FAST MULTIPLY

Most microprocessors offer a 16×16 or 32×32 bit multiply instruction. Multiply when performed sequentially takes one shift/add per bit, or 32 cycles for 32 bit data. The microprocessor **50** provides a high speed multiply which allows multiplication by small numbers using only a small number of cycles. FIG. **23** shows the logic used to implement the high speed algorithm. To perform a multiply, the size of the multiplier less one is placed in the DOWN COUNTER **472**. For a four bit multiplier, the number three would be stored in the DOWN COUNTER **472**. Zero is loaded into the A register **474**. The multiplier is written bit reversed into the B Register **476**. For example, a bit reversed five (binary 0101) would be written into B as 1010. The multiplicand is written into the C register **470**. Executing the FAST MULT instruction will leave the result in the A Register **474**, when the count has been completed. The fast multiply instruction is important because many applications scale one number by a much smaller number. The difference in speed between multiplying a 32×32 bit and a 32×4 bit is a factor of 8. If the least significant bit of the multiplier is a "ONE", the contents of the A register **474** and the C register **470** are added. If the least significant bit of the multiplier is a "ZERO", the contents of the A register are passed through the ALU **80** unaltered. The output of the ALU **80** is shifted left by shifter **482** in each iteration. The contents of the B register **476** are shifted right by the shifter **480** in each iteration.

INSTRUCTION EXECUTION PHILOSOPHY

The microprocessor **50** uses high speed D latches in most of the speed critical areas. Slower on-chip RAM is used as secondary storage.

The microprocessor **50** philosophy of instruction execution is to create a hierarchy of speed as follows:

Logic and D latch transfers	1 cycle	20 nsec
Math	2 cycles	40 nsec
Fetch/store on-chip RAM	2 cycles	40 nsec
Fetch/store in current RAS page	4 cycles	80 nsec
Fetch/store with RAS cycle	11 cycles	220 nsec

With a 50 MHZ clock, many operations can be performed in 20 nsec. and almost everything else in 40 nsec.

To maximize speed, certain techniques in processor design have been used. They include:

Eliminating arithmetic operations on addresses,
Fetching up to four instructions per memory cycle,
Pipelineless instruction decoding
Generating results before they are needed,
Use of three level stack caching.

PIPELINE PHILOSOPHY

Computer instructions are usually broken down into sequential pieces, for example: fetch, decode, register read, execute, and store. Each piece will require a single machine cycle. In most Reduced Instruction Set Computer (RISC) chips, instruction require from three to six cycles.

RISC instructions are very parallel. For example, each of 70 different instructions in the SPARC (SUN Computer's RISC chip) has five cycles. Using a technique called "pipelining", the different phases of consecutive instructions can be overlapped.

To understand pipelining, think of building five residential homes. Each home will require in sequence, a foundation, framing, plumbing and wiring, roofing, and interior finish. Assume that each activity takes one week. To build one house will take five weeks.

But what if you want to build an entire subdivision? You have only one of each work crew, but when the foundation men finish on the first house, you immediately start them on the second one, and so on. At the end of five weeks, the first home is complete, but you also have five foundations. If you have kept the framing, plumbing, roofing, and interior guys all busy, from five weeks on, a new house will be completed each week.

This is the way a RISC chip like SPARC appears to execute an instruction in a single machine cycle. In reality, a RISC chip is executing one fifth of five instructions each machine cycle. And if five instructions stay in sequence, an instruction will be completed each machine cycle.

The problems with a pipeline are keeping the pipe full with instructions. Each time an out of sequence instruction such as a BRANCH or CALL occurs, the pipe must be refilled with the next sequence. The resulting dead time to refill the pipeline can become substantial when many IF/THEN/ELSE statements or subroutines are encountered.

THE PIPELINE APPROACH

The microprocessor **50** has no pipeline as such. The approach of this microprocessor to speed is to overlap instruction fetching with execution of the previously fetched instruction(s). Beyond that, over half the instructions (the most common ones) execute entirely in a single machine cycle of 20 nsec. This is possible because:

1. Instruction decoding resolves in 2.5 nsec.
2. Incremented/decremented and some math values are calculated before they are needed, requiring only a latching signal to execute.
3. Slower memory is hidden from high speed operations by high-speed D latches which access in 4 nsec.

The disadvantage for this microprocessor is a more complex chip design process. The advantage for the chip user is faster ultimate throughput since pipeline stalls cannot exist. Pipeline synchronization with availability flag bits and other such pipeline handling is not required by this microprocessor.

For example, in some RISC machines an instruction which tests a status flag may have to wait for up to four cycles for the flag set by the previous instruction to be available to be tested. Hardware and software debugging is also somewhat easier because the user doesn't have to visualize five instructions simultaneously in the pipe.

OVERLAPPING INSTRUCTION FETCH/EXECUTE

The slowest procedure the microprocessor **50** performs is to access memory. Memory is accessed when data is read or written. Memory is also read when instructions are fetched. The microprocessor **50** is able to hide fetch of the next instruction behind the execution of the previously fetched instruction(s). The microprocessor **50** fetches instructions in 4-byte instruction groups. An instruction group may contain from one to four instructions. The amount of time required to execute the instruction group ranges from 4 cycles for simple instructions to 64 cycles for a multiply.

When a new instruction group is fetched, the microprocessor instruction decoder looks at the most significant bit of all four of the bytes. The most significant bit of an instruction determines if a memory access is required. For example, CALL, FETCH, and STORE all require a memory access to execute. If all four bytes have nonzero most significant bits, the microprocessor initiates the memory fetch of the next sequential 4-byte instruction group. When the last instruction in the group finishes executing, the next 4-byte instruction group is ready and waiting on the data bus needing only to be latched into the instruction register. If the 4-byte instruction group required four or more cycles to execute and the next sequential access was a column address strobe (CAS) cycle, the instruction fetch was completely overlapped with execution.

INTERNAL ARCHITECTURE

The microprocessor **50** architecture consists of the following:

PARAMETER STACK	<--> ALU* <-->	Y REGISTER RETURN STACK
<--- 32 BITS --- >		<---32 BITS--->
16 DEEP		16 DEEP
Used for math and logic.		Used for subroutine and interrupt return addresses as well as local variables.
Push down stack. Can overflow into off-chip RAM.		Push down stack. Can overflow into off-chip RAM. Can also be accessed relative to top of stack.
LOOP COUNTER		(32-bits, can decrement by 1) Used by class of test and loop instructions.
X REGISTER		(32-bits, can increment or decrement by 4). Used to point to RAM locations.
PROGRAM COUNTER		(32-bits, increments by 4). Points to 4-byte instruction groups in RAM.
INSTRUCTION REG		(32-Bits). Holds 4-byte instruction groups while they are being decoded and executed.

*Math and logic operations use the TOP item and NEXT to top Parameter Stack items as the operands. The result is pushed onto the Parameter Stack.
*Return addresses from subroutines are placed on the Return Stack. The Y REGISTER is used as a pointer to RAM locations. Since the Y REGISTER is the top item of the Return Stack, nesting of indices is straightforward.
MODE - A register with mode and status bits.
MODE-BITS:
Slow down memory accesses by 8 if "1". Run full speed if "0". (Provided for access to slow EPROM.)
Divide the system clock by 1023 if "1" to reduce power consumption. Run full speed if "0". (On-chip counters slow down if this bit is set.)
Enable external interrupt 1.
Enable external interrupt 2.
Enable external interrupt 3.
Enable external interrupt 4.
Enable external interrupt 5.
Enable external interrupt 6.
Enable external interrupt 7.
ON-CHIP MEMORY LOCATIONS:

MODE-BITS	
DMA-POINTER	
DMA-COUNTER	
STACK-POINTER	Pointer into Parameter Stack.
STACK-DEPTH	Depth of on-chip Parameter Stack
RSTACK-POINTER	Pointer into Return Stack
RSTACK-DEPTH	Depth of on-chip Return Stack

ADDRESSING MODE HIGH POINTS

The data bus is 32-bits wide. All memory fetches and stores are 32-bits. Memory bus addresses are 30 bits. The least significant 2 bits are used to select one-of-four bytes in some addressing modes. The Program Counter, X Register, and Y Register are implemented as D latches with their outputs going to the memory address bus and the bus incrementer/decrementer. Incrementing one of these registers can happen quickly, because the incremented value has already rippled through the inc/dec logic and need only be clocked into the latch. Branches and Calls are made to 32-bit word-boundaries.

INSTRUCTION SET

32-BIT INSTRUCTION FORMAT

The thirty two bit instructions are CALL, BRANCH, BRANCH-IF-ZERO, and LOOP-IF-NOT-DONE. These instructions require the calculation of an effective address. In many computers, the effective address is calculated by adding or subtracting an operand with the current Program Counter. This math operation requires from four to seven machine cycles to perform and can definitely bog down machine execution. The microprocessor's strategy is to perform the required math operation at assembly or linking time and do a much simpler "Increment to next page" or "Decrement to previous page" operation at run time. As a result, the microprocessor branches execute in a single cycle.

24-BIT OPERAND FORM:
Byte 1 Byte 2 Byte 3 Byte 4
WWWWWW XX - YYYYYYYY - YYYYYYYY - YYYYYYYY
With a 24-bit operand, the current page is considered to be defined by the most significant 6 bits of the Program Counter.
16-BIT OPERAND FORM:
QQQQQQQQ - WWWWWW XX - YYYYYYYY - YYYYYYYY
With a 16-bit operand, the current page is considered to be defined by the most significant 14 bits of the Program Counter.
8-BIT OPERAND FORM:
QQQQQQQQ - QQQQQQQQ - WWWWWW XX - YYYYYYYY
With an 8-bit operand, the current page is considered to be defined by the most significant 22 bits of the Program Counter.
QQQQQQQQ - Any 8-bit instruction.
WWWWWW - Instruction op-code.
XX - Select how the address bits will be used:
00 - Make all high-order bits zero. (Page zero addressing)
01 - Increment the high-order bits. (Use next page)
10 - Decrement the high-order bits. (Use previous page)
11 - Leave the high-order bits unchanged. (Use current page)
YYYYYYYY - The address operand field. This field is always shifted left two bits (to generate a word rather than byte address) and loaded into the Program Counter. The microprocessor instruction decoder figures out the width of the operand field by the location of the instruction op-code in the four bytes.

The compiler or assembler will normally use the shortest operand required to reach the desired address so that the leading bytes can be used to hold other instructions. The effective address is calculated by combining:

The current Program Counter,

The 8, 16, or 24 bit address operand in the instruction,

Using one of the four allowed addressing modes.

EXAMPLES OF EFFECTIVE ADDRESS CALCULATION

Example 1:

Byte 1	Byte 2	Byte 3	Byte 4
QQQQQQQQ	QQQQQQQQ	00000011	10011000

The "QQQQQQQQs" in Byte **1** and **2** indicate space in the 4-byte memory fetch which could be hold two other instructions to be executed prior to the CALL instruction. Byte **3** indicates a CALL instruction (six zeros) in the current page (indicated by the 11 bits). Byte **4** indicates that the hexadecimal number 98 will be forced into the Program Counter bits **2** through **10**. (Remember, a CALL or BRANCH always goes to a word boundary so the two least

significant bits are always set to zero). The effect of this instruction would be to CALL a subroutine at WORD location HEX 98 in the current page. The most significant 22 bits of the Program Counter define the current page and will be unchanged.

Example 2:

Byte 1	Byte 2	Byte 3	Byte 4
000001 01	00000001	00000000	00000000

If we assume that the Program Counter was HEX 0000 0156 which is binary:

00000000 00000000 00000001 01010110 = OLD PROGRAM COUNTER.

Byte **1** indicates a BRANCH instruction op code (000001) and "01" indicates select the next page. Byte **2,3**, and **4** are the address operand . These 24-bits will be shifted to the left two places to define a WORD address. HEX 0156 shifted left two places is HEX 0558. Since this is a 24-bit operand instruction, the most significant 6 bits of the Program Counter define the current page. These six bits will be incremented to select the next page. Executing this instruction will cause the Program Counter to be loaded with HEX 0400 0558 which is binary:

00000100 00000000 00000101 01011000 = NEW PROGRAM COUNTER.

INSTRUCTIONS

CALL-LONG

0000 00XX - YYYYYYYY - YYYYYYYY - YYYYYYYY

Load the Program Counter with the effective WORD address specified. Push the current PC contents onto the RETURN STACK.

OTHER EFFECTS: CARRY or modes, no effect. May cause Return Stack to force an external memory cycle if on-chip Return Stack is full.

BRANCH

0000 01XX - YYYYYYYY - YYYYYYYY - YYYYYYYY

Load the Program Counter with the effective WORD address specified.

OTHER EFFECTS: NONE

BRANCH-IF-ZERO

0000 10XX - YYYYYYYY - YYYYYYYY - YYYYYYYY

Test the TOP value on the Parameter Stack. If the value is equal to zero, load the Program Counter with the effective WORD address specified. If the TOP value is not equal to zero, increment the Program Counter and fetch and execute the next instruction.

OTHER EFFECTS: NONE

LOOP-IF-NOT-DONE

0000 11YY - (XXXX XXXX) - (XXXX XXXX) - (XXXX XXXX)

If the LOOP COUNTER is not zero, load the Program Counter with the effective WORD address specified. If the LOOP COUNTER is zero, decrement the LOOP COUNTER, increment the Program Counter and fetch and execute the next instruction.

OTHER EFFECTS: NONE

8-BIT INSTRUCTIONS PHILOSOPHY

Most of the work in the microprocessor **50** is done by the 8-bit instructions. Eight bit instructions are possible with the microprocessor because of the extensive use of implied stack addressing. Many 32-bit architectures use 8-bits to specify the operation to perform but use an additional 24-bits to specify two sources and a destination.

For math and logic operations, the microprocessor **50** exploits the inherent advantage of a stack by designating the source operand(s) as the top stack item and the next stack item. The math or logic operation is performed, the operands are popped from the stack, and the result is pushed back on the stack. The result is a very efficient utilization of instruction bits as well as registers. A comparable situation exists between Hewlett Packard calculators (which use a stack) and Texas Instrument calculators which don't. The identical operation on an HP will require one half to one third the keystrokes of the TI.

The availability of 8-bit instructions also allows another architectural innovation, the fetching of four instructions in a single 32-bit memory cycle. The advantages of fetching multiple instructions are:

Increased execution speed even with slow memories,

Similar performance to the Harvard (separate data and instruction busses) without the expense,

Opportunities to optimize groups of instructions,

The capability to perform loops within this mini-cache. The microloops inside the four instruction group are effective for searches and block moves.

SKIP INSTRUCTIONS

The microprocessor **50** fetches instructions in 32-bit chunks called 4-byte instruction groups. These four bytes may contain four 8-bit instructions or some mix of 8-bit and 16 or 24-bit instructions. SKIP instructions in the microprocessor skip any remaining instructions in a 4-byte instruction group and cause a memory fetch to get the next 4-byte instruction group. Conditional SKIPs when combined with 3-byte BRANCHES will create conditional BRANCHES. SKIPs may also be used in situations when no use can be made of the remaining bytes in a 4-instruction group. A SKIP executes in a single cycle, whereas a group of three NOPs would take three cycles.

SKIP-ALWAYS -	skip any remaining instructions in this 4-byte instruction group. Increment the most significant 30-bits of the Program Counter and proceed to fetch the next 4-byte instruction group.
SKIP-IF-ZERO -	If the TOP item of the Parameter Stack is zero, skip any remaining instructions in the 4-byte instruction group. Increment the most significant 30-bits of the Program Counter and proceed to fetch the next 4-byte instruction group. If the TOP item is not zero, execute the next sequential instruction.
SKIP-IF-POSITIVE -	If the TOP item of the Parameter Stack has a the most significant bit (the sign bit) equal to "0", skip any remaining instructions in the 4-byte instruction group. Increment the most significant 30-bits of the Program Counter and proceed to fetch the next 4-byte instruction group. If the TOP item is not "0", execute the next sequential instruction.

-continued

SKIP-IF-NO-CARRY -	If the CARRY flag from a SHIFT or arithmetic operation is not equal to "1", skip any remaining instructions in the 4-byte instruction group. Increment the most significant 30-bits of the Program Counter and proceed to fetch the next 4-byte instruction group. If the CARRY is equal to "1", execute the next sequential instruction.
SKIP-NEVER (NOP)	Execute the next sequential instruction. (Delay one machine cycle).
SKIP-IF-NOT-ZERO -	If the TOP item on the Parameter Stack is not equal to "0", skip any remaining instructions in the 4-byte instruction group. Increment the most significant 30-bits of the Program Counter and proceed to fetch the next 4-byte instruction group. If the TOP item is equal 0", execute the next sequential instruction.
SKIP-IF-NEGATIVE -	If the TOP item on the Parameter Stack has its most significant bit (sign bit) set to "1", skip any remaining instructions in the 4-byte instruction group. Increment the most significant 30-bits of the Program Counter and proceed to fetch the next 4-byte instruction group. If the TOP item has its most significant bit set to "0", execute the next sequential instruction.
SKIP-IF-CARRY -	If the CARRY flag is set to "1" as a result of SHIFT or arithmetic operation, skip any remaining instructions in the 4-byte instruction group. Increment the most significant 30-bits of the Program Counter and proceed to fetch the next 4-byte instruction group. If the CARRY flag is "0", execute the next sequential instruction.

MICROLOOPS

Microloops are a unique feature of the microprocessor architecture which allows controlled looping within a 4-byte instruction group. A microloop instruction tests the LOOP COUNTER for "0" and may perform an additional test. If the LOOP COUNTER is not "0" and the test is met, instruction execution continues with the first instruction in the 4-byte instruction group, and the LOOP COUNTER is decremented. A microloop instruction will usually be the last byte in a 4-byte instruction group, but it can be any byte. If the LOOP COUNTER is "0" or the test is not met, instruction execution continues with the next instruction. If the microloop is the last byte in the 4-byte instruction group, the most significant 30-bits of the Program Counter are incremented and the next 4-byte instruction group is fetched from memory. On a termination of the loop on LOOP COUNTER equal to "0", the LOOP COUNTER will remain at "0". Microloops allow short iterative work such as moves and searches to be performed without slowing down to fetch instructions from memory.

EXAMPLE:

Byte 1 FETCH-VIA-X-AUTOINCREMENT	Byte 2 STORE-VIA-Y-AUTO-INCREMENT
Byte 3 ULOOP-UNTIL-DONE	Byte 4 QQQQQQQQ

This example will perform a block move. To initiate the transfer, X will be loaded with the starting address of the source. Y will be loaded with the starting address of the destination. The LOOP COUNTER will be loaded with the number of 32-bit words to move. The microloop will FETCH and STORE and count down the LOOP COUNTER until it reaches zero. QQQQQQQQ indicates any instruction can follow.

MICROLOOP INSTRUCTIONS

ULOOP-UNTIL-DONE—If the LOOP COUNTER is not "0", continue execution with the first instruction in the 4-byte instruction group. Decrement the LOOP COUNTER. If the LOOP COUNTER is "0", continue execution with the next instruction.

ULOOP-IF-ZERO—If the LOOP COUNTER is not "0" and the TOP item on the Parameter Stack is "0", continue execution with the first instruction in the 4-byte instruction group. Decrement the LOOP COUNTER. If the LOOP COUNTER is "0" or the TOP item is "1", continue execution with the next instruction.

ULOOP-IF-POSITIVE—If the LOOP COUNTER is not "0" and the most significant bit (sign bit) is "0", continue execution with the first instruction in the 4-byte instruction group. Decrement the LOOP COUNTER. If the LOOP COUNTER is "0" or the TOP item is "1", continue execution with the next instruction.

ULOOP-IF-NOT-CARRY-CLEAR—If the LOOP COUNTER is not "0" and the floating point exponents found in TOP and NEXT are not aligned, continue execution with the first instruction in the 4-byte instruction group. Decrement the LOOP COUNTER. If the LOOP COUNTER is "0" or the exponents are aligned, continue execution with the next instruction. This instruction is specifically designed for combination with special SHIFT instructions to align two floating point numbers.

ULOOP-NEVER—(DECREMENT-LOOP-COUNTER) Decrement the LOOP COUNTER. Continue execution with the next instruction.

ULOOP-IF-NOT-ZERO—If the LOOP COUNTER is not "0" and the TOP item of the Parameter Stack is "0", continue execution with the first instruction in the 4-byte instruction group. Decrement the LOOP COUNTER. If the LOOP COUNTER is "0" or the TOP item is "1", continue execution with the next instruction.

ULOOP-IF-NEGATIVE—If the LOOP COUNTER is not "0" and the most significant bit (sign bit) of the TOP item of the Parameter Stack is "1", continue execution with the first instruction in the 4-byte instruction group. Decrement the LOOP COUNTER. If the LOOP COUNTER is "0" or the most significant bit of the Parameter Stack is "0", continue execution with the next instruction.

ULOOP-IF-CARRY-SET—If the LOOP COUNTER is not "0" and the exponents of the floating point numbers found in TOP and NEXT are not aligned, continue execution with the first instruction in the 4-byte instruction group. Decrement the LOOP COUNTER. If the LOOP COUNTER is "0" or the exponents are aligned, continue execution with the next instruction.

RETURN FROM SUBROUTINE OR INTERRUPT

Subroutine calls and interrupt acknowledgements cause a redirection of normal program execution. In both cases, the current Program Counter is pushed onto the Return Stack, so the microprocessor can return to its place in the program after executing the subroutine or interrupt service routine.

NOTE: When a CALL to subroutine or interrupt is acknowledged the Program Counter has already been incremented and is pointing to the 4-byte instruction group following the 4-byte group currently being executed. The instruction decoding logic allows the microprocessor to

perform a test and execute a return conditional on the outcome of the test in a single cycle. A RETURN pops an address from the Return Stack and stores it to the Program Counter.

RETURN INSTRUCTIONS	
RETURN-ALWAYS -	Pop the top item from the Return Stack and transfer it to the Program Counter.
RETURN-IF-ZERO -	If the TOP item on the Parameter Stack is "0", pop the top item from the Return Stack and transfer it to the Program Counter. Otherwise execute the next instruction.
RETURN-IF-POSITIVE -	If the most significant bit (sign bit) of the TOP item on the Parameter Stack is a "0", pop the top item from the Return Stack and transfer it to the Program Counter. Otherwise execute the next instruction.
RETURN-IF-CARRY-CLEAR -	If the exponents of the floating point numbers found in TOP and NEXT are not aligned, pop the top item from the Return Stack and transfer it to the Program Counter. Otherwise execute the next instruction.
RETURN-NEVER - (NOP)	Execute the next instruction.
RETURN-IF-NOT-ZERO -	If the TOP item on the Parameter Stack is not "0", pop the top item from the Return Stack and transfer it to the Program Counter. Otherwise execute the next instruction.
RETURN-IF-NEGATIVE -	If the most significant bit (sign bit) of the TOP item on the Parameter Stack is a "1", pop the top item from the Return Stack and transfer it to the Program Counter. Otherwise execute the next instruction.
RETURN-IF-CARRY-SET -	If the exponents of the floating point numbers found in TOP and NEXT are aligned, pop the top item from the Return Stack and transfer it to the Program. Counter. Otherwise execute the next instruction.

HANDLING MEMORY FROM DYNAMIC RAM

The microprocessor **50**, like any RISC type architecture, is optimized to handle as many operations as possible on-chip for maximum speed. External memory operations take from 80 nsec. to 220 nsec. compared with on-chip memory speeds of from 4 nsec. to 30 nsec. There are times when external memory must be accessed.

External memory is accessed using three registers:

X-REGISTER—A 30-bit memory pointer which can be used for memory access and simultaneously incremented or decremented.

Y-REGISTER—A 30-bit memory pointer which can be used for memory access and simultaneously incremented or decremented.

PROGRAM-COUNTER—A 30-bit memory pointer normally used to point to 4-byte instruction groups. External memory may be accessed at addresses relative to the PC. The operands are sometimes called "Immediate" or "Literal" in other computers. When used as memory pointer, the PC is also incremented after each operation.

MEMORY LOAD & STORE INSTRUCTIONS

FETCH-VIA-X—Fetch the 32-bit memory content pointed to by X and push it onto the Parameter Stack. X is unchanged.

FETCH-VIA-Y—Fetch the 32-bit memory content pointed to by X and push it onto the Parameter Stack. Y is unchanged.

FETCH-VIA-X-AUTOINCREMENT—Fetch the 32-bit memory content pointed to by X and push it onto the Parameter Stack. After fetching, increment the most significant 30 bits of X to point to the next 32-bit word address.

FETCH-VIA-Y-AUTOINCREMENT—Fetch the 32-bit memory content pointed to by Y and push it onto the Parameter Stack. After fetching, increment the most significant 30 bits of Y to point to the next 32-bit word address.

FETCH-VIA-X-AUTODECREMENT—Fetch the 32-bit memory content pointed to by X and push it onto the Parameter Stack. After fetching, decrement the most significant 30 bits of X to point to the previous 32-bit word address.

FETCH-VIA-Y-AUTODECREMENT—Fetch the 32-bit memory content pointed to by Y and push it onto the Parameter Stack. After fetching, decrement the most significant 30 bits of Y to point to the previous 32-bit word address.

STORE-VIA-X—Pop the top item of the Parameter Stack and store it in the memory location pointed to by X. X is unchanged.

STORE-VIA-Y—Pop the top item of the Parameter Stack and store it in the memory location pointed to by Y. Y is unchanged.

STORE-VIA-X-AUTOINCREMENT—Pop the top item of the Parameter Stack and store it in the memory location pointed to by X. After storing, increment the most significant 30 bits of X to point to the next 32-bit word address.

STORE-VIA-Y-AUTOINCREMENT—Pop the top item of the Parameter Stack and store it in the memory location pointed to by Y. After storing, increment the most significant 30 bits of Y to point to the next 32-bit word address.

STORE-VIA-X-AUTODECREMENT—Pop the top item of the Parameter Stack and store it in the memory location pointed to by X. After storing, decrement the most significant 30 bits of X to point to the previous 32-bit word address.

STORE-VIA-Y-AUTODECREMENT—Pop the top item of the Parameter Stack and store it in the memory location pointed to by Y. After storing, decrement the most significant 30 bits of Y to point to the previous 32-bit word address.

FETCH-VIA-PC—Fetch the 32-bit memory content pointed to by the Program Counter and push it onto the Parameter Stack. After fetching, increment the most significant 30 bits of the Program Counter to point to the next 32-bit word address.

*NOTE When this instruction executes, the PC is pointing to the memory location following the instruction. The effect is of loading a 32-bit immediate operand. This is an 8-bit instruction and therefore will be combined with other 8-bit instructions in a 4-byte instruction fetch. It is possible to have from one to four FETCH-VIA-PC instructions in a 4-byte instruction fetch. The PC incre-

ments after each execution of FETCH-VIA-PC, so it is possible to push four immediate operands on the stack. The four operands would be the found in the four memory locations following the instruction.

BYTE-FETCH-VIA-X—Fetch the 32-bit memory content pointed to by the most significant 30 bits of X. Using the two least significant bits of X, select one of four bytes from the 32-bit memory fetch, right justify the byte in a 32-bit field and push the selected byte preceded by leading zeros onto the Parameter Stack.

BYTE-STORE-VIA-X—Fetch the 32-bit memory content pointed to by the most significant 30 bits of X. Pop the TOP item from the Parameter Stack. Using the two least significant bits of X place the least significant byte into the 32-bit memory data and write the 32-bit entity back to the location pointed to by the most significant 30 bits of X.

OTHER EFFECTS OF MEMORY ACCESS INSTRUCTIONS:

Any FETCH instruction will push a value on the Parameter Stack **74**. If the on-chip stack is full, the stack will overflow into off-chip memory stack resulting in an additional memory cycle. Any STORE instruction will pop a value from the Parameter Stack **74**. If the on-chip stack is empty, a memory cycle will be generated to fetch a value from off-chip memory stack.

HANDLING ON-CHIP VARIABLES

High-level languages often allow the creation of LOCAL VARIABLES. These variables are used by a particular procedure and discarded. In cases of nested procedures, layers of these variables must be maintained. On-chip storage is up to five times faster than off-chip RAM, so a means of keeping local variables on-chip can make operations run faster. The microprocessor **50** provides the capability for both on-chip storage of local variables and nesting of multiple levels of variables through the Return Stack.

The Return Stack **134** is implemented as 16 on-chip RAM locations. The most common use for the Return Stack **134** is storage of return addresses from subroutines and interrupt calls. The microprocessor allows these 16 locations to also be used as addressable registers. The 16 locations may be read and written by two instructions which indicate a Return Stack relative address from 0–15. When high-level procedures are nested, the current procedure variables push the previous procedure variables further down the Return Stack **134**. Eventually, the Return Stack will automatically overflow into off-chip RAM.

ON-CHIP VARIABLE INSTRUCTIONS

READ-LOCAL-VARIABLE XXXX—Read the XXXXth location relative to the top of the Return Stack. (XXXX is a binary number from 0000–1111). Push the item read onto the Parameter Stack.

OTHER EFFECTS: If the Parameter Stack is full, the push operation will cause a memory cycle to be generated as one item of the stack is automatically stored to external RAM. The logic which selects the location performs a modulo 16 subtraction. If four local variables have been pushed onto the Return Stack, and an instruction attempts to READ the fifth item, unknown data will be returned.

WRITE-LOCAL-VARIABLE XXXX—Pop the TOP item of the Parameter Stack and write it into the XXXXth location relative to the top of the Return Stack. (XXXX is a binary number from 0000–1111.)

OTHER EFFECTS: If the Parameter Stack is empty, the pop operation will cause a memory cycle to be generated to fetch the Parameter Stack item from external RAM. The logic which selects the location performs a modulo 16 subtraction. If four local variables have been pushed onto the Return Stack, and an instruction attempts to WRITE to the fifth item, it is possible to clobber return addresses or wreak other havoc.

REGISTER AND FLIP-FLOP TRANSFER AND PUSH INSTRUCTIONS

DROP—Pop the TOP item from the Parameter Stack and discard it.

SWAP—Exchange the data in the TOP Parameter Stack location with the data in the NEXT Parameter Stack location.

DUP—Duplicate the TOP item on the Parameter Stack and push it onto the Parameter Stack.

PUSH-LOOP-COUNTER—Push the value in LOOP COUNTER onto the Parameter Stack.

POP-RSTACK-PUSH-TO-STACK—Pop the top item from the Return Stack and push it onto the Parameter Stack.

PUSH-X-REG—Push the value in the X Register onto the Parameter Stack.

PUSH-STACK-POINTER—Push the value of the Parameter Stack pointer onto the Parameter Stack.

PUSH-RSTACK-POINTER—Push the value of the Return Stack pointer onto the Return Stack.

PUSH-MODE-BITS—Push the value of the MODE REGISTER onto the Parameter Stack.

PUSH-INPUT—Read the 10 dedicated input bits and push the value (right justified and padded with leading zeros) onto the Parameter Stack.

SET-LOOP-COUNTER—Pop the TOP value from the Parameter Stack and store it into LOOP COUNTER.

POP-STACK-PUSH-TO-RSTACK—Pop the TOP item from the Parameter Stack and push it onto the Return Stack.

SET-X-REG—Pop the TOP item from the Parameter Stack and store it into the X Register.

SET-STACK-POINTER—Pop the TOP item from the Parameter Stack and store it into the Stack Pointer.

SET-RSTACK-POINTER—Pop the TOP item from the Parameter Stack and store it into the Return Stack Pointer.

SET-MODE-BITS—Pop the TOP value from the Parameter Stack and store it into the MODE BITS.

SET-OUTPUT—Pop the TOP item from the Parameter Stack and output it to the 10 dedicated output bits.

OTHER EFFECTS: Instructions which push or pop the Parameter Stack or Return Stack may cause a memory cycle as the stacks overflow back and forth between on-chip and off-chip memory.

LOADING A SHORT LITERAL

A special case of register transfer instruction is used to push an 8-bit literal onto the Parameter Stack. This instruction requires that the 8-bits to be pushed reside in the last byte of a 4-byte instruction group. The instruction op-code loading the literal may reside in ANY of the other three bytes in the instruction group.

EXAMPLE:

BYTE 1	BYTE 2	BYTE 3
LOAD-SHORT-LITERAL	QQQQQQQQ	QQQQQQQQ
BYTE 4		
00001111		

In this example, QQQQQQQQ indicates any other 8-bit instruction. When Byte **1** is executed, binary 00001111 (HEX 0f) from Byte **4** will be pushed (right justified and padded by leading zeros) onto the Parameter Stack. Then the instructions in Byte **2** and Byte **3** will execute. The microprocessor instruction decoder knows not to execute Byte **4**. It is possible to push three identical 8-bit values as follows:

BYTE 1	BYTE 2
LOAD-SHORT-LITERAL	LOAD-SHORT-LITERAL
BYTE 3	BYTE 4
LOAD-SHORT-LITERAL	00001111
SHORT-LITERAL-INSTRUCTION	
LOAD-SHORT-LITERAL -	Push the 8-bit value found in Byte 4 of the current 4-byte instruction group onto the Parameter Stack.

LOGIC INSTRUCTIONS

Logical and math operations used the stack for the source of one or two operands and as the destination for results. The stack organization is a particularly convenient arrangement for evaluating expressions. TOP indicates the top value on the Parameter Stack **74**. NEXT indicates the next to top value on the Parameter Stack **74**.

AND—Pop TOP and NEXT from the Parameter Stack, perform the logical AND operation on these two operands, and push the result onto the Parameter Stack.

OR—Pop TOP and NEXT from the Parameter Stack, perform the logical OR operation on these two operands, and push the result onto the Parameter Stack.

XOR—Pop TOP and NEXT from the Parameter Stack, perform the logical exclusive OR on these two operands, and push the result onto the Parameter Stack.

BIT-CLEAR—Pop TOP and NEXT from the Parameter Stack, toggle all bits in NEXT, perform the logical AND operation on TOP, and push the result onto the Parameter Stack. (Another way of understanding this instruction is thinking of it as clearing all bits in TOP that are set in NEXT.)

MATH INSTRUCTIONS

Math instruction pop the TOP item and NEXT to top item of the Parameter Stack **74** to use as the operands. The results are pushed back on the Parameter Stack. The CARRY flag is used to latch the "33rd bit" of the ALU result.

ADD—Pop the TOP item and NEXT to top item from the Parameter Stack, add the values together and push the result back on the Parameter Stack. The CARRY flag may be changed.

ADD-WITH-CARRY—Pop the TOP item and the NEXT to top item from the Parameter Stack, add the values together. If the CARRY flag is "1" increment the result. Push the ultimate result back on the Parameter Stack. The CARRY flag may be changed.

ADD-X—Pop the TOP item from the Parameter Stack and read the third item from the top of the Parameter Stack. Add the values together and push the result back on the Parameter Stack. The CARRY flag may be changed.

SUB—Pop the TOP item and NEXT to top item from the Parameter Stack, Subtract NEXT from TOP and push the result back on the Parameter Stack. The CARRY flag may be changed.

SUB-WITH-CARRY—Pop the TOP item and NEXT to top item from the Parameter Stack. Subtract NEXT from TOP. If the CARRY flag is "1" increment the result. Push the ultimate result back on the Parameter Stack. The CARRY flag may be changed.

SUB-X—

SIGNED-MULT-STEP—

UNSIGNED-MULT-STEP—

SIGNED-FAST-MULT—

FAST-MULT-STEP—

UNSIGNED-DIV-STEP—

GENERATE-POLYNOMIAL

ROUND—

COMPARE—Pop the TOP item and NEXT to top item from the Parameter Stack. Subtract NEXT from TOP. If the result has the most significant bit equal to "0" (the result is positive), push the result onto the Parameter Stack. If the result has the most significant bit equal to "1" (the result is negative), push the old value of TOP onto the Parameter Stack. The CARRY flag may be affected.

SHIFT/ROTATE

SHIFT-LEFT—Shift the TOP Parameter Stack item left one bit. The CARRY flag is shifted into the least significant bit of TOP.

SHIFT-RIGHT—Shift the TOP Parameter Stack item right one bit. The least significant bit of TOP is shifted into the CARRY flag. Zero is shifted into the most significant bit of TOP.

DOUBLE-SHIFT-LEFT—Treating the TOP item of the Parameter Stack as the most significant word of a 64-bit number and the NEXT stack item as the least significant word, shift the combined 64-bit entity left one bit. The CARRY flag is shifted into the least significant bit of NEXT.

DOUBLE-SHIFT-RIGHT—Treating the TOP item of the Parameter Stack as the most significant word of a 64-bit number and the NEXT stack item as the least significant word, shift the combined 64-bit entity right one bit. The least significant bit of NEXT is shifted into the CARRY flag. Zero is shifted into the most significant bit of TOP.

OTHER INSTRUCTIONS

FLUSH-STACK—Empty all on-chip Parameter Stack locations into off-chip RAM. (This instruction is useful for multitasking applications). This instruction accesses a counter which holds the depth of the on-chip stack and can require from none to 16 external memory cycles.

FLUSH-RSTACK—Empty all on-chip Return Stack locations into off-chip RAM. (This instruction is useful for multitasking applications). This instruction accesses a counter which holds the depth of the on-chip Return Stack and can require from none to 16 external memory cycles.

It should further be apparent to those skilled in the art that various changes in form and details of the invention as shown and described may be made. It is intended that such changes be included within the spirit and scope of the claims appended hereto.

What is claimed is:

1. A microprocessor, which comprises a main central processing unit and a separate direct memory access central processing unit in a single integrated circuit comprising said microprocessor, said main central processing unit having an arithmetic logic unit, a first push down stack with a top item register and a next item register, connected to provide inputs to said arithmetic logic unit, an output of said arithmetic logic unit being connected to said top item register, said top item register also being connected to provide inputs to an internal data bus, said internal data bus being bidirectionally connected to a loop counter, said loop counter being connected to a decrementer, said internal data bus being bidirectionally connected to a stack pointer, return stack pointer, mode register and instruction register, said internal data bus being connected to a memory controller, to a Y register of a return push down stack, an X register and a program counter, said Y register, X register and program counter providing outputs to an internal address bus, said internal address bus providing inputs to said memory controller and to an incrementer, said incrementer being connected to said internal data bus, said direct memory access central processing unit providing inputs to said memory controller, said memory controller having an address/data bus and a plurality of control lines for connection to a random access memory.

2. The microprocessor of claim **1** in which said memory controller includes a multiplexing means between said central processing unit and said address/data bus, said multiplexing means being connected and configured to provide row addresses, column addresses and data on said address/data bus.

3. The microprocessor of claim **1** in which said memory controller includes means for fetching instructions for said central processing unit on said address/data bus, said means for fetching instructions being configured to fetch multiple sequential instructions in a single memory cycle.

4. The microprocessor of claim **3** additionally comprising means connected to said means for fetching instructions for determining if multiple instructions fetched by said means for fetching instructions require a memory access, said means for fetching instructions fetching additional multiple instructions if the multiple instructions do not require a memory access.

5. The microprocessor of claim **3** in which said microprocessor and a dynamic random access memory are contained in a single integrated circuit and said means for fetching instructions includes a column latch for receiving the multiple instructions.

6. The microprocessor of claim **1** in which said microprocessor includes a sensing circuit and a driver circuit, and an output enable line for connection between the random access memory, said sensing circuit and said driver circuit, said sensing circuit being configured to provide a ready signal when said output enable line reaches a predetermined electrical level, said microprocessor being configured so that said driver circuit provides an enabling signal on said output enable line responsive to the ready signal.

7. The microprocessor of claim **1** additionally comprising a ring oscillator variable speed system clock connected to said main central processing unit, said main central processing unit and said ring oscillator variable speed system clock being provided in a single integrated circuit.

8. The microprocessor of claim **7** in which said memory controller includes an input/output interface connected to exchange coupling control signals, addresses and data with said main central processing unit, said microprocessor additionally including a second clock independent of said ring oscillator variable speed system clock connected to said input/output interface.

9. The microprocessor of claim **1** in which said first push down stack has a first plurality of stack elements configured as latches, a second plurality of stack elements configured as a random access memory, said first and second plurality of stack elements and said central processing unit being provided in a single integrated circuit, and a third plurality of stack elements configured as a random access memory external to said single integrated circuit.

10. The microprocessor of claim **9** additionally comprising a first pointer connected to said first plurality of stack elements, a second pointer connected to said second plurality of stack elements, and a third pointer connected to said third plurality of stack elements, said central processing unit being connected to pop items from said first plurality of stack elements, said first stack pointer being connected to said second stack pointer to pop a first plurality of items from said second plurality of stack elements when said first plurality of stack elements are empty from successive pop operations by said central processing unit, said second stack pointer being connected to said third stack pointer to pop a second plurality of items from said third plurality of stack elements when said second plurality of stack elements are empty from successive pop operations by said central processing unit.

* * * * *

US005530890C1

(12) EX PARTE REEXAMINATION CERTIFICATE (8052nd)

United States Patent

Moore et al.

(10) Number: US 5,530,890 C1

(45) Certificate Issued: Mar. 1, 2011

(54) **HIGH PERFORMANCE, LOW COST MICROPROCESSOR**

(75) Inventors: **Charles H. Moore**, Woodside, CA (US); **Russell H. Fish, III**, Mt. View, CA (US)

(73) Assignee: **Patriot Scientific Corporation**, San Diego, CA (US)

Reexamination Request:
No. 90/009,388, Jan. 16, 2009

Reexamination Certificate for:
Patent No.: **5,530,890**
Issued: **Jun. 25, 1996**
Appl. No.: **08/480,206**
Filed: **Jun. 7, 1995**

Related U.S. Application Data

(62) Division of application No. 07/389,334, filed on Aug. 3, 1989, now Pat. No. 5,440,749.

(51) **Int. Cl.**
G06F 15/76 (2006.01)

(52) **U.S. Cl.** **712/32**; 711/E12.02; 712/E9.016; 712/E9.028; 712/E9.046; 712/E9.055; 712/E9.057; 712/E9.058; 712/E9.062; 712/E9.078; 712/E9.08; 712/E9.081

(58) **Field of Classification Search** ..... None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,603,934	A	9/1971	Heath, Jr. et al.
3,696,414	A	10/1972	Allen et al.
3,810,117	A	5/1974	Healey
3,849,765	A	11/1974	Hamano
3,878,513	A	4/1975	Werner

(Continued)

FOREIGN PATENT DOCUMENTS

EP 113516 A2 7/1984

EP 200797 A1 11/1986

(Continued)

OTHER PUBLICATIONS

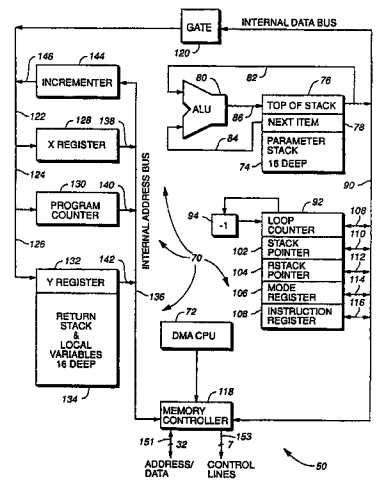
"35ns 256K Device, VLSI Debuts SRAM Designed With Hitachi," Electronic News, p. 25 (Apr. 17, 1989).

(Continued)

Primary Examiner—Joseph R. Pokrzywa

(57) **ABSTRACT**

A microprocessor (**50**) includes a main central processing unit (CPU) (**70**) and a separate direct memory access (DMA) CPU (**72**) in a single integrated circuit making up the microprocessor (**50**). The main CPU (**70**) has a first 16 deep push down stack (**74**), which has a top item register (**76**) and a next item register (**78**), respectively connected to provide inputs to an arithmetic logic unit (ALU) (**80**) by lines (**82**) and (**84**). An output of the ALU (**80**) is connected to the top item register (**76**) by line (**86**). The output of the top item register (**82**) is also connected by line (**88**) to an internal data bus (**90**). A loop counter (**92**) is connected to a decrementer (**94**) by lines (**96**) and (**98**). The loop counter (**92**) is birectionally connected to the internal data bus (**90**) by line (**100**). Stack pointer (**102**), return stack pointer (**104**), mode register (**106**) and instruction register (**108**) are also connected to the internal data bus (**90**) by lines (**110**), (**112**), (**114**) and (**116**), respectively. The internal data bus (**90**) is connected to memory controller (**118**) and to gate (**120**). The gate (**120**) provides inputs on lines (**122**), (**124**), and (**126**) to X register (**128**), program counter (**130**) and Y register (**132**) of return push down stack (**134**). The X register (**128**), program counter (**130**) and Y register (**132**) provide outputs to internal address bus (**136**) on lines (**138**), (**140**) and (**142**). The internal address bus provides inputs to the memory controller (**118**) and to an incremeter (**144**). The incrementer (**144**) provides inputs to the X register, program counter and Y register via lines (**146**), (**122**), (**124**) and (**126**). The DMA CPU (**72**) provides inputs to the memory controller (**118**) on line (**148**). The memory controller (**118**) is connected to RAM by address/data bus (**150**) and control lines (**152**).



U.S. PATENT DOCUMENTS

3,919,695 A 11/1975 Gooding
3,924,245 A 12/1975 Eaton et al.
3,930,688 A 1/1976 Rau et al.
3,967,104 A 6/1976 Brantingham et al.
3,968,501 A 7/1976 Gilbert
3,969,706 A 7/1976 Proebsting et al.
3,976,977 A 8/1976 Porter et al.
3,980,993 A 9/1976 Bredart et al.
3,988,717 A 10/1976 Kisylia
4,003,028 A 1/1977 Bennett et al.
4,003,033 A 1/1977 O'Keefe et al.
4,016,545 A 4/1977 Lipovski
4,037,090 A 7/1977 Raymond, Jr.
4,042,972 A 8/1977 Gruner et al.
4,050,058 A 9/1977 Garlic
4,050,096 A 9/1977 Bennett et al.
4,050,297 A 9/1977 Pettingell et al.
4,067,058 A 1/1978 Brandstaetter et al.
4,067,059 A 1/1978 Derchak
4,075,691 A 2/1978 Davis et al.
4,079,338 A 3/1978 Kronlage
4,079,455 A 3/1978 Ozga
4,107,773 A 8/1978 Gilbreath et al.
4,110,822 A 8/1978 Porter et al.
4,112,490 A 9/1978 Pohlman et al.
4,125,871 A 11/1978 Martin
4,128,873 A 12/1978 Lamiaux
4,144,562 A 3/1979 Cooper
4,181,938 A 1/1980 Suzuki et al.
4,215,401 A 7/1980 Holsztynski et al.
4,217,637 A 8/1980 Faulkner et al.
4,217,652 A 8/1980 Klaus et al.
4,223,380 A 9/1980 Antonaccio et al.
4,223,880 A 9/1980 Brems
4,224,676 A 9/1980 Appelt
4,236,152 A 11/1980 Masuzawa et al.
4,240,137 A 12/1980 Matsumoto et al.
4,242,735 A 12/1980 Sexton
4,253,785 A 3/1981 Bronstein
4,255,785 A 3/1981 Chamberlin
4,292,668 A 9/1981 Miller et al.
4,295,193 A 10/1981 Pomerene
4,305,045 A 12/1981 Metz et al.
4,315,305 A 2/1982 Siemon
4,315,308 A 2/1982 Jackson
4,317,227 A 2/1982 Skerlos
4,320,467 A 3/1982 Glass
4,321,706 A 3/1982 Craft
4,328,557 A 5/1982 Gastinel
4,334,268 A 6/1982 Boney et al.
4,335,447 A 6/1982 Jerrim
4,338,675 A 7/1982 Palmer et al.
4,348,720 A 9/1982 Blahut et al.
4,348,743 A 9/1982 Dozier
4,354,228 A 10/1982 Moore et al.
4,358,728 A 11/1982 Hashimoto
4,361,869 A 11/1982 Johnson et al.
4,364,112 A 12/1982 Onodera et al.
4,376,977 A 3/1983 Bruinshorst
4,382,279 A 5/1983 Ugon
4,390,946 A 6/1983 Lane
4,396,979 A 8/1983 Mor et al.
4,398,263 A 8/1983 Ito
4,398,265 A 8/1983 Puhl et al.
4,402,042 A 8/1983 Guttag
4,403,303 A 9/1983 Howes et al.
4,412,283 A 10/1983 Mor et al.
4,425,628 A 1/1984 Bedard et al.
4,449,201 A 5/1984 Clark
4,450,519 A 5/1984 Guttag et al.
4,453,229 A 6/1984 Schaire
4,462,073 A 7/1984 Grondalski
4,463,421 A 7/1984 Laws
4,467,420 A 8/1984 Murakami et al.
4,467,444 A 8/1984 Harmon, Jr. et al.
4,467,810 A 8/1984 Vollmann
4,471,426 A 9/1984 McDonough
4,472,789 A 9/1984 Sibley
4,488,217 A 12/1984 Binder et al.
4,488,227 A 12/1984 Miu et al.
4,491,938 A 1/1985 Leach
4,494,021 A 1/1985 Bell et al.
4,494,187 A 1/1985 Simpson
4,503,500 A 3/1985 Magar
4,509,115 A 4/1985 Manton et al.
4,538,239 A 8/1985 Magar
4,539,655 A 9/1985 Trussell et al.
4,541,045 A 9/1985 Kromer, III
4,541,111 A 9/1985 Takashima et al.
4,553,201 A 11/1985 Pollack, Jr.
4,556,063 A 12/1985 Thompson et al.
4,558,176 A 12/1985 Arnold et al.
4,562,537 A 12/1985 Barnett et al.
4,566,063 A 1/1986 Zolnowsky et al.
4,571,709 A 2/1986 Skupnjak et al.
4,577,282 A 3/1986 Caudel et al.
4,586,127 A 4/1986 Horvath
4,607,332 A 8/1986 Goldberg
4,616,338 A 10/1986 Helen et al.
4,626,798 A 12/1986 Fried
4,626,985 A 12/1986 Briggs
4,626,988 A 12/1986 George
4,627,082 A 12/1986 Pelgrom et al.
4,630,195 A 12/1986 Hester et al.
4,630,934 A 12/1986 Arber
4,641,246 A 2/1987 Halbert et al.
4,649,471 A 3/1987 Briggs et al.
4,660,155 A 4/1987 Thaden et al.
4,660,180 A 4/1987 Tanimura et al.
4,665,495 A 5/1987 Thaden
4,670,837 A 6/1987 Sheets
4,679,166 A 7/1987 Berger et al.
4,680,698 A 7/1987 Edwards et al.
4,689,581 A 8/1987 Talbot
4,691,124 A 9/1987 Ledzius et al.
4,698,750 A 10/1987 Wilkie et al.
4,701,884 A 10/1987 Aoki et al.
4,704,678 A 11/1987 May
4,708,490 A 11/1987 Arber
4,709,329 A 11/1987 Hecker
4,710,648 A 12/1987 Hanamura et al.
4,713,749 A 12/1987 Magar et al.
4,714,994 A 12/1987 Oklobdzija et al.
4,718,081 A 1/1988 Brenig
4,720,812 A 1/1988 Kao et al.
4,724,517 A 2/1988 May
4,739,475 A 4/1988 Mensch, Jr.
4,750,111 A 6/1988 Crosby, Jr. et al.
4,758,948 A 7/1988 May et al.
4,760,521 A 7/1988 Rehwald et al.
4,761,763 A 8/1988 Hicks
4,763,297 A 8/1988 Uhlenhoff
4,766,567 A 8/1988 Kato
4,772,888 A 9/1988 Kimura
4,777,591 A 10/1988 Chang et al.
4,780,814 A 10/1988 Hayek
4,782,439 A 11/1988 Borkar et al.
4,783,734 A 11/1988 May et al.
4,783,764 A 11/1988 Tsuchiya et al.
4,787,032 A 11/1988 Culley
4,791,590 A 12/1988 Ku et al.

4,794,526	A		12/1988	May et al.
4,794,558	A		12/1988	Thompson
4,797,850	A		1/1989	Amitai
4,803,621	A		2/1989	Kelly
4,805,091	A		2/1989	Thiel et al.
4,809,169	A		2/1989	Sfarti et al.
4,809,269	A		2/1989	Gulick
4,811,208	A		3/1989	Myers et al.
4,816,989	A		3/1989	Finn et al.
4,816,996	A		3/1989	Hill et al.
4,819,151	A		4/1989	May
4,833,599	A		5/1989	Colwell et al.
4,835,733	A		5/1989	Powell
4,835,738	A		5/1989	Niehaus et al.
4,837,563	A		6/1989	Mansfield et al.
4,837,682	A		6/1989	Culler
4,847,752	A		7/1989	Akashi
4,847,757	A		7/1989	Smith
4,849,875	A		7/1989	Fairman et al.
4,853,841	A		8/1989	Richter
4,860,198	A		8/1989	Takenaka
4,868,735	A		9/1989	Moller et al.
4,870,562	A		9/1989	Kimoto et al.
4,872,003	A		10/1989	Yoshida
4,882,710	A		11/1989	Hashimoto et al.
4,885,785	A		12/1989	Reynolds et al.
4,890,225	A		12/1989	Ellis, Jr. et al.
4,899,275	A		2/1990	Sachs et al.
4,907,225	A		3/1990	Gulick et al.
4,910,703	A		3/1990	Ikeda et al.
4,912,632	A		3/1990	Gach et al.
4,914,578	A		4/1990	MacGregor et al.
4,924,384	A		5/1990	Hao et al.
4,926,323	A		5/1990	Baror et al.
4,931,748	A		6/1990	McDermott et al.
4,931,986	A		6/1990	Daniel et al.
4,933,835	A		6/1990	Sachs et al.
4,942,553	A		7/1990	Dalrymple et al.
4,956,811	A		9/1990	Kajigaya et al.
4,959,782	A		9/1990	Tulpule et al.
4,967,326	A		10/1990	May
4,967,352	A		10/1990	Keida et al.
4,967,398	A		10/1990	Jamoua et al.
4,969,091	A		11/1990	Muller
4,974,157	A		11/1990	Winfield et al.
4,979,102	A		12/1990	Tokuume
4,980,821	A		12/1990	Koopman et al.
4,984,176	A		1/1991	Van den Heuvel
4,988,892	A		1/1991	Needle
4,989,113	A	*	1/1991	Asal 710/22
4,989,133	A		1/1991	May et al.
4,989,135	A		1/1991	Miki
4,990,847	A		2/1991	Ishimaru et al.
5,008,816	A		4/1991	Fogg, Jr. et al.
5,013,985	A		5/1991	Itoh et al.
5,021,991	A		6/1991	MacGregor et al.
5,022,395	A		6/1991	Russie
5,023,689	A		6/1991	Sugawara
5,031,092	A		7/1991	Edwards et al.
5,036,300	A		7/1991	Nicolai
5,036,460	A		7/1991	Takahira et al.
5,047,921	A		9/1991	Kinter et al.
5,053,952	A		10/1991	Koopman, Jr. et al.
5,068,781	A		11/1991	Gillett, Jr. et al.
5,070,451	A		12/1991	Moore et al.
5,081,574	A		1/1992	Larsen et al.
5,091,846	A		2/1992	Sachs et al.
5,097,437	A		3/1992	Larson
5,103,499	A		4/1992	Miner et al.
5,107,457	A		4/1992	Hayes et al.
5,109,495	A		4/1992	Fite et al.
5,121,502	A		6/1992	Rau et al.
5,127,091	A		6/1992	Boufarah et al.
5,127,092	A		6/1992	Gupta et al.
5,133,064	A		7/1992	Hotta et al.
5,134,701	A		7/1992	Mueller et al.
5,146,592	A		9/1992	Pfeiffer et al.
5,148,385	A		9/1992	Frazier
5,157,772	A		10/1992	Watanabe
5,179,689	A		1/1993	Leach et al.
5,179,734	A		1/1993	Candy et al.
5,187,799	A		2/1993	McAuley et al.
5,226,147	A		7/1993	Fujishima et al.
5,237,699	A		8/1993	Little et al.
5,239,631	A		8/1993	Boury et al.
5,241,636	A		8/1993	Kohn
5,261,057	A		11/1993	Coyle et al.
5,261,082	A		11/1993	Ito et al.
5,261,109	A		11/1993	Cadambi et al.
5,325,513	A		6/1994	Tanaka et al.
5,339,448	A		8/1994	Tanaka et al.
5,353,417	A		10/1994	Fuoco et al.
5,353,427	A		10/1994	Fujishima et al.
5,379,438	A		1/1995	Bell et al.
5,410,654	A		4/1995	Foster et al.
5,410,682	A		4/1995	Sites et al.
5,414,862	A		5/1995	Suzuki et al.
5,421,000	A		5/1995	Fortino et al.
5,440,749	A		8/1995	Moore et al.
5,459,846	A		10/1995	Hyatt
5,511,209	A		4/1996	Mensch, Jr.
5,530,890	A		6/1996	Moore et al.
5,537,565	A		7/1996	Hyatt
5,604,915	A		2/1997	Moore et al.
5,659,703	A		8/1997	Moore et al.
5,809,336	A		9/1998	Moore et al.
5,874,584	A		2/1999	Wear et al.
6,598,148	B1		7/2003	Moore et al.

FOREIGN PATENT DOCUMENTS

EP	208287	A2	1/1987
EP	0 238 810		9/1987
EP	288649	A1	11/1988
EP	0 786 730		6/2000
GB	8233733		11/1982
JP	57-20979		2/1982
JP	57-196334		12/1982
JP	58025710	A	2/1983
JP	58-103043		6/1983
JP	61127228	A	6/1986
JP	61138356	A	6/1986
JP	62145413	A	6/1987
JP	63-026753		2/1988
JP	5189383	A	7/1993
WO	WO 81/00473		2/1981
WO	8803091	A1	5/1988
WO	WO 91/02311		2/1991

OTHER PUBLICATIONS

"IBM RT Personal Computer Technology," IBM Corp. 1986. (collection of papers by developers).

Acorn Computers, Ltd., Acorn RISC Machine CPU Software Manual, Issue 1.00 Oct. 1985.

Acorn's RISC leapfrog, Acorn User special issue, Jun. 1987; 59: 149–153.

Agrawal et al., "Design Considerations for a Bipolar Implementation of SPARC," Compcon Spring apos;88. Thirty–Third IEEE Computer Society International Conference, Digest of Papers, Feb. 29–Mar. 3, 1988, pp. 6–9.

Agrawal, "An 80 MHz Bipolar ECL Implementation of SPARC," Sun Microsystems, Inc., Jun. 25, 1989, 40 pages total.

Alliacense Product Report—Preliminary Review, USP 5,440,749; GPS Car Navigation Main Processor.

Alliacense Product Report—Preliminary Review, USP 5,440,749; GPS Navigation System GPS Chipset.

Alliacense Product Report—Preliminary Review, USP 5,440,749; GPS Navigation System Main Microprocessor.

Alliacense Product Report—Preliminary Review, USP 5,440,749; Kyocera Digital Camera Finecam S3R Image Processor.

Alliacense Product Report—Preliminary Review, USP 5,440,749; PDA/Mobile Navigation, GPS Processor.

Alliacense U.S. Patent No. 5,784,584 Product Report, NEC Microcomputer, V850E2 32 Bit Microcontroller, pp. 1–8 (2006).

Alliacense U.S. Patent No. 5,784,584 Product Report, TLCS–900/H1 Series TMP92C820, Toshiba Microcontroller TMP92C820 / TCLS–900/H1 Series 16–bit Microcontroller, pp. 1–9 (2006).

Alliacense U.S. Patent No. 5,809,336 Product Report, Toshiba Microcontroller TMP93CS44/S45 / TLCS–900/L Series 16–bit Microcontroller (2006).

Alliacense U.S. Patent No. 5,809,336 Product Report, NEC Microcontroller UDP789478, 8 Bit Microcontroller, 38 pages (2006).

Alliacense U.S. Patent No. 5,809,336 Product Report, Toshiba Microcontroller TMP92CZ26 / TMP92CW26, 32 bit Microcontroller (2006).

Alliacense U.S. Patent No. 5,809,336 Product Report, Toshiba MPEG–4 Audiovisual LSI TC35273 MPEG–4 Audiovisual Code LSI (2006).

Anderson, D.W., The IBM System/360 Model 91: Machine Philosophy and Instruction Handling, IBM Journal, IBM, Jan. 1967, pp. 8–24.

ARM Datasheet, Part No. 1 85250 0360 0, Issue No. 1 (Mar. 17, 1987).

ATMEL SPARC Instruction Set, "Assembly Language Syntax," Rev. 4168C–AERO–08/01, 2002.

Bagula, "A 5V Self–Adaptive Microcomputer with 16Kb of #2 Program Storage and Security," IEEE International Solid–State Circuit Conference, 1983, pp. 34–35.

Bayko, Great Microprocessors of the Past and Present (V 11.7.0), downloaded from: <<http://web.archive.org/web/20010107210400/http://bwrc.eecs.berkeley.edu/CIC/Archive/cup_history.html>>, Feb. 2007, 60 pages total.

Bit SPARC Integer Unit B5000 Datasheet attached to a presentation by Anant Agrawal in 1989, 28 pages.

Books Review: Operating Systems A Systematic View, William S. Davis, Addison–Wesley Publishing Company, Inc., 1987; 26(4):453–454.

Bosshart et al., "A 533K–Transistor LISP Processor Chip," IEEE Journal of Solid State Circuits, SC–22(5): 808–819 (Oct. 1987).

Bourke, "Character Synchronization During Overrun Conditions," Delphion, IBM Technical Disclosure Bulletin, Dec. 1977.

Cal Run Fortran Guide, University of California, Computer Center, Berkeley, 292 pages total. (Sep. 1974).

CDC 6000 Computer Systems—COBOL Instant 6000, Version 3; Control Data Publication No. 60327600A (Apr. 1971).

CDC 6000 Computer Systems, 7600 Computer Systems: Fortran Extended Instant 6000 Version 3, 7600 Version 1; Control Publication No. 60305900A (May 1971).

CDC 6000 Computer Systems/ 7600 Computer Systems: Fortran Extended Reference Manual, 6000 Version 3, 7600 Version 1; Control Data Publication No. 60329100, Revision D (1972).

CDC 6000 Series Computer System, 6642 Distributive Data Path Operation and Programming Reference Manual; Control Data Publication No. 60376300C (Mar. 1979).

CDC 6000 Series Computer Systems—Chippewa Operating System Documentation, vol. I, Preliminary Edition (updated May 1966).

CDC 6000 Series Computer Systems—Chippewa Operating System Documentation, vol. II, Preliminary Edition, Peripheral Packages and Overlays (Oct. 1965).

CDC 6000 Series Computer Systems—Chippewa Operating System Documentation, vol. III, Preliminary Edition, DSD—The Systems Display, (Nov. 1965).

CDC 6000 Series Computer Systems Ascent General Information Manual; Control Data Publication No. 60135400 (Feb. 1966).

CDC 6000 Series Computer Systems Chippewa Operating System Reference Manual; Control Data Publication No. 60134400 (Dec. 1965).

CDC 6000 Series Computer Systems Hardware Reference Manual; Control Data Publication No. 60100000 (Aug. 1978).

CDC 6000 Series Computer Systems/ 7600 Computer System: 6000 Compass Version 2, 7600 Versions 1 &2 Reference Manual; Control Data Publication No. 60279900, Revision D. (1970, 1971, 1972).

CDC 6000 Series Computer Systems: Chippewa Operating System Fortran Reference Manual; Control Data Publication No. 60132700A (May 1966).

CDC 6000 Series Computer Systems: Peripheral Equipment Reference Manual; Control Data Publication No. 60156100, Revision J (Mar. 1970).

CDC 6000 Series Computer Systems: Site Preparation and Installation Manual; Control Data Publication No. 60142400, Revision B (Sep. 1965).

CDC 6000 Series Computer Systems: FORTRAN Extended General Information, Control Data Publication No. 60176400 (Oct. 1966).

CDC 6000 Series FORTRAN Extended 4.0, Internal Maintenance Specifications, (1971).

CDC 6000 Series Introduction and Peripheral Processors Training Manual, Second Edition; Control Data Publication No. 60250400 (Nov. 1968).

CDC 6400 Central Processor; Control Data Publication No. 60257200 (Feb. 1967).

CDC 6400/6500/6600 Ascent–To–Compass Translator; Control Data Publication No. 60191000 (Mar. 1967).

CDC 6400/6500/6600 Computer System Input/Output Specification; Control Data Publication No. 60045100 (Sep. 1967).

CDC 6400/6500/6600 Computer System Instant SMM; Control Data Publication No. 60299500 (Nov. 1969).

CDC 6400/6500/6600 Computer Systems Compass Reference Manual; Data 60190900, Revision B (Mar. 1969).

CDC 6400/6500/6600 Computer Systems Reference Manual; Control Data Publication No. 60100000, Revision K (Aug. 1970).

CDC 6400/6500/6600 Computer Systems Reference Manual; Control Data Publication No. 60100000D (1965, 1966, 1967).
CDC 6400/6500/6600 Extended Core Storage Systems Reference Manual; Control Data Publication No. 60225100, (Feb. 1968).
CDC 6400/6500/6600 Jovial General Information Manual; Control Data Publication No. 60252100A (Mar. 1969).
CDC 6400/6600 Computer Systems: Ascent/Asper Reference Manual; Control Data Publication No. 60172700 (Jul. 1966).
CDC 6400/6600 Fortan Conversion Guide; Data Publication No. 60175500 (Aug. 1966).
CDC 6400/6600 Systems Bulletin (Oct. 10, 1966), 84 pages.
CDC 6400/6600, Export/Import 8231 Reference Manual; Data Publication No. 60189100 (Apr. 1967).
CDC 6600 Central Processor vol. 1; Control &Memory; Data Control Publication No. 020167 (Mar. 1967).
CDC 6600 Central Processor, vol. 2; Functional Units; Control Data Publication No. 60239700 (Mar. 1967).
CDC 6600 Chassis Tabs; Control Data Publication No. 63016700A (Apr. 1965).
CDC 6600 Chassis Tabs; Control Data Publication No. 63019800 (Mar. 1965).
CDC 6600 Chippewa Operating System; Control Data Publication No. 60124500, Revision C (Apr. 1965).
CDC 6600 Computer System 6601 A–J, 6613A/B/C, 6604A/B/C, 6614–A/B/C Central Processor (Including Functional Units) vol. 1, Diagrams & Circuit Description; Control Data Publication No. 60119300BT(Jan. 1968).
CDC 6600 Computer System 6601 A–J, 6613A/B/C, 6604A/B/C, 6614–A/B/C Peripheral and Control MW Processor/Central Memory/Clock/Extended Core Storage Coupler (Std Opt 10102 and/or Spec Opt 60080)/ Power Wiring, vol. 2, Diagrams & Circuit Descriptions; Control Data Publication No. 60119300BT (Jan. 1968).
CDC 6600 Computer System Operating System/Reference Manual, SIPROS66, 1st ed.; Control Data Publication No. 60101800A (1965).
CDC 6600 Computer System Programming System/Reference Manual, vol. 1. Ascent; Control Data Publication No. 60101600B (1965).
CDC 6600 Computer System Programming System/Reference Manual, vol. 2. Asper; Control Data Publication No. 60101700B (1965).
CDC 6600 Computer System Programming vol. 3, Fortran 66; Control Data Publication No. 60101500B (1965).
CDC 6600 Computer Training Manual vol. 2, Preliminary Edition, Section 7.2 Shift Functional Units, 164 pages.
CDC 6600 Data Channel Equipment 6602–B/6612–A, 6603–B, 6622–A, 6681–B, 6682–A/6683–A, S. O. 60022, 60028,60029, Diagrams & Circuit Description' Control Data Publication No. 60125000, Revision G (Jun. 1966).
CDC 6600 Training Manual; Control Data Publication No. 60147400, Revision A (Jun. 1965).
CDC 6603—A/B/C Disk File Controller Reference Manual; Control Data Publication No. 60334000 (1970).
CDC 6638 Disk File Subsystem Training Supplement; Control Data Publication No. 602500800A (Oct. 1968).
CDC 6638 Disk File System: Standard Option 10037–A, 6639–A/B File Controller—Diagrams and Circuit Description/Maintenance/Maintenance Aids/Parts List/Wire Lists/ Chassis Tabs; Control Data Publication No. 60227300, Revision H (Mar. 1974).
CDC 6639—A/B Disk File Controller Reference Manual; Control Data Publication No. 60334100E (Aug. 1973).
CDC 6639 Disk Controller Training Manual Test Edition (Sep. 1967), 28 pages.
CDC APL Version 2 Reference Manual, CDC Operating Systems: NOS; Control Data Publication No. 60454000F (Nov. 1980).
CDC Basic Version 3 Reference Manual: NOS 1 &NOS/BE 1; Control Data Publication No. 19983900G (Oct. 1980).
CDC Chippewa Operating System, Document Class ERS, System No. E012, Version 1.1; External Reference Specifications—Chippewa Operating System, (Jun. 1966).
CDC Chippewa Operating System, Document Class SIR, System No. E012, Version 1.0, (Mar. 3, 1966).
CDC Chippewa Operating System, Document Class SIR, System No. E012, Version 1.1, (Mar. 3, 1966).
CDC Cobol Version 4 Instant Manual, Cyber 170 Series, Cyber 70 Models 72, 73, 74, 6000 Series; Control Data Publication No. 60497000A (Feb. 1976).
CDC Cobol Version 5 Instant Manual,Operating Systems: NOS 1/BE 1; Control Data Publication No. 60497300B (Feb. 1981).
CDC Codes/Control Data 6400/6500/6600 Computer Systems; Control Data Publication No. 60141900, Revision C (Jun. 15, 1967).
CDC Codes/Control Data 6400/6500/6600/6700 Computer Systems; Control Data Publication No. 60141900, Revision D (Jun. 10, 1970).
CDC Codes/Control Data 6600; Control Data Publication No. 60141900, Revision A (May 1965).
CDC Compass Version3 Instant, Operating Systems: NOS 1, NOS 2, NOS/ BE 1, Scope 2; Control Data Publication No. 60492800D (Jun. 1982).
CDC Course No. FH4010–1C, NOS Analysis, Student Handout, Revision C (Apr. 1980).
CDC Course No. FH4010–4C NOS Analysis, Study Dump (Apr. 1980).
CDC Cyber 170 Models 720, 730, 740, 750 and 760 Model 176 (Level B/C) Computer Systems, Codes; Control Data Publication No. 60456920C, (Jun. 1981).
CDC Cyber 70 Computer Systems Models 72,73,74,6000 Computer Systems: Fortran Reference Manual Models 72,73,74 Version 2.3; 6000 Version 2.3; Control Data Publication No. 19980400, Revision F (Jul. 1972).
CDC Cyber 70 Computer Systems Models 72,73,74,76,7600 Computer System, 6000 Computer Systems—ALGOL Reference Manual, Cyber 70 Series Version 4, 6000 Series Version 4, 7600 Series Version 4; Control Data Publication No. 60384700A (Aug. 1973).
CDC Cyber 70 Computer Systems Models 72,73,74,76,7600 Computer System, 6000 Computer Systems: Cobol Instant Models 72, 73, 74 Version 4, Model 76 Version 1,6000 Version 4; Control Data Publication No. 60328400A (Dec. 1971).
CDC Cyber 70 Computer Systems Models 72,73,74,76,7600 Computer System, 6000 Computer OA Systems: Fortran Extended Instant Models 72, 73, 74 Version 4, Model 76 Version 2,7600 Version 2; Control Data Publication No. 60357900A (Nov. 1971).
CDC Cyber 70 Computer Systems Models 72,73,74,76,7600 Computer System, 6000 Computer Systems: Fortran Extended Reference Manual Models 72, 73, 74 Version 4, Model 76 Version 2,6000 Version 4; Control Data Publication No. 60305600A (Oct. 1971).

CDC Cyber 70 Series 6000 Series Computer Systems: APL *Cyber Reference Manual; Control Data Publication No. 19980400B (Jul. 1973).

CDC Cyber 70 Series Computer Systems Models 72,73,74, 6000 Series Computer Systems—Kornos 2.1 Workshop Reference Manual; Control Data Publication No. 974047000 (1976).

CDC Cyber 70 Series Models 72/73/74,6000 Series Computer Systems, Krono 2.1 Operator Guide; Control Data Publication Guide Control Data Publication No. 60407700A (Jun. 1973).

CDC Cyber 70 Series Models 72/73/74,6000 Series Computer Systems, Kronos 2.1 Installation Handbook; Control Data Publication No. 60407500A (Jun. 1973).

CDC Cyber 70 Series Models 72/73/74, 6000 Series Computer Systems, Kronos 2.1 Time–Sharing User's Reference Manual; Control Data Publication No. 60407600B (May 1974).

CDC Cyber 701 Model 76 Computer System, 7600 Computer System: Fortran Run, Version 2 Reference Manual; Control Data Publication No. 60360700C (May 1974).

CDC Cyber Interactive Debug Version 1 Guide for Users of Fortran Extended Version 4, CDC Operating Systems: NOS 1, NOS/BE 1, Control Data Publication No. 60482700A (Feb. 1979).

CDC Cyber Interactive Debug Version 1 Guide for Users of Fortran Version 5, Operating Systems: NOS 1, NOS / BE 1, Control Data Publication No. 60484100C (Sep. 1984).

CDC Cyber Interactive Debug Version 1 Reference Manual, NOS 1, NOS 2, NOS/ BE 1, Data Control Publication No. 60481400D (Jun. 1984).

CDC Cyber Loader Version 1 Instant, CDC Operating System NOS1, NOS/BE 1, Control Data Publication No. 60449800C (Aug. 1979).

CDC Disk Storage Subsystem—Operation and Programming Manual; Control Data Publication No. 60363900, Version T (1972–1980).

CDC Fortran Extended 2.0, Document Class ERS, System No. C012, (Dec. 1966).

CDC Fortran Extended 2.0, Document Class IMS, Internal Maintenance Specifications—64/65/6600 V Fortran Extended Version 2 (Mar. 1969).

CDC Fortran Extended Version 4 Instant Manual, Operating Systems: NOS 1, NOS/BE 1, Scope 2; Control Data Publication No. 60497900B (Jun. 1981).

CDC Fortran Extended, Sales Technical Memorandum (May 1967).

CDC Fortran Extended Version 5 Instant, CDC Operating Systems: NOS 1, NOS/BE 1, Scope 2; Control Data Publication No. 60483900A (Jan. 1981).

CDC GED Fortran Extended 1.0, Product No. C012, Dept. No. 254, Project No. 4P63FTN (Aug. 1967).

CDC Instant 6400/3500/6500 Simula; Control Data Publication No. 60235100, Revision A (Feb. 1969).

CDC Instant 6400/6500/6600 Compass; Control Data Publication No. 60191900, Revision A (1968).

CDC Instant Fortran 2.3 (6000 Series); Data Publication No. 60189500D (May 1969).

CDC Internal Maintenance Specification: Fortran V5, Part 1; Control Data Publication No. 77987506A.

CDC Internal Maintenance Specification: Fortran V5, Part 2; Control Data Publication No. 77987506A.

CDC Kronos 2.1 Reference Manual vol. 1 of 2; Control Data Cyber 70 Series Models 72/76/74, 6000 Series Computer Systems; Control Data Publication No. 60407000D (Jun. 1975).

CDC Model dd 60A Computer Control Console/Customer Engineering Manual; Control Data Publication No. 82100010 (Mar. 1965).

CDC Model dd60b Computer Control Console/Customer Engineering Manual; Control Data Publication No. 82103500 (Feb. 1967).

CDC Network Products: Network Access Method Version 1, Network Definition Language Reference Manual; Control Data Publication No. 60480000J (May 1981).

CDC Network Products: Network Terminal User's Instant—Operating System NOS 1; Control Data Publication No. 60455270C (Oct. 1980).

CDC NOS 2 Operations Handbook; Control Data Publication No. 60459310, (Aug. 1994).

CDC NOS Version 1 Applications Programmer's Instant, CDC Computer Systems: Cyber 170 Series, Cyber 70 Models 71,72,73,74,6000 Series; Control Data Publication No. 60436000H (Jan. 1980).

CDC NOS Version 1 Internal Maintenance Specification vol. 1 of 3; Control Data Publication No. 60454300B (Aug. 1979).

CDC NOS Version 1 Internal Maintenance Specification vol. 2 of 3; Control Data Publication No. 60454300B (Aug. 1979).

CDC NOS Version 1 Internal Mainenance Specification vol. 3 of 3; Control Data Publication No. 60454300B (Aug. 1979).

CDC NOS Version 1 Operator's Guide, CDC Computer Systems: Cyber 170 Series, Cyber 70 Model 71, 72,73,74,6000 Series (Dec. 1980).

CDC NOS Version 1 Reference Manual vol. 1 of 2, CDC Computer Systems: Cyber 170 Series, Cyber 70 Models 71,72,73,74,6000 Series; Control Data Publication No. 60435400J (1979).

CDC NOS Version 1 Reference Manual vol. 2 of 2, CDC Computer Systems: Cyber 170 Series, Cyber 70 Models 71, 72, 73, 74,6000 Series; Control Data Publication No. 60445300E (1977).

CDC NOS Version 1 System Maintenance Reference Manual; Control Data Publication No. 60455380H (Apr. 1981).

CDC NOS Version 2 Analysis Handbook, Cyber 180, Cyber 170, Cyber 70 Models 71, 72, 73, 74, 6000; Control Data Publication No. 60459300D (Oct. 1984).

CDC NOS Version 2 Analysis Handbook, Control Data Publication No. 60459300U (Jul. 1994).

CDC NOS Version 2 Installation Handbook, Cyber 180, Cyber 170, Cyber 70 Models 71, 72, 73, 74, 6000; Control Data Publication No. 60459320E_(Mar. 1985).

CDC NOS Version 2 Operation/Analysis Handbook, Cyber 170, Cyber 70 Models 71, 72, 73, 74, 6000, Control Data Publication No. 60459310C (Oct. 1983).

CDC NOS Version 2 System Maintenance Reference Manual, Cyber 170, Cyber 70 Models 71, 72, 73, 74, 6000; Control Data Publication No. 60459300C (Oct. 1983).

CDC NOS/BE Version 1 Diagnostic Handbook, Cyber 180, Cyber 170, Cyber 70 Models 71,72,73,74, 6000; Control Data Publication No. 60494400–V (1986).

CDC NOS/BE Version 1 Installation Handbook, Cyber 180, Cyber 170, Cyber 70 Models 71,72,73,74, 6000; Control Data Publication No. 60494300aB (Dec. 1986).
CDC NOS/BE Version 1 Reference Manual, Cyber 170 Series, Cyber 70 Models 71, 72, 73, 74, 6000 Series; Control Data Publication No. 60493800M (1981).
CDC Outline of Reports on Feasibility Study of 64/6600 Fortran Ver 3.0 and Conversational Fortran, Fortran Study Project, Product No. X010, Dept No. 254, Project No. 4P63, (Jun. 1966).
CDC Pascal Version 1 Reference Manual, Operating Systems: NOS 2; Control Data Publication No. 60497700 (Sep. 1983).
CDC Pascal Version 1 Reference Manual, Operating Systems: NOS 2; Control Data Publication No. 60497700A (Dec. 1982).
CDC PL/1 Version 1 Instant, CDC Operating Systems: NOS 1, NOS/ BE 1; Control Data Publication No. 60483700A (Nov. 1979).
CDC Simscript 11.5 Instant; Control Data Publication No. 84000450B (Sep. 1978).
CDC Sort/Merge Version 4 and 1 Instant, Operating Systems: NOS 1, NOS/BE 1, Scope 2; Control Data Publication No. 60497600C (Jan. 1981).
CDC Sort/Merge Version 5 Reference Manual, Operating Systems: NOS 2, NOS/BE 1, Control Data Publication No. 60484800C (Feb. 1984).
CDC SYMPL Version 1 Instant, NOS 1, NOS/BE 1, Scope 2; Control Data Publication No. 60482600A (May 1978).
CDC SYMPL Version 1 Users Guide, Operating Systems: NOS 1, NOS/BE 1, Scope 2; Control Data Publication No. 60499800B (Apr. 1978).
CDC Update Instant, Cyber 170 Series, Cyber 70 Series, 6000 Series, 7600 Computer Systems; Control Data Publication No. 60450000A (Nov. 1975).
CDC Update Reference Manual Operating Systems: Scope 3.4, Kronos 2.1; Control Data Publication No. 60342500, Revision H (1971–1976).
CDC Xedit Version 3 Reference Manual, Operating System: NOS 1; Control Data Publication No. 60455730B (Aug. 1979).
Chippewa Laboratories Fortran Compiler Run, Preliminary Edition, CDC 6000 Series Computer Systems, (Apr. 1966).
Cho et al., WAM 3.6: A 40K Cache Memory and Memory Management Unit, ISSCC '86, Feb. 19, 1986.
Cho et al., "The Memory Architecture and the Cache and Memory Management Unit for the Fairchild Clipper Processor," Report No. UCB/CSD 86/289, Computer Science Division (EECS), University of California (Apr. 1986).
CLIPPERTM 32–Bit Microprocessor, Introduction to the Clipper Architecture, published by Fairchild in 1986.
Cordell, II et al., "Advanced Interactive Executive Program Development Environment," IBM Systems Journal, 1987; 26(4):361–382.
Crawford, "The i486 Executing Instructions in One Clock Cycle," IEEE Micro, pp. 28–36 (Feb. 1990).
Cray–1 Computer System Hardware Reference Manual, Publication No. 2240004, Rev C, Nov. 4, 1977.
Disk Routines and Overlays, Chippewa Operating System, CDC Development Division—Applications, (Nov. 1965).
Ditzel et al., "The Hardware Architecture of the Crisp Microprocessor," AT &T Information Systems, ACM, pp. 309–319 and table of contents (1987).
DS5000 Soft Microcontroller User's Guide Preliminary V 1.0, Dallas Semiconductor.
Duell. C. H., "Everything that can be invented has been invented," 2 pages downloaded from http://www.tplgroup.net/patents/index.php.
Evans et al., "An Experimental 512–bit Nonvolatile Memory with Ferroelectric Storage Cell," IEEE Journal of Solid–Sate Circuits, 23(5):1171–1175.
Excerpt from A Seymour Cray Perspective http//research.microsoft.com/users/gbell/craytalk/sld029.htm (Slide 29).
Excerpts from A Seymour Cray Perspective http//research.Microsoft.com/users/gbell/craytalk/sld001.htm (Slide 1).
Fiasconaro, J., "Microarchitecture of the HP9000 Series 500 CPU," Microarchitecture of VLSI Computers, NATO ASI Series No. 96, Antognetti, eds., pp. 55–81.
Field Maintenance Print Set, KA780–01–01 Rev. A.
Fisher et al., "Very Long Instruction Word Architectures and the ELI–512," ACM pp. 140–150 (1983).
Fukui et al., "High Speed CMOS 4–bit Microcomputer SM550 Series," pp. 107–109 published 1982, 1983. (Document in Japanese).
Furber, VSLI RISC Architecture and Organization, Chapter 3: Commercial VLSI RISC, pp. 124–129, Marcel Dekker, Inc., 1989.
GE 600 Series, publication.
GE–625 / 635 Programming Reference Manual, revised Jan. 1996.
Gershon, Preface, IBM Systems Journal 26(4):324–325.
Green et al., "A Perspective on Advanced Peer–to–Peer Networking," IBM Systems Journal, 1987; 26(4):414–428.
Grimes et al., "64 bit Processor, The Intel i860 64–Bit Processor: A General–Purpose CPU with 3D Graphics Capabilities", published by Intel, p. 85 (Jul. 1989).
Grishman, R., "Assembly Language Programming for the Control Data 6000 and Cyber Series Algorithmics".
Grondalski et al., "Microprocessors–Special Purpose—THPM 16.3: A VLSI Chip Set for a Massively Parallel Architecture," 1987 IEEE International Solid–State Circuits Conference, Feb. 26, 1987, pp. 1998–198.
Gross et al., "Measurement and evaluation of MIPS architecture and processor," ACM Trans. Computer Systems, pp. 229–257 Aug. 1988.
Guttag, "The TMS34010: An Embedded Microprocessor", IEEE Micro, vol. 8, No. 3, May 1988, pp. 39–52.
Hansen, "A RISC Microprocessor with Integral MMU and Cache Interface," ICCD '86, pp. 145–148, 1986.
Hennessy et al., "Hardware/software tradeoff for increased performance," Proceedings of the Symposium on Architectural Support for Programming Languages and Operating Systems pp. 2–11. ACM, Apr. 1982.
Hennessy et al., "Hardware/software tradeoff for increased performance," Technical Report No. 22.8, Computer Systems Laboratory, Feb. 1983, 24 pages.
Hennessy et al., "MIPS: A Microprocessor Architecture," IEEE, pp. 17–22 (1982).
Hennessy et al., "MIPS: A VLSI Processor Architecture" VLSI Systems and Computer, Kung eds., Carnegi–Mellon University, pp. 337–346 (1981).
Hennessy et al., "The MIPS Machine", Compcon, IEEE, Spring 1982, pp. 2–7.
Hennessy, "Performance Issues in VLSI Processor Design," IEEE on VLSI in Computers, , pp. 153–156. (1983).
Hinton, 80960—Next Generation, Compcon Spring 89, IEEE, 13–16 (1989).

Hitachi America Ltd., "8–Bit Single–Chip Microprocessor Data Book", Jul. 1985, Table of Contents and pp. 251–279.
Hollingsworth et al., "The Fairchild Clipper: Instruction Set Architecture and Processor Implementation," Report No. UCB/CSD 87/329, Computer Science Division (EECS), University of California Berkeley, California, (Feb. 11, 1987).
Horowitz et al., "A 20—MIPS Peak, 32–bit Microprocessor with On–Chip Cache," IEEE Journal of Solid State Circuits, SC–22(5):790–799 (Oct. 1987).
HP 9000 Instrument Controllers, Technical Specifications Guide, Oct. 1989.
HP 9000 Series Computer Systems, HP–UX Reference 09000–090004, Preliminary Nov. 1982.
HP Sacajawea External Reference Specification Preliminary Version 1.1 (Jan. 14, 1987).
Hughes, "Off–Chip Module Clock Controller," Delphion, IBM Technical Disclosure Bulletin, Sep. 1989.
Hunter, "Introduction to the Clipper Architecture," IEEE Micro, pp. 6–26 (Aug. 1987).
IBM RT PC, Byte 1986 Extra Edition, Inside The IBM PCs, pp. 60–78.
IBM Systems Reference Library, IBM System/360 Model 67 Functional Characteristics, File No. S360–01, Form A27–2719–0, published by IBM (1967).
IEEE Std 796–1983, Microcomputer System Bus, pp. 9–46.
Index of/pdf/cdc/6x00, downloaded from http://www.bitsavers.org/pdf/cdc/6x00/.
INMOS Engineering Data, IMS T414M Transputer, Extended Temperature, (Aug. 1987).
INMOS IMS T212 Engineering Data Preliminary Data Sheet (Aug. 1987).
INMOS IMS T414 Data Sheet, (Jun. 1987).
INMOS IMS T414 Transputer, Engineering Data, pp. 107–163.
INMOS IMS T414 Transputer, Preliminary Data (Feb. 1987).
INMOS M212 Disk Processor Product Overview Oct. 1987, 12 pages total.
Intel 386TM DX Microprocessor 32–Bit CHMOS Microprocessor With Integrated Memory Management (1995).
Intel 4004 Data Sheet Single Chip 4–Bit 9–Channel Microprocessor, pp. 8–15 to 8–23.
Intel 8008 8–Bit Parallel Central Processor Unit, published by Intel (Nov. 1972), Users Manual.
Intel 80960CA User's Manual published by Intel (1989).
Intel Architecture Optimization Manual, Order No. 242816–003, published by Intel (1997).
Intel Architecture Software Developer's Manual, vol. 1: Basic Architecture, published by Intel (1997).
Intel i860 64–Bit Microprocessor, Intel Corporation Feb. 1989.
Intel MCS–4 Micro Computer Set, Integrated Circuit Engineering Collection (Nov. 1971).
Intel, iAPX 386 High Performance 32–Bit Microprocessor Product Review (Apr. 1984).
Intel 8080A/8080A–1/8080A–2, 8–Bit N–Channel Microprocessor, Order No. 231453–001, Its Respective Manufacturer (Nov. 1986).
Jguppi et al., "A 20 MIPS Sustained 32b CMOS with 64b Data Bus," IEEE Int'l Solid State Circuits Conf., pp. 84–86 (1989).
Johnson et al., "A Variable Delay Line PLL for CPU–Coprocessor Synchronization," IEEE Journal of Solid–State Circuits, 23(5): 1218–1223, Oct. 1988.
Katevenis et al., "Reduced Instruction Set Computer Architecture for VLSI," Doctoral Dessertation, Oct. 1983.
Katevenis et al., "The RISC II Micro–Architecture," Journal of VLSI and Computer Systems, 1(2): 138–152 (1984).
Kipp, "Micron Technology Inc. Reports Financial Results," Business Wire, New York, Sep. 26, 1988.
Kohn et al., "Introducing Intel i860 64–Bit Microprocessor," Intel Corporation, IEEE Micro (Aug. 1989).
Koopman, "RTX 4000," Proceedings of 1989 Rochester Forth Conference, pp. 84–86.
Koopman, "The WISC Concept: A proposal for a writable instruction set computer," Byte, pp. 187–193. (Apr. 1987).
Koopman, Jr. et al. "MVP Microcoded CPU/16 Architecture," Proceedings of 1986 Rochester Forth Conference, pp. 277–280.
Koopman, Jr. et al., "WISC Technologies, Inc., Writable Instruction Set, Stack Oriented Computers: The WISC Concept," 1987 Rochester Forth Conference, Journal of Forth Application and Research, 5(1):49–71.
"Stack Computers: the new wave, Chapter 5.2: Architecture of the FRISC 3 (SC32)", Philip Koopman, 1989, 9 pages.
Loucks et al., "Advanced Interactive Executive (AIX) Operating System Overview," IBM Systems Journal, 1987; 26(4):326–345.
LSI Logic Corporation MIPS Architecture RISC Technology Backgrounder, "Introduction to RISC Technology," LSI Logic Corporation (Apr. 1988).
Matick, "Self–Clocked Cache," Delphion, IBM Technical Disclosure Bulletin, Apr. 1985.
Matsushita Electric, 8 bit Dual1–chip Microcomputer MN1890 Series User's Manual, translation of original Japanese language document, by Matsushita Electric Industrial Co., Ltd. Semiconductor Sales Division.
Matsushita Electronics Corporation, MN1880 (MN18882) Instruction Manual, (document in Japanese), 1988.
Matsushita Electronics Corporation, MN188166 User's Manual, Japanese language document.
Matsushita Electronics Corporation, MN18882 LSI User's Manual, Japanese language document, 1987.
Matsushita Electronics Corporation, Specification Sheet, MN18882 (Book1) translation of the Japanese language original, Code No. MIG0175, Matsushita Electronics Industry Corporation, Microcomputer Products Division, Oct. 22, 1990.
Matthys R. J., Crystal Oscillator Circuits, John Wiley &Sons, pp. 25–64 (1983).
May, "The Transputer and Occam," International Conference on the Impact of Digital Microelectronics and Microprocessors on Particle Physics, held Mar. 28–30, 1988, published by World Scientific in 1988, Budnich, eds. pp. 205–211.
May, D., "The Influence of VLSI Technology on Computer Architecture," INMOS Ltd., pp. 247–256 (1988).
Mead et al., eds., Introduction to VLSI Systems, Addison Wesley Publishers, (1980), 144 pages.
Miller, Frequency Modulated Ring Oscillator for a Mode Regulated Substrate Bias Generator, Delphion, IBM Technical Disclosure Bulletin, Sep. 1989.
Mills et al., "Box Structured Information Systems," IBM Systems Journal, 1987; 26(4):395–413.

Minyard, Using a TMS320C30 Serial Port as an Asynchronous RS–232 Port, Application Brief: SPRA240, Texas Instruments (May 1994).
MMP Portfolio, News Release: Roland Becomes 50th Licensee, Setting a Major Milestone in Moore Microprocessor Patent Licensing Program, 3 pages (May 1, 2009).
Moelands, A. P. M., "Serial I/O with the MAB8400 series microcomputers," Electronic Components and Applications, 3(1):38–46 (1980).
Moore, P., "INMOS Technical Note 15: IMS B005 Design of a Disk Controller board with drives," Dec. 3, 1986.
Mostek Corp., Adverisement, EDN, Nov. 20, 1976.
Motorola Inc., MC 68332 32–Bit Microcontroller System Integration User's Manual Preliminary Edition, Revision 0.8, (1989).
Motorola MC146805H2, Advance Information, pp. 1–12.
Motorola MC68HC11A8 HCMOS Single–Chip Microcomputer, table of contents and introduction (1985).
Motorola Semiconductors MC146805H2, Product Brochure.
Motorola, "How to Take Control" product brochure by Motorola (1988).
Motorola, MC68300 Family MC68332 User's Manual, (1995).
Moussouris et al., "A CMOS RISC Processor Integrated System Functions," Proceedings of 31st IEEE Computer Society International Conference, Cathedral Hill Hotel, San Francisco, CA Mar. 3–6, 1986, pp. 126–131, 1986.
National Semiconductor HPC16400/HPC36400/HPC46400 High–Performance MicroControllers with HDLC Controller product literature.
NEC Data Sheet MOS Integrated Circuit uPD75008, 4 bit Single–Chip Microcomputer (1989).
NEC Electronics Inc. High–End, 8–Bit, Single–Chip CMOS Microcomputers product literature.
NEC Electronics Inc. Microcomputer Products Microprocessors, Peripherals, & DSP Products Data Book vol. 2 of 2 cover page.
NEC Electronics Inc. Microcomputer Products Single–Chip Products Data Book vol. 1 of 2 cover page.
NEC Electronics Inc. MOS Integrated Circuit uPD70208H, 70216H Data Sheet, V40HL, V50HL 16/8, 16–Bit Microprocessor (1995).
NEC Electronics Inc. MOS Integrated Circuit uPD7225 Programmable LCD Controller/Driver (1986, 1999).
NEC Electronics Inc. uPD78C10/C11/C14 8–Bit, Single–Chip CMOS Microcomputers with A/D Converter product literature.
Olson, Semiconductor Die with Wiring Skirt (Packaging Structure), Delphion, IBM Technical Disclosure Bulletin, Jul. 1978.
O'Neil, "Pipeline Memory System for Drams", Delphion, IBM Technical Disclosure Bulletin, May 1989.
Paker, Y, Multi–Processor Systems, Academic Press, pp. 1–23 (1983).
Patterson et al., "Architecture of a VLSI Instruction Cache for a RISC," ACM, pp. 108–116 (1983).
Patterson et al., "RISC I: A Reduced Instruction Set VLSI Computer," Proceedings of the 8th annual symposium on Computer Architecture on Computer Architecture, Minneapolis, Minnesota, pp. 443–457 (May 1981).
Patterson, "RISC Watch", ACM, vol. 12 (1):11–19 (Mar. 1984).
Patterson, D. A., "Reduced Instruction Set Computers" Communication of the ACM, 28(1):.8–21, Jan. 1985.
Pountain, "The Archimedes A310," Byte, 1987.
Przybyiski et al., "Organizational and VLSI Implementation of MIPS," Technical Report: CSL–TR–84–259, Apr. 1984.
Przybyiski, "The Design Verification and Testing of MIPS", 1984 Conference on Advanced Research in VLSI, pp. 100–109.
Rau et al., "The Cydra 5 Departmental Supercomputer, Design Philosophies, Decisions, and Tradeoffs," IEEE, pp. 12–36 (1989).
Reekie, Realtime DSP: The TMS320C30 Course, Revision 3 (Feb. 20, 1994).
RISC Roots: CDC 6000 (1965) www.//bwrc.eecs.berkeley.edu/CIC/archive/cpu_history.html, downloaded Oct. 27, 2006.
Roche et al., "Method of Assuming a Two–Cycle Start, Zero Cycle Stop, Non–Chopping on Chip Clock Control Throughout a VLSI Clock System," Delphion, IBM Technical Disclosure Bulletin, Sep. 1989.
Rowen et al., "A Pipelined 32b NMOS Microprocessors and Microcontrollers," IEEE International Solida–State Circuits Conference, pp. 180–181, 1984.
Rubinfeld et al., "The CVAX CPU, A CMOS VAX Microprocessor Chip", International Conference on Computer Design, Oct. 1987.
Ryan, D.P., "Intel's 80960: An Architecture Optimized for Embedded Control," IEEE Micro, published in Jun. 1988.
Sanamrad et al., "A Hardware Syntactic Analysis Processor," IEEE, Aug. 1987, pp. 73–80.
Sequin et al., "Design and Implementation of RISC I," pp. 276–298 from VLSI Architecture, B. Randell and P.C. Treleaven, editors, Prentice Hall, 1983.
Shepherd et al., "Current and Future Transputers," INMOS Presentation given at Jun. 15, 1988 Workshop on Computer Architecture.
Sherburne, R. W., "Processor Design Tradeoffs in VLSI," U.C. Berkeley, May 1984. PhD Dissertation.
Shih, "Microprogramming Heritage of RISC Design," Proceedings of the 23rd annual workshop and symposium on Microprogramming and microarchitecture, pp. 275–280. (1990).
Shyam, "Hardware External Reference Specification for Enhanced Champion/Paladin," Revision of Nov. 11, 1986.
Sibigtroth, J. M., "Motorola's MC68HC11: Definition and Design of a VLSI Microprocessor," IEEE Micro, 4(1):54–65 (1984).
Signetics Microprocessor Data manual cover page.
Signetics Microprocessor Products Data manual, 8X330 Floppy Disk Formatter/Controller product specification.
Signetics Microprocessor Products Data manual, SC96AH Series Single–Chip 16–Bit Microcontrollers preliminary specification.
Simpson et al., "The IBM RT PC ROMP Processor and Memory Management Unit Archiecture," IBM systems Journal, Dec. 1987; 26(4):346–360.
Simpson, R.O., "The IBM RT Personal Computer," Byte 11 (11):43–78 (Oct. 1986).
Skruhak et al., "Modular Design of a High Performance 32–bit Microcontroller," IEEE 1989 Custom Integrated Circuits Conference, pp. 23.8.1–23.8.4 (1989).
Stanley, R. C., "Microprocessors in brief," IBM J. Res. Develop., 29(2):110–118 (Mar. 1985).

Submicron Systems Architecture Project, Caltech Computer Science Technical Report, Nov. 1, 1991.
Sultan et al., "Implementing System–36 Advanced Peer–to–Peer Networking," IBM Systems Journal, 1987; 26(4):429–452.
Texas Instrument, "TMS 370 Microcontroller Family User's Guide," (1996), 873 pages.
Texas Instruments TMS320C30 Digital Signal Processor product literature, SPRS032A (Apr. 1996, Revised Jun. 1997).
Texas Instruments TMS34010 Graphics System Processor product literature.
The Ring Oscillator VCO Schematic, 1 page.
Thornton, J. E., "Considerations in Computer Design Leading Up To the Control Data 6600," Control Data Chippewa Laboratory (1970).
Thornton, J. E., "Design of a Computer, The Control Data 6600," published by Advanced Design Laboratory (1970).
Toshiba TLCS–42, 47, 470 User's Manual Published in Apr. 1986.
Ungar et al., "Architecture of SOAR: Smalltalk on a RISC," Proceedings of the 11th Annual International Symposium on Computer Architecture ISCA '84. ACM Press, New York, NY pp. 188–197 (1984).
VAX 11/780 Architecture Handbook vol. 1, 1977–1978, 2–7 and G–8.
VAX 8800 System Technical Description vol. 2, EK–KA881–TD–PRE, Section 6, Instruction Box (IBOX), Preliminary Edition (Jul. 1986).
VAX Maintenance Handbook: VAX–11/780, EK–VAXV2–HB–002, 1983 Edition.
Waters et al., "AIX Usability Enhancements and Human Factors," IBM Systems Journal, 1987; 26(4):383–394.
Whiteby–Streven, "Transputer Technical Notes from INMOS," Google Groups; comp.sys.transputer, dated Sep. 7, 1988.
Williams, "Chip Set Tackles Laptop Design Issues, Offers Flat–Panel VGA Control," Computer Design, Oct. 15, 1988; 27(19):21–22.
Agrawal, "Bipolar ECL Implementation," The SPARC Technical Papers, Catanzaro, eds., Springer–Verlag, NY, pp. 201–211. (1991).
Gill et al. Summary of MIPS Instruction. CSL Technical Note No. 237, Computer Systems Laboratory, Stanford University, Nov. 1983. 50 pages total.
Hennessy et al., "Design of a High Performance VSL Processor," Third Caltech Conference on Very Large Scale Integration, Bryant eds., California Institute of Technology, Computer Science Press, pp. 33–54. (1983).
Horowitz et al., "A 32b Microprocessor with On–Chip 2Kbyte Instruction Cache," IEEE International Solid State Circuits Conference, pp. 30, 31 and 328 (1987).
Knapp, "Frequency Stability Analysis of Transistorized Crystal Oscillator," IEEE Transactions on Instrumentation and Measurement, vol. 12, No. 1, pp. 2–5. (Jun. 1963).
Nicoud et al., "The Transputer Instruction Set," IEEE Micro, vol. 9, No. 3, pp. 60–75 (May 1989).
Parasuraman, "High Performance Microprocessor Architectures," Proceedings of the IEEE, vol. 64, No. 6, pp. 851–859. (Jun. 1976).
Walls et al., "A Frequency–Lock System for Improved Quartz Crystal Oscillator Performance," IEEE Transactions on Instrumentation and Measurement, vol. IM–27, No. 3, pp. 249–252 (Sep. 1978).
Acorn Computers Limited; "ARM Datasheet, Part No. 1 85250 0360 0"; Issue No. 1.0; Mar. 17, 1987.
Barron, I. et al.; "Transputer does 5 or more MIPS even when not used in parallel"; Electronics; McGraw–Hill; vol. 56, No. 23; Nov. 17, 1983; pp. 109–115.
Best et al.; "An Advanced–Architecture CMOS/SOS Microprocessor"; IEEE Micro; vol. 2, No. 3; Jul. 1982; pp. 10–26.
Burroughs Corp.; "Burroughs B5500 Information Processing Systems Reference Manual"; 1964.
Dowsing and Woodhams; "Computer Architecture, A First Course"; Van Nostrand Reinhold Co., Ltd. (UK); 1985; pp. 126–139.
Haley, A. et al.; "Forth as Machine Code"; Electronics & Wireless World; vol. 93; No. 1616; Jun. 1987; pp. 584–587.
IC Master; "F8/3870 F6800 Bit–Slice Faichild Microcomputers"; United Technical Publications; 1980; pp. 2016–2040.
INMOS Limited; "IMS T212 Transputer Data Sheet"; Aug. 1987.
INMOS Limited; "IMS T414 Transputer Data Sheet"; Feb. 1987.
INMOS Limited; "IMS T424 Transputer Reference Manual"; Nov. 1984; p. i–62.
INMOS Limited; "IMS T800 Transputer Data Sheet"; Apr. 1987.
Intel; "80386 Programmer's Reference Manual"; 1986.
Koopman, Jr., Philip; "Stack Computers, the new wave"; 1989.
Motorola; "Motorola MC68020 32–Bit Microprocessor User's Manual, Second Edition"; Prentice–Hall; 1985.
Proebsting et al.; "A TTL Compatible 4096–bit N–channel RAM"; Solid–State Circuits Conference; Digest of Technical Papers; 1973 IEEE International vol. XVI; Feb. 1973; pp. 28–29.
Schoeffler, J.; "Microprocessor Architecture"; Industrial Electronics and Control Instrumentation, IEEE Transactions on; vol. IECI–22, issue 3; Aug. 1975; pp. 256–272.
Whitby–Strevens, C.; "The Transputer"; The 12th Annual International Symposium on Computer Architecture, Conference Proceedings; Jun. 17–19, 1985; pp. 292–300.
VLSI Technologies, Inc.; "VL86C010 RISC Family Data Manual"; 1987.
PCT Appl. No. PCT/US90/04245, International Search Report dated Jan. 23, 1991.
EP Appln No. 97200767.8, European Search Report (illegible date).
JP Appln No. 1990–511130.
"A single chip digital signal processor, Part I—architecture and addressing", Richard Pickvance, Electronic Engineering, Feb. 1985, vol. 57, No. 698, pp. 53–56, 59 and 63.
"The Architecture of the SC32 Forth Engine", Hayes et al., The Journal of Forth Application and Research, 1989, vol. 5, No. 4, pp. 493–506.
"Machine Forth, Machine Forth for the ARM processor", Reuben Thomas, Aug. 23, 1999, 10 pages.
"Stack Computers; the new wave, Chapter 5.2: Architecture of the FRISC 3 (SC32)", Philip Koopman, 1989, 9 pages.
The Motorola MC68020, MacGregor, D. et al., IEEE Micro, vol. 4, issue 4, Aug. 1984, pp. 101–118.
MC68020 32–Bit Microprocessor User's Manual, Motorola, Prentice–Hall, 1984.
MOSTEK 1981 3870 / F8 Microcomputer Data Book Feb. 1981, pp. III–76—VI–11.
F8/3870 F6800 Bit–Slice Fairchild Microcomputers United Technical Publications, IC Master, 1980, pp. 2016–2040.
80386 Programmer's Reference Manual Intel, 1986.
Transputer Reference Manual INMOS, Prentice Hall, 1988.

* cited by examiner

EX PARTE REEXAMINATION CERTIFICATE ISSUED UNDER 35 U.S.C. 307

THE PATENT IS HEREBY AMENDED AS INDICATED BELOW.

Matter enclosed in heavy brackets [ ] appeared in the patent, but has been deleted and is no longer a part of the patent; matter printed in italics indicates additions made to the patent.

AS A RESULT OF REEXAMINATION, IT HAS BEEN DETERMINED THAT:

The patentability of claims **5-10** is confirmed.

Claims **1-4** are cancelled.

New claims **11-20** are added and determined to be patentable.

11. A microprocessor, which comprises a main central processing unit and a separate direct memory access central processing unit in a single integrated circuit comprising said microprocessor, said main central processing unit having an arithmetic logic unit, a first push down stack with a top item register and a next item register, connected to provide inputs to said arithmetic logic unit, an output of said arithmetic logic unit being connected to said top item register, said top item register also being connected to provide inputs to an internal data bus, said internal data bus being bidirectionally connected to a loop counter, said loop counter being connected to a decrementer, said internal data bus being bidirectionally connected to a stack pointer, return stack pointer, mode register and instruction register, said stack pointer pointing into said first push down stack, said internal data bus being connected to a memory controller, to a Y register of a return push down stack, an X register and a program counter, said Y register, X register and program counter providing outputs to an internal address bus, said internal address bus providing inputs to said memory controller and to an incrementer, said incrementer being connected to said internal data bus, said direct memory access central processing unit providing inputs to said memory controller, said memory controller having an address/data bus and a plurality of control lines for connection to a random access memory.

12. The microprocessor of claim 11 in which said memory controller includes a multiplexing means between said central processing unit and said address/data bus, said multiplexing means being connected and configured to provide row addresses, column addresses and data on said address/data bus.

13. The microprocessor of claim 11 in which said memory controller includes means for fetching instructions for said central processing unit on said address/data bus, said means for fetching instructions being configured to fetch multiple sequential instructions in a single memory cycle.

14. The microprocessor of claim 13 additionally comprising means connected to said means for fetching instructions for determining if multiple instructions fetched by said means for fetching instructions require a memory access, said means for fetching instructions fetching additional multiple instructions if the multiple instructions do not require a memory access.

15. The microprocessor of claim 13 in which said microprocessor and a dynamic random access memory are contained in a single integrated circuit and said means for fetching instructions includes a column latch for receiving the multiple instructions.

16. The microprocessor of claim 11 in which said microprocessor includes a sensing circuit and a driver circuit, and an output enable line for connection between the random access memory, said sensing circuit and said driver circuit, said sensing circuit being configured to provide a ready signal when said output enable line reaches a predetermined electrical level, said microprocessor being configured so that said driver circuit provides an enabling signal on said output enable line responsive to the ready signal.

17. The microprocessor of claim 11 additionally comprising a ring oscillator variable speed system clock connected to said main central processing unit, said main central processing unit and said ring oscillator variable speed clock being provided in a single integrated circuit.

18. The microprocessor of claim 17 in which said memory controller includes an input/output interface connected to exchange coupling control signals, addresses and data with said main central processing unit, said microprocessor additionally including a second clock independent of said ring oscillator variable speed system clock connected to said input/output interface.

19. The microprocessor of claim 11 in which said first push down stack has a first plurality of stack elements configured as latches, a second plurality of stack elements configured as a random access memory, said first and second plurality of stack elements and said central processing unit being provided in a single integrated circuit, and a third plurality of stack elements configured as a random access memory external to said single integrated circuit.

20. The microprocessor of claim 19 additionally comprising a first pointer connected to said first plurality of stack elements, a second pointer connected to said second plurality of stack elements, and a third pointer connected to said plurality of stack elements, said central processing unit being connected to pop items from said first plurality of stack elements, said first stack pointer connected to said second stack pointer to pop a first plurality of items from said second plurality of stack elements when said first plurality of stack elements are empty from successive pop operations by said central processing unit, said second stack pointer being connected to said third stack pointer to pop a second plurality of items from said third plurality of stack elements when said second plurality of stack elements are empty from successive pop operations by said central processing unit.

* * * * *

UNITED STATES COURT OF APPEALS
FOR THE FEDERAL CIRCUIT
HTC Corporation v. Technology Properties Limited, 2014-1076, -1317

PROOF OF SERVICE

I, John C. Kruesi, Jr., being duly sworn according to law and being over the age of 18, upon my oath depose and say that:

Counsel Press was retained by AGILITY IP LAW, LLP, Attorneys for Appellants to print this document. I am an employee of Counsel Press.

On **May 1, 2014**, Counsel for Plaintiff-Appellee has authorized me to electronically file the foregoing **PRINCIPAL BRIEF FOR DEFENDANTS-APPELLANTS** with the Clerk of Court using the CM/ECF System, which will serve via e-mail notice of such filing to any of the following counsel registered as CM/ECF users:

Heidi Lyn Keefe
Principal Counsel for Cross-Appellants
Kyle D. Chen
Mark R. Weinstein
Cooley LLP
Five Palo Alto Square
3000 El Camino Real
Palo Alto, CA 94306
650-843-5000
hkeefe@cooley.com
kyle.chen@cooley.com
mweinstein@cooley.com

Additionally, paper copies will be mailed to the principal counsel, at the addresses noted above, on the date paper copies are sent to the Court.

Upon acceptance by the Court of the e-filed document, six paper copies will be filed with the Court, via Federal Express, within the time provided in the Court's rules.

May 1, 2014

/s/ John C. Kruesi, Jr.
John C. Kruesi, Jr.
Counsel Press

CERTIFICATE OF COMPLIANCE

1. This brief complies with the type-volume limitation of Federal Rule of Appellate Procedure 32(a)(7)(B) or Federal Rule of Appellate Procedure 28.1(e).

__X__ The brief contains 9,423 words, excluding the parts of the brief exempted by Federal Rule of Appellate Procedure 32(a)(7)(B)(iii),or

____ The brief uses a monospaced typeface and contains _____ lines of text, excluding the parts of the brief exempted by Federal Rule of Appellate Procedure 32(a)(7)(B)(iii).

2. This brief complies with the typeface requirements of Federal Rule of Appellate Procedure 32(a)(5) and the type style requirements of Federal Rule of Appellate Procedure 32(a)(6).

__X__ The brief has been prepared in a proportionally spaced typeface using MS Word 2010 in a 14 point Times New Roman font or

____ The brief has been prepared in a monospaced typeface using _____ _____in a ___ characters per inch_____ font.

May 1, 2014
Date

/s/ James C. Otteson
James C. Otteson
Attorney for Defendants-Appellants