Michael J. McKeon (*Pro Hac Vice*), mckeon@fr.com
Christian A. Chu (SBN 218336), chu@fr.com
Richard A. Sterba (*Pro Hac Vice*), sterba@fr.com
FISH & RICHARDSON P.C.
1425 K Street, NW, Suite 1100
Washington, DC 20005
Telephone: (202) 783-5070
Facsimile: (202) 783-2331

Wasif Qureshi (*Pro Hac Vice*), qureshi@fr.com
FISH & RICHARDSON P.C.
1221 McKinney Street, Suite 2800
Houston, TX 77010
Telephone: (713) 654-5300
Facsimile: (713) 652-0109

Olga I. May (SBN 232012), omay@fr.com
FISH & RICHARDSON P.C.
12390 El Camino Real
San Diego, CA 92130
Telephone: (858) 678-4745
Facsimile: (858) 678-5099

Attorneys for Defendants
LG ELECTRONICS, INC. and LG ELECTRONICS U.S.A., INC.

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN JOSE DIVISION

TECHNOLOGY PROPERTIES LIMITED LLC, PHOENIX DIGITAL SOLUTIONS LLC, and PATRIOT SCIENTIFIC CORPORATION,

Plaintiffs,

v.

LG ELECTRONICS, INC. AND LG ELECTRONICS U.S.A., INC.,

Defendants.

Case No. 3:12-cv-03880-VC

DEFENDANTS LG ELECTRONICS, INC. AND LG ELECTRONICS U.S.A., INC.'S NOTICE OF MOTION AND MOTION TO STRIKE INFRINGEMENT CONTENTIONS OR, ALTERNATIVELY, COMPEL SUPPLEMENTAL INFRINGEMENT CONTENTIONS ON CHARTED PROCESSORS

DATE: June 30, 2015 (by order)
TIME: 10:00 AM
PLACE: Courtroom 5
JUDGE: Hon. Paul S. Grewal

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Page(s)

Cases

Statutes

Other Authorities

NOTICE OF MOTION AND MOTION

TO THE COURT AND ALL COUNSEL OF RECORD:

NOTICE IS HEREBY GIVEN that on August 11, 2015, at 10:00 a.m., or as soon thereafter as counsel may be heard in Courtroom 5 of the above-titled court, located at 280 S. 1st St, San Jose, CA 95113, Defendants LG Electronics, Inc. and LG Electronics U.S.A., Inc. (collectively "LG") will and hereby do move the Court for an order striking Plaintiffs Technology Properties Limited LLC, Phoenix Digital Solutions LLC, and Patriot Scientific Corporation (collectively "Plaintiffs")' Patent L.R. 3-1 Disclosure of Asserted Claims and Infringement Contentions or, in the alternative, an order compelling Plaintiffs to provide supplemental infringement contentions that comply with Patent L.R. 3-1.

This motion is based upon this notice, the accompanying memorandum of points and authorities, the accompanying declaration of Olga May, all pleadings, papers and records on file in this action, and such oral argument as may be presented before or at the hearing in this matter.

STATEMENT OF REQUESTED RELIEF

LG requests that the Court strike Plaintiffs' January 20, 2015 Disclosure of Asserted Claims and Infringement Contentions in its entirety for failure to comply with Patent Local Rule 3-1.

In the alternative, and to the extent the Court permits Plaintiffs to serve supplemental infringement contentions, LG requests that the supplemental Infringement Contentions: (a) be limited to the accused LG products that were previously identified in Table A.6 of Exhibit A to Plaintiffs' January 20, 2015 Infringement Contentions and that include one of the following processors: Qualcomm MSM8960, Qualcomm MSM8660, Qualcomm MSM8260, Qualcomm MSM7227, TI OMAP4430, TI OMAP4460, or TI OMAP4470; (b) identify specifically where each limitation of each asserted claim is found within each accused LG product, including, at a minimum, providing a separate claim chart for each of the above seven identified microprocessors; and (c) exclude all other products (identified in Table A.6 of Exhibit A or otherwise) from the scope of this case for all purposes.

MEMORANDUM OF POINTS AND AUTHORITIES

I. INTRODUCTION

Plaintiffs' Disclosure of Asserted Claims and Infringement Contentions served on January 20, 2015 ("Infringement Contentions") should be stricken for failure to comply with the specificity requirement of Patent Local Rule 3-1 to identify where each element of each asserted claim is found in each accused instrumentality.

Plaintiffs' Infringement Contentions purport to accuse 437 LG products, including mobile phones, tablets, TVs, and Blu-ray players. But Plaintiffs' infringement charts barely mention only two of these products without charting either one through the claim elements of any single claim. Instead, the charts focus on accusing the microprocessors contained within the accused LG end products. Plaintiffs' list of the accused products and processors, produced as a separate attachment to the charts, lists a profusion of processors of different makes and models by more than half a dozen different manufacturers. For this indiscriminate list of processors, Plaintiffs produced only three charts, one for each asserted patent. Each chart lumps "each Accused Microprocessor in each Accused Product" together and purports to accuse them all at once without providing the requisite specific identification as to where each claim element is found in every individual accused processor. Although the charts reference seven specific processors and provide some evidentiary support as to those, the charts fail to identify any basis for why the rest of the processors can be accused along with these seven or be in any way represented by them. Indeed, the most Plaintiffs endeavored was to rely on impermissible conclusory assertions based on "information and belief." Such conclusory allegations do not approach the specificity requirement of Rule 3-1 and should be stricken.

The allegations with respect to the seven charted processors, although better supported, also fail to meet the specificity requirements. None of these processors are charted through an entire claim and also rely on conclusory allegations of "information and belief," unsupported understanding of one of ordinary skill in the art, and on generic descriptions of technology.

Further, a number of the accused products were released after the respective August 8, 2012 and June 25, 2013 expirations of the '749 and '890 Patents. Because an infringement claim can only be asserted for the patent's term, contentions as to the products released after the expiration of the '749 and '890 Patents should be stricken.

Finally, a large number of the accused products LG has never made, sold, used, or imported into the United States. Because the general rule under United States patent law is that no infringement occurs when a patented product is made and sold in a country other than the United States, all contentions as to such products should be stricken.

Therefore, LG respectfully requests that Plaintiffs' Infringement Contentions be stricken in their entirety. Alternatively, if this Court is inclined to grant Plaintiffs leave to supplement, LG requests that any supplementation be limited to the LG products already included on Plaintiffs' list of the accused products, sold in the U.S., and containing the seven charted processors.

Any broader supplementation would result in undue expansion of the case, substantially prejudice LG, and give Plaintiffs an unfair second bite at the apple. Plaintiffs produced their Infringement Contentions two and a half years after this case was filed (this case was stayed while the parties litigated one of the asserted patents—the '336 Patent—in a parallel ITC investigation). Fact discovery will close on September 8, 2015. LG already produced its invalidity contentions on April 21, 2015. Restarting and expanding the case three months before the end of fact discovery and after service of invalidity contentions based in part on Plaintiffs' infringement positions will significantly impair LG's ability to prepare its defenses. Accordingly, LG respectfully requests that any permitted supplementation be limited to the already identified LG products containing the seven charted processors.

II. FACTUAL BACKGROUND

A. Case History

Plaintiffs filed this case against LG on July 24, 2012, alleging infringement of three patents: U.S. Patent Nos. 5,809,336 ("the '336 Patent"), 5,440,749 ("the '749 Patent") and 5,530,890 ("the '890 Patent") (collectively the "Patents-in-Suit"). [Dkt. No. 1.] Concurrently,

Plaintiffs filed seven more cases ("the related cases") asserting the same patents against a total of eight groups of defendants.

The Patents-in-Suit are generally directed to processor design and operation. The LG accused products include mobile phones, tablets, TVs, and Blu-ray players.

On October 2, 2012, the Court stayed this case and the related cases pending the resolution of a pending U.S. International Trade Commission investigation filed by Plaintiffs against LG (and a number of other respondents) alleging infringement of the '336 Patent (Inv. No. 337-TA-853, the "853 Investigation"). [Dkt. No. 12.] The '749 and '890 Patents were not at issue in the 853 Investigation. On September 6, 2013, the Administrative Law Judge issued an Initial Determination in the 853 Investigation finding that LG's products did not infringe the '336 Patent. [Dkt. No. 20 at 5-6]. On February 19, 2014, the full Commission issued a notice affirming the ALJ's non-infringement findings and terminating the investigation. Plaintiffs did not appeal the Commission's final determination.

After the stay in the present case was lifted, on November 20, 2014, this Court issued a scheduling order setting the deadline for Plaintiffs' Rule 3-1 Infringement Contentions for January 20, 2015. [Dkt. No. 40.]

B. Deficiencies in Plaintiffs' Infringement Contentions

On January 20, 2015—nearly two and a half years after filing their Complaint—Plaintiffs served their Infringement Contentions. The Infringement Contentions purported to accuse 437 LG products—mobile phones, tablets, TVs and Blu-ray players—of infringing all of the asserted claims of each of the '336, '749 and '890 Patents. [May Decl., Exs. A-B, G-1, G-2, G-3.]

Plaintiffs' Infringement Contentions consist of:

(1) a cover pleading that sets forth infringement allegations with respect to all eight related cases [May Decl., Ex. B];

(2) "Exhibit A" to the cover pleading that purports to list all accused products of all defendants in the eight related cases, including LG [May Decl., Ex. A]; and

(3) three claim charts per case, one for each of the three Patents-in-Suit [May Decl., Exs. G-1, G-2, G-3].

1. Cover Pleading

The cover pleading states that the three claim charts identify "where each element of each asserted claim [] may be found within LG's Accused Instrumentalities identified in Ex. A." [May Decl., Ex. B at 8.] The cover pleading also states that "[t]he list of accused instrumentalities in this action includes those listed in Ex. A, and all models thereof." [*Id.* at 5.]

2. Exhibit A

Exhibit A to Plaintiffs' Infringement Contentions contains eight tables identifying the purportedly accused products of each defendant in the related cases. "Table A.6: LG" of Exhibit A lists the 437 purportedly accused LG products. [May Decl., Ex. A at 56-68.] Of these 437 products, 116 are not identified by name but by what appears to be a part or product number.

Table A.6 has columns for the following information for the products: including "Memory," "Processor," "CPU Core," and "Instruction Set(s)." [*Id.*] However, the table fails to identify this information for each product. For example, for 118 of the accused products, the "Processor" information is missing.

The table also fails to indicate which products are accused against which patent.

3. Claim Charts

Exhibits G-1, G-2, and G-3 are each a claim chart for a Patent-in-Suit. Of the 437 purportedly accused products, these charts mention a total of two: LG Escape and LG Optimus 9. [May Decl., Ex. G-1 at 1, 19, G-3 at 1 (references to Escape); G-3 at 1 (reference to Optimus).] ***The other 435 products are never named.*** The two named products are only briefly mentioned in the beginning of a chart and not charted through each element of each asserted claim.

Instead, the charts focus on a handful of processors, referred to as "the Accused Microprocessors," contained within the accused end products. Each chart (using the '336 chart as an example), makes substantially the same statement in the preamble of the first asserted claim:

> On information and belief, each Accused Product listed in the attached list of Accused Products (Ex. A.6), including phones and televisions, contains a microprocessor ("Accused Microprocessors"). http://www.phonescoop.com/; http://pdadb.net/; http://www.gsmarena.com/; *see* PDSLG000001-PDSLG000050; *see also* PDSND077821-PDSND078576 for datasheets identifying microprocessors. [1] ***For example, the Escape contains a Qualcomm MSM8960L.*** *See* Ex. A.6 for listings of microprocessors in the Accused Products with information obtained from http://www.phonescoop.com/; http://pdadb.net/; http://www.gsmarena.com/; *see* PDSLG000001-PDSLG000050; *see also* PDSND077821-PDSND078576 for datasheets identifying microprocessors."

[*Id.,* G-1 at 1; *see also* G-2 at 1, G-3 at 1] (emphasis added).

The chart then broadly accuses "each Accused Microprocessor in each Accused Product" without citing any support or documents for such "each Microprocessor." Instead, the charts mainly refer to the processors in large groups organized not by any relevant functionality but by the manufacturer: for example, ARM, Qualcomm, or Texas Instruments ("TI").

Taking one of those examples, the Infringement Contentions refer to ARM processors as a single undifferentiated group.[1] ARM processors are a large family of processing cores developed over the last thirty years.[2] Even the subset of ARM processing cores allegedly found in the Accused Products contains several different design generations and different members within those generations. TPL does not provide a comparative analysis of any of the relevant characteristics of these different ARM processors. TPL applies the same undifferentiated treatment to TI processors and Qualcomm processors throughout its Infringement Contentions.[3]

Specific processors are few and referenced only as examples. The charts name a total of seven specific processors used as examples for different claims:

1. Qualcomm MSM8960
2. Qualcomm MSM8660
3. Qualcomm MSM8260

---

[1] *See, e.g.*, May Decl., Exs. G-2 at 1, 5-6, 10-11, 16, 20-22, 27, 63, 67-68, 72, 78, 82-84, 89, 94-95, 115 ('890 elements 1.b, 1.c, 1.d, 1.e, 1.f, 1.g, 1.h, 1.i, 11.b, 11.c, 11.d, 11.e, 11.f, 11.g, 11.h, 11.i, 11.j and 13.b); and G-3 at 1-2, 4-8, 10-11, 15-16, 24, 46-47, 54-55, 71-72, 74, 89, ('749 elements 1 (preamble), 1.a, 1.b, 1.c, 1.d, 1.e, 1.f, 1.g, 1.i, 1.k, and 9.d). For those claim charts without element labels, the elements are referred to in alphabetical order, starting with "X.a" for the preamble.

[2] *See generally* http://www.arm.com/about/company-profile/milestones.php.

[3] *See, e.g.*, May Decl., Exs. G-1 at 12-15, 20, 22 ('336 elements 6.d and 9.b); G-2 at 51, 117 ('890 elements 7.b and 17.b); and G-3 at 2-11, 15-16, 24, 45-47, 54, 70 ('749 elements 1.a, 1.b, 1.c, 1.d, 1.e, 1.f, 1.g, and 1.i).

4. Qualcomm MSM7227
5. TI OMAP4430
6. TI OMAP4460
7. TI OMAP4470

The rest of the purportedly accused processors listed in Exhibit A ***are never mentioned*** in the charts and there is no evidence in support of any claims against them.

The charts do not provide supporting evidence for a large number of the allegations and thus even the seven specifically mentioned processors are not charted through every element of each claim with consistent evidentiary support. Many of the allegations are made "on information and belief."[4] For certain claim elements, TPL simply states that such elements are present without providing any supporting evidence or by merely stating that the microprocessor has the recited claim language.[5] TPL also contends that certain claim elements are present based on the understanding of one of ordinary skill in the art.[6] For yet other elements, TPL cites to generic descriptions of technology in public articles and Wikipedia instead of identifying where the element is in the actual accused product or processor.[7] This lack of support could not have resulted from lack of available information regarding the design and operation of the accused functionality, because Plaintiffs admit that they had public technical information available at the time they prepared the Contentions. [May Decl., Ex. D (Pls.' Ltr. at 2 ("For example, PDS provided numerous citations to public ARM documents to support its contentions.").]

---

[4] *See, e.g.*, May Decl., Exs. G-1 at 1, 4-5, 17-20, 23 ('336 elements 6.a, 6.b, 6.c, 6.f, 6.g and 9.b); G-2 at 1, 11, 20, 23-24, 27, 32, 35, 41-43, 47, 50, 53-54, 56, 59-60, 63, 65, 67-68, 72-73, 82, 85-86, 89, 95, 98, 104-106, 109, 112, 115, 117, 119-120, 122-123, 126-127 ('890 elements 1.a, 1.b, 1.c, 1.f, 1.g, 1.i, 1.j, 1.k, 1.l, 1.m, 1.n, 1.o, 7.b, 9.b, 9.c, 9.d, 9.e, 11.a, 11.b, 11.c, 11.d, 11.e, 11.f, 11.g, 11.i, 11.k, 11.l, 11.m, 11.n, 11.o, 11.p, 12.b, 13.b, 17.b, 19.b, 19.c, 19.d, and 9.e); and G-3 at 1, 3, 8, 15, 24, 45, 54, 63, 65, 70, 74, 84, 88-89, 92, 96, 99, 104, 106 ('749 elements 1 (preamble), 1.a, 1.b, 1.c, 1.d, 1.e, 1.f, 1.g, 1.h, 1.i, 1.j, 1.k, 1.l, 9.d, 9.e, 43 (preamble), 43.a, 43.b and 59 (preamble)).

[5] *See, e.g.,* May Decl., Exs. G-1 at 19 ('336 element 6.h); G-2 at 53, 72-73 ('890 elements 7.b and 11.f)

[6] *See, e.g.,* May Decl., Exs. G-1 at 16-17 ('336 elements 6.d and 6.e), 18-19 (6.g); G-2 at 112 ('890 element 12.b).

[7] *See, e.g.*, Infr. Cont., Ex. G-1 at 10-11, 16-20 ('336 elements 6.d, 6.e, 6.g and 9.b); G-2 at 1, 5, 50, 67, 113, 117 ('890 elements 1.b, 1.c, 7.b, 11.c, 12.b and 17.b)

Finally, although the charts purport to make the infringement allegations on a representative basis, this is based solely on "information and belief."[8] Each chart contains a footnote stating: "This chart provides some examples of such operation that, ***on information and belief, are representative*** of the operation of the processors in each of the Accused Products." [May Decl., Exs. G-1, G-2, G-3 at 1, respectively] (emphasis added). But nowhere do the Infringement Contentions attempt to show how the rest of the accused processors are similar to the seven representative processors used as examples.

C. The Parties' Meet-and-Confer Process

On February 27, 2015, LG sent Plaintiffs a letter regarding the deficiencies in their Infringement Contentions and requesting to meet and confer. [May Decl., Ex. C.] On March 18, 2015, Plaintiffs responded by letter attempting to defend the adequacy of their contentions. [*Id.*, Ex. D.]

On June 8, 2015, in response to Plaintiffs' question regarding the scope of financial information LG would produce for the accused products, LG informed Plaintiffs it would produce financial information for the accused U.S. products that contain the seven processors charted in Plaintiffs' Infringement Contentions. On June 15, 2015, the parties had a call regarding these issues, deficiencies in Plaintiffs' Infringement Contentions, and LG's prospective motion to strike. [May Decl. ¶ 10.] Because the parties were unable to resolve the dispute, LG informed Plaintiffs it will file the present motion. [*Id.*]

A motion to strike on similar grounds was filed on June 9, 2015, in related case *Technology Properties Limited LLC, et al. v. Samsung Electronics Co., LTD.*, Case No. 3:12-cv-03877-VC (PSG) and is currently set for hearing on August 11. For judicial efficiency purposes, LG noticed the hearing on its motion for the same date and time.

---

[8] *See, e.g.*, May Decl., Exs. G-1 at 1, 4-5, 17-19 ('336 elements 6.a, 6.b, 6.c, 6.f, 6.g and 6.h); Exs. G-2 at 1, 50, 63, 78, 112 ('890 elements 1.a, 1.b, 7.b, 11.a, 11.b, 11.g and 12.b); and G-3 at 1-2, 99 ('749 elements 1 (preamble), 1.a, 43 (preamble), 43.a). For most claim elements, TPL simply does not address whether or why the specifically cited microprocessors in its infringement charts should be considered "representative" of the operation and/or implementation of other microprocessors incorporated in LG's hundreds of accused products.

III. LEGAL STANDARD

Patent Local Rule 3-1(c) requires a patentee to provide "[a] chart identifying ***specifically*** where ***each limitation of each asserted claim*** is found within ***each Accused Instrumentality***." Patent L.R. 3-1(c) (emphasis added); *see Silicon Labs., Inc. v. Cresta Tech. Corp.*, No. 5:14-cv-03227-PSG, 2015 WL 846679, at *1 (N.D. Cal. Feb. 25, 2015); *Shared Memory Graphics LLC v. Apple, Inc.*, 812 F. Supp. 2d 1022, 1024 (N.D. Cal. 2010); *Bender v. Freescale Semiconductor, Inc.*, No. C 09–1156 PHJ (MEJ), 2010 WL 1689465, at *3 (N.D. Cal. Apr. 26, 2010).

"[T]he degree of specificity under Local Rule 3-1 must be sufficient to provide reasonable notice to the defendant why the plaintiff believes it has a reasonable chance of proving infringement." *Shared Memory Graphics LLC v. Apple, Inc.*, 812 F. Supp. 2d 1022, 1025 (N.D. Cal. 2010) (citing *View Eng'g Inc. v. Robotic Vision Sys., Inc.*, 208 F.3d 981, 986 (Fed. Cir. 2000)). This requires that a party "map specific elements of Defendants' alleged infringing products onto Plaintiff's claim construction." *Id.*

This rule was designed to "make the parties more efficient, to streamline the litigation process, and to articulate with specificity the claims and theory of a plaintiff's infringement claims." *InterTrust Techs. Corp. v. Microsoft Corp.*, No. C 01-1640-SBA, 2003 WL 23120174, at *2 (N.D. Cal. Dec. 1, 2003). A plaintiff violates this rule where it fails to "provide reasonable notice to the defendant why the plaintiff believes it has a 'reasonable chance of proving infringement.'" *Shared Memory Graphics*, 812 F.Supp.2d at 1025 (quoting *View Eng'g, Inc. v. Robotic Vision Sys., Inc.*, 208 F.3d 981, 986 (Fed. Cir. 2000)).

IV. ARGUMENT

A. Plaintiffs' Contentions with Respect to All Products that Do Not Contain the Seven Charted Processors Should Be Stricken for Lack of Any Evidentiary Support

Plaintiffs' Infringement Contentions fail to meet Patent L.R. 3-1(c) requirement to identify "specifically where each limitation of each asserted claim is found within ***each*** Accused Instrumentality." Patent L.R. 3-1(c) (emphasis added); *see also Silicon Labs.*, 2015 WL 846679, at *1; *Bender*, 2010 WL 1689465, at *3.

At most, Plaintiffs provided such identification to the seven processors specifically named in the charts: Qualcomm MSM8960, Qualcomm MSM8660, Qualcomm MSM8260, Qualcomm MSM7227, TI OMAP4430, TI OMAP4460, and TI OMAP4470, and end products containing those processors. Accordingly, the bare allegations as to "each Accused Microprocessor in each Accused Product" beyond the seven charted processors should be stricken.

1. Allegations as to "Each Accused Microprocessor" Based on "Information and Belief," Knowledge of One Skilled in the Art or Generic Information Fail to Meet Rule 3-1

The Infringement Contentions as to "each Accused Microprocessor in each Accused Product" are vague, conclusory and fail to meet the specificity requirements of Patent L.R. 3-1. "Vague contentions and conclusory statements invite Defendants and the Court to merely assume the presence of" an element and "fall short of the specificity required by Local Rule 3-1." *Shared Memory Graphics*, 812 F.Supp.2d at 1026. The "[p]laintiff may not craft its infringement contentions without *specifically identifying what in the device satisfies the limitation*." *Infineon Techs. v. Volterra Semiconductor*, No. 11-cv-6239-MMC, 2012 U.S. Dist. LEXIS 146499, at *4 (N.D. Cal. Oct. 9, 2012) (emphasis in original).

First, although the charts purport to accuse "each Accused Microprocessor in each Accused Product" as to every element of every claim, these blanket allegations are made largely "on information and belief."[9] As this Court previously found, "simply alleging 'on information and belief' and representing 'vague, conclusory, and confusing statements' does not satisfy the requirement that the identifications be 'as specific as possible.'" *Solannex, Inc. v. MiaSole, Inc.*, 2013 WL 1701062, at *3 (N.D. Cal. April 18, 2013) (citing *Theranos, Inc. v. Fuisz Pharma LLC,* 11–CV–05236–YGR, 2012 WL 6000798, at *3 (N.D. Cal. Nov. 30, 2012)); *see also CSR Tech.*

[9] *See, e.g.*, May Decl., Exs. G-1 at 1, 4-5, 17-20, 23 ('336 elements 6.a, 6.b, 6.c, 6.f, 6.g and 9.b); G-2 at 1, 11, 20, 23-24, 27, 32, 35, 41-43, 47, 50, 53-54, 56, 59-60, 63, 65, 67-68, 72-73, 82, 85-86, 89, 95, 98, 104-106, 109, 112, 115, 117, 119-120, 122-123, 126-127 ('890 elements 1.a, 1.b, 1.c, 1.f, 1.g, 1.i, 1.j, 1.k, 1.l, 1.m, 1.n, 1.o, 7.b, 9.b, 9.c, 9.d, 9.e, 11.a, 11.b, 11.c, 11.d, 11.e, 11.f, 11.g, 11.i, 11.k, 11.l, 11.m, 11.n, 11.o, 11.p, 12.b, 13.b, 17.b, 19.b, 19.c, 19.d, and 9.e); and G-3 at 1, 3, 8, 15, 24, 45, 54, 63, 65, 70, 74, 84, 88-89, 92, 96, 99, 104, 106 ('749 elements 1 (preamble), 1.a, 1.b, 1.c, 1.d, 1.e, 1.f, 1.g, 1.h, 1.i, 1.j, 1.k, 1.l, 9.d, 9.e, 43 (preamble), 43.a, 43.b and 59 (preamble)).

Inc. v. Freescale Semiconductor, No. C–12–02619 RS (JSC), 2013 WL 503077, at *7-8 (N.D. Cal. Feb. 8, 2013). Therefore reliance on "information and belief" is impermissible and fails to comply with Rule 3-1.

Second, to the extent the contentions rely on knowledge of a person of ordinary skill in the art,[10] a bare assertion that a person of ordinary skill in the art could locate the element in the accused instrumentality does not comply with Rule 3-1. *See, e.g., Bender*, 2010 WL 1689465, at *4 ("[m]erely alluding to the fact that any electrical engineer would understand the infringement contentions is not sufficient."). The bald assertions in Plaintiffs' Infringement Contentions that certain claim elements are present based on the understanding of one of ordinary skill in the art are not substantiated by any other evidence and do not rise above an arbitrary unsupported conclusion. Thus they cannot serve as support of Plaintiffs' allegations.

Third, the Contentions attempt to rely on generic articles and Wikipedia for conclusory statements about general technology.[11] Plaintiffs' generic descriptions of technology lack the specific identification of each element of an asserted claim in each accused instrumentality and cannot meet the specificity requirement of Rule 3-1.

Plaintiffs' failure to provide sufficient support is inexcusable given the admittedly available public materials regarding the design of the accused microprocessors. [May Decl., Ex. D at 2, Ex. E.]

2. Allegations based on "Representative" Products Fail to Meet Rule 3-1

Nor can the contentions as to "each Accused Microprocessor" survive based on the seven named processors being "representative" of the rest. First, none of the seven "representative" processors are charted through every element of every claim and can "represent" the rest of the products as to each asserted claim. Second, as this Court has previously explained, in order to rely on a claim that one accused product is representative of another for purposes of Rule 3-1(c), a patentee must do more than state as much—it must show how. *Silicon Labs.*, 2015 WL 846679, at

[10] *See, e.g.,* May Decl., Exs. G-1 at 10-11, 16-17 ('336 elements 6.d and 6.e), 18-19 (6.g); G-2 at 112 ('890 element 12.b).

[11] [May Decl., Ex. G-1 at 10-11, 16-20 ('336 elements 6.d, 6.e, 6.g and 9.b); Ex. G-2 at 1, 5, 50, 67, 113, 117 ('890 elements 1.b, 1.c, 7.b, 11.c, 12.b and 17.b)).]

*1 (finding that where the claim required "a single integrated circuit," the charted product did not appear representative because it consisted of a single die in a package and the non-charted "represented" product consisted of two dies which would correspond to two circuits). Rule 3-1 "requires Plaintiff to articulate how the accused products share the same, or substantially the same, infringing [qualities] with any other product or with the . . . 'representative' product[]." *Id.* (citing *Bender*, 2010 WL 1689465, at *3); *see also Ameranth, Inc. v. Pizza Hut, Inc.*, Nos. 12-cv-00729 et al., 2013 WL 3894880, at *7 (S.D. Cal. July 26, 2013) ("While [the plaintiff] says it identified the accused versions temporally, by their functional aspects or by their version names or numbers, it must at least state how the accused previous versions are the same or reasonably similar to the charted version, or else provide a separate chart for each version.").

Here, Plaintiffs have failed to provide any evidence showing how the seven charted processors "share the same, or substantially the same, infringing [qualities]" with "each Accused Microprocessor" with respect to each element of each asserted claim. In fact, Plaintiffs' charts openly acknowledge that their allegations regarding "representative" products are speculative and based solely on "information and belief": "[t]his chart provides some examples of such operation that, ***on information and belief, are representative*** of the operation of the processors in each of the Accused Products." [May Decl., Exs. G-1, G-2, G-3 at 1, respectively] (emphasis added).

For 118 of the 437 accused products, plaintiffs fail to even identify the corresponding processors and it is impossible to tell what features are being "represented." [May Decl., Ex. A at 56-58.] The identified processors and cores include dozens of different makes and models, including made by Qualcomm, TI, ARM, NVidia, LG, and MediaTek. Some of them, for example NVidia and MediaTek, are not mentioned in the charts at all. The others, for example, the subset of ARM processing cores, contain several different design generations and different members within those generations.

By way of example only, Plaintiffs' Infringement Contentions fail to establish a reasonable chance of success in proving infringement of claim 1 of the '890 Patent by any product including only an ARM1176 Processor. Like in *Silicon Labs*, the "representative" and "represented"

products have "different structures that are material to the infringement of at least one of the asserted claims" and therefore Plaintiffs' contentions based on a representative basis fail. *Silicon Labs.*, 2015 WL 846679, at *1.

Claim 1 of the '890 Patent requires both a "main central processing unit" and "a separate direct memory access central processing unit in a single integrated circuit . . ." May Decl., Ex. 4 (Exhibit G-2 at 1). Plaintiffs allege that the first central processing unit ("CPU") limitation is satisfied by the presence of a "first ARM core," and the second, separate CPU limitation is satisfied because "many of the Accused Microprocessors are multicore processors that contain more than one ARM core." *Id*. However, even publicly available documents (of the type already cited by Plaintiffs in their contentions) show that many of the allegedly "represented" processors in Table A.6 have only one core. For example, as shown below, the specifications available on ARM's website show that the ARM1136 Processor has only one core.

ARM1136	
Architecture	ARMv6
Dhrystone Performance	1.25 DMIPS/MHz
Multicore	No - Single core only

[See May Decl., Ex. E; http://www.arm.com/products/processors/classic/arm11/arm1136.php.] Thus by itself, the ARM1136 Processor cannot infringe.

B. Infringement Charts as to Seven Named Processors Are Deficient

Despite providing specific processor names and some cites to guides, the infringement contentions based on the seven named processors also fail to meet the requirements of Rule 3-1.

Plaintiffs fail to chart a single microprocessor against every limitation of any one asserted claim, which alone makes the Contentions deficient under Rule 3-1. For example, for certain elements, Plaintiffs rely, instead of showing where the element is in the accused processor, on "information and belief" [*see, e.g.,* May Decl., Ex. G-1 at 15-16, 18-19], knowledge of a person of ordinary skill in the art [*id.,* G-1 at 11, 16], or generic evidence [*id.,* G-1 at 15, 19]. Therefore

these charts violate Rule 3-1 for the same reasons as the allegations regarding "each Accused Microprocessor" stated above.

C. Infringement Contentions based on the '890 and '749 Patents as to the Accused Products that Were Released after the Expiration of These Patents Should Be Stricken

A claim of patent infringement can be asserted only for the term of the patent. *See* 35 U.S.C. § 271(a) ("whoever without authority makes, uses, offers to sell, or sells any patented invention, within the United States or imports into the United States any patented invention ***during the term of the patent therefor***, infringes the patent") (emphasis added).

The '749 Patent expired on August 8, 2012, and the '890 Patent expired on June 25, 2013. [35 U.S.C. § 154(c); Dkt. No. 1, Exs. A, B.] Plaintiffs accused the same products as to all three patents. A number of the accused LG products were first released after the expiration of the '749 Patent and after the expiration of the '890 Patent. [*See* May Decl., Ex. C at 2.] Therefore Plaintiffs' Infringement Contentions based on the '749 and '890 Patents should be stricken as to the products released after the expiration of these patents.

D. Infringement Contentions as to the Accused Products that LG Has Never Made, Used, Imported, Sold, or Offered for Sale in the United States Should Be Stricken

Direct infringement liability is "limited to infringing activities that occur within the United States." *MEMC Elec. Materials, Inc. v. Mitsubishi Materials Silicon Corp*., 420 F.3d 1369, 1375 (Fed. Cir. 2005); *see also* 35 U.S.C. § 271(a) ("A person who, without a license, "makes, uses, offers to sell, or sells any patented invention within the United States or imports into the United States" is liable for direct patent infringement."). There can be no induced or contributory infringement without direct infringement. *MEMC Elec. Materials*, 420 F.3d at 1378; *Fujitsu Ltd. v. Netgear Inc.*, 620 F.3d 1321, 1326 (Fed. Cir. 2010). Therefore the "general rule under United States patent law is that no infringement occurs when a patented product is made and sold in another country." *Microsoft Corp. v. AT & T Corp*., 550 U.S. 437, 441 (2007). There is a "strong policy against extraterritorial liability" in the patent law. *See Halo Elecs., Inc. v. Pulse Elecs., Inc*., 769 F.3d 1371, 1378 (Fed. Cir. 2014).

At least 208 of the 437 accused LG products in Plaintiffs' Exhibit A LG has not made, sold, offered for sale, or imported into the United States. [May Decl., Ex. F.] Therefore LG cannot be liable for infringement based on these products. Plaintiffs' Infringement Contentions as to these products should be stricken.

E. If the Court Permits Plaintiffs to Supplement Their Contentions, Any Amendment Should Be Limited to Products with the Seven Charted Processors

As demonstrated above, Plaintiffs' infringement Contentions are deficient and should be stricken in their entirety. But in the event the Court permits Plaintiffs to cure the deficiencies in their Infringement Contentions and supplement with additional support, such amendment should avoid expanding the case beyond the seven charted processors because it would substantially prejudice LG.

The purpose of Rule 3-1 is 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." *Digital Reg of Texas, LLC v. Adobe Sys., Inc.*, No. 12-CV-01971-CW (KAW), 2013 U.S. Dist. LEXIS 93814, at *9 (N.D. Cal. July 3, 2013) (quoting *InterTrust Techs. Corp. v. Microsoft Corp.*, No. 01-1640 SBA, 2003 U.S. Dist. LEXIS 22736, at *1 (N.D. Cal. Dec. 1, 2003)).

> The patent local rules were adopted by this district in order to give claim charts more "bite." The rules are 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.... 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.

Atmel Corp. v. Information Storage Devices, Inc., 1998 WL 775115 at *2 (N.D. Cal. Nov. 5, 1998).

Expanding this case beyond the seven charted processors at this late point in the schedule would prejudice LG's ability to prepare its defenses. This case was filed on July 24, 2012. Fact discovery is set to close on September 8, 2015—less than three months away. All defendants in the related cases, including LG, already produced their invalidity contentions on April 21, 2015 in reliance, in part, on their understanding of Plaintiffs' infringement contentions.

Permitting Plaintiffs to expand the case would also give Plaintiffs an unfair second bite at the apple. Plaintiffs produced their Infringement Contentions on January 20, 2015—two and half years after they filed this case and also after they litigated the ITC investigation on the '336 Patent on some of the same products. Plaintiffs have had years, based on extensive publicly available information, to investigate their claims and prepare properly supported contentions, and are not entitled to restart the case less than three months before fact discovery cut-off.

V. CONCLUSION

Based on the above, LG respectfully requests that its motion be granted and Plaintiffs' Infringement Contentions be stricken in their entirety. In the alternative, to the extent the Court grants Plaintiffs leave to serve supplemental contentions, LG requests an order compelling supplemental contentions that:

(a) are limited to the accused LG products that were previously identified in Table A.6 of Exhibit A to Plaintiffs' January 20, 2015 contentions, that include one of the following processors: Qualcomm MSM8960, Qualcomm MSM8660, Qualcomm MSM8260, Qualcomm MSM7227, TI OMAP4430, TI OMAP4460, and TI OMAP4470; were not released after the expiration of the corresponding Patent-in-Suit; and are not on the list of LG products that have not been sold in the U.S.;

(b) comply with Patent Local Rule 3-1 by identifying specifically where each limitation of each asserted claim is found within each accused LG product, including, at a minimum, providing a separate claim chart for each of the above seven accused microprocessors; and

(c) exclude all other products (identified in Table A.6 of Exhibit A or otherwise) from the scope of this case for all purposes.

Dated: June 29, 2015

FISH & RICHARDSON P.C.

By: */s/ Olga I. May*
Olga I. May

Attorneys for Defendants LG ELECTRONICS, INC. and LG ELECTRONICS U.S.A., INC.

Michael J. McKeon (*Pro Hac Vice*), mckeon@fr.com
Christian A. Chu (SBN 218336), chu@fr.com
Richard A. Sterba (*Pro Hac Vice*), sterba@fr.com
FISH & RICHARDSON P.C.
1425 K Street, NW, Suite 1100
Washington, DC 20005
Telephone: (202) 783-5070
Facsimile: (202) 783-2331

Wasif Qureshi (*Pro Hac Vice* to be filed), qureshi@fr.com
FISH & RICHARDSON P.C.
1221 McKinney Street, Suite 2800
Houston, TX 77010
Telephone: (713) 654-5300
Facsimile: (713) 652-0109

Olga I. May (SBN 232012), omay@fr.com
FISH & RICHARDSON P.C.
12390 El Camino Real
San Diego, CA 92130
Telephone: (858) 678-4745
Facsimile: (858) 678-5099

Attorneys for Defendants
LG ELECTRONICS, INC. and LG ELECTRONICS U.S.A., INC.

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN JOSE DIVISION

TECHNOLOGY PROPERTIES LIMITED LLC, PHOENIX DIGITAL SOLUTIONS LLC, and PATRIOT SCIENTIFIC CORPORATION,

Plaintiffs,

v.

LG ELECTRONICS, INC. AND LG ELECTRONICS U.S.A., INC.,

Defendants.

Case No. 3:12-cv-03880-VC

EXHIBITS G-1 TO G-3 TO THE DECLARATION OF OLGA I. MAY IN SUPPORT OF DEFENDANTS LG ELECTRONICS, INC. AND LG ELECTRONICS U.S.A., INC.'S MOTION TO STRIKE INFRINGEMENT CONTENTIONS OR, ALTERNATIVELY, COMPEL SUPPLEMENTAL INFRINGEMENT CONTENTIONS ON CHARTED PROCESSORS

DATE: June 30, 2015 (by order)
TIME: 10:00 AM
PLACE: Courtroom 5
JUDGE: Hon. Paul S. Grewal

EXHIBIT G-1

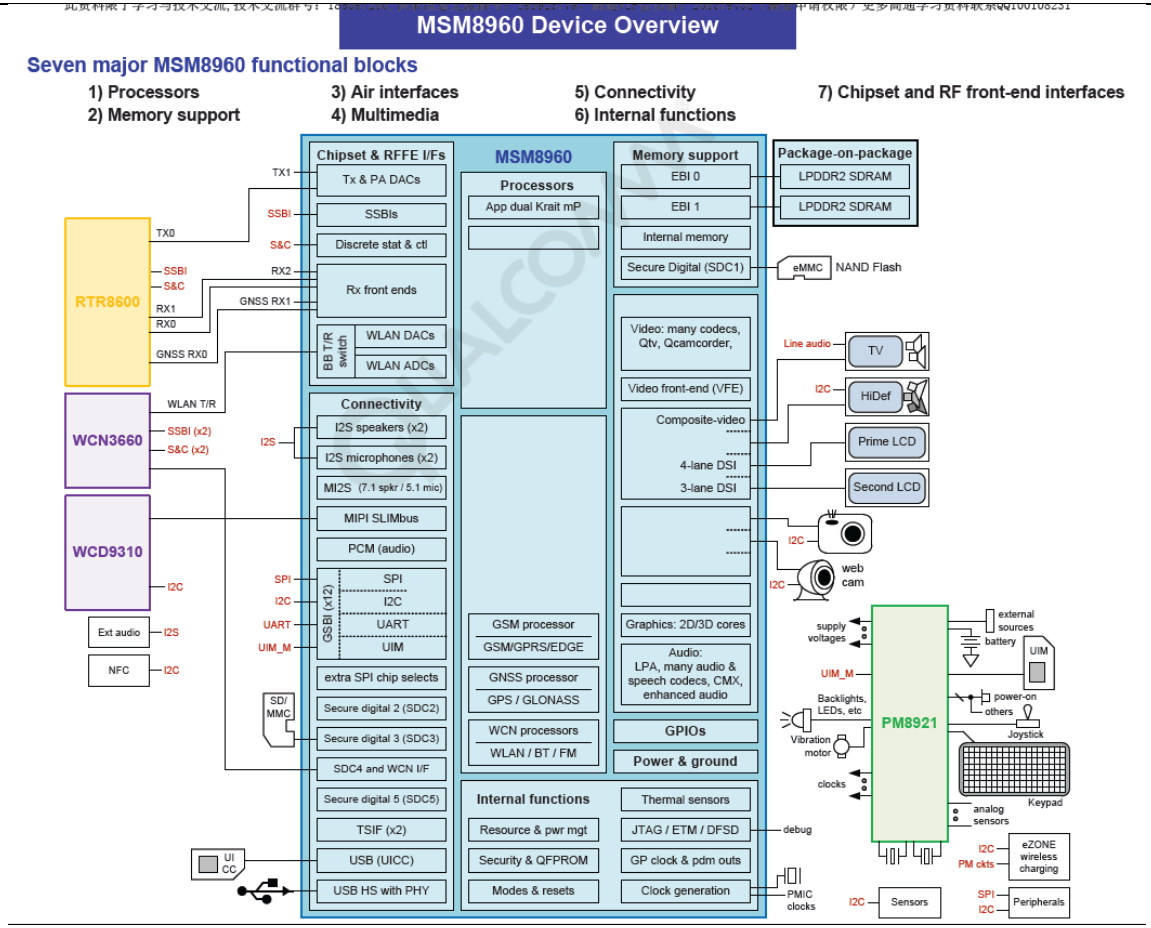
DECLARATION OF OLGA I. MAY IN SUPPORT OF DEFENDANTS LG ELECTRONICS, INC. AND LG ELECTRONICS U.S.A., INC.'S MOTION TO STRIKE INFRINGEMENT CONTENTIONS OR, ALTERNATIVELY, COMPEL SUPPLEMENTAL INFRINGEMENT CONTENTIONS ON CHARTED PROCESSORS

EXHIBIT G-1 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,809,336 By LG

Claim 6		Claim Element
6. A microprocessor system comprising:	On information and belief, each Accused Product listed in the attached list of Accused Products (Ex. A.6), including phones and televisions, contains a **microprocessor** ("Accused Microprocessors").[1] For example, the Escape contains a Qualcomm MSM8960L. *See* Ex. A.6 for listings of microprocessors in the Accused Products with information obtained from http://www.phonescoop.com/; http://pdadb.net/; http://www.gsmarena.com/; *see* PDSLG000001-PDSLG000050; *see also* PDSND077821-PDSND078576 for datasheets identifying microprocessors. Each microprocessor is an electronic circuit that interprets and executes programmed instructions.	6.a
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;	On information and belief, each Accused Microprocessor in each Accused Product contains a **central processing unit** (CPU), which is an electronic circuit on an integrated circuit that controls the interpretation and execution of programmed instructions. *See* http://www.merriam-webster.com/dictionary/microprocessor. Each CPU operates at a processing frequency. Each CPU and an entire oscillator are constructed on a single integrated circuit comprising semiconductor-based transistors, or electronic devices. An integrated circuit is a miniature circuit on a single semiconductor substrate. For example: • The Qualcomm MSM8960 microprocessor includes a Krait CPU.	6.b

[1] Infringement by the Accused Products is largely based on the operation of and implementation of the microprocessors they contain. This chart provides some examples of such operation that, on information and belief, are representative of the operation of the processors in each of the Accused Products. Discovery is in the early stages, and Plaintiffs anticipate receiving additional documents showing the exact operation of the processor in each of the Accused Products with respect to the accused functionality. But because many documents that Plaintiffs would rely on to establish infringement are confidential and have not yet been produced in this litigation, Plaintiffs anticipate receiving additional documents to confirm the operational principles shown in this chart from Defendants and/or third parties. Accordingly, Plaintiffs reserve the right to amend, supplement, or augment their claim charts, infringement contentions, or infringement theories based on documents and information later received through discovery.

EXHIBIT G-1 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,809,336 By LG



MSM8960 Chipset Training at 9.

- The Qualcomm MSM8660 microprocessor includes a Scorpion processor

EXHIBIT G-1 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,809,336 By LG

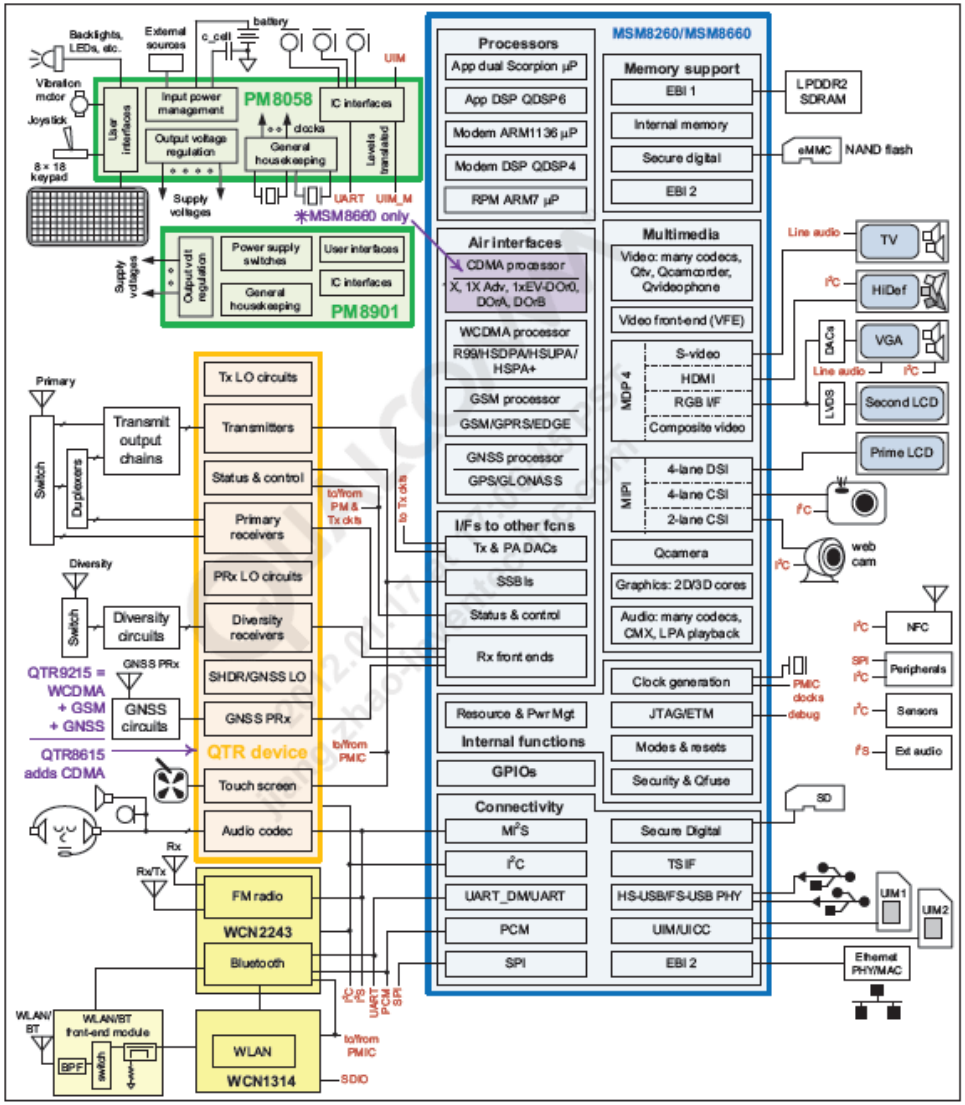


Figure 1-1 MSM8260/MSM8660 functional block diagram in a typical application

MSM8260/MSM8660 Mobile Station Modem User Guide at 16.

EXHIBIT G-1 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,809,336 By LG

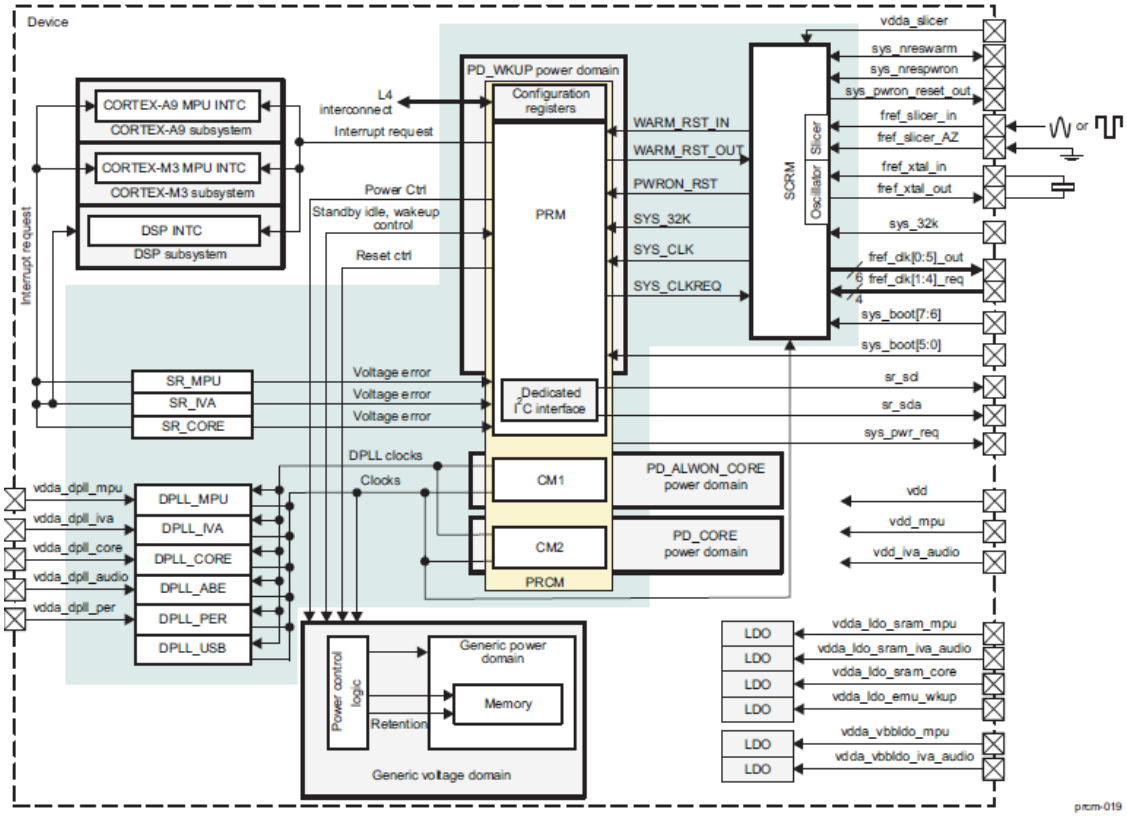
	• The TI OMAP4430 includes an ARM Cortex-A9 CPU. **Figure 3-18. PMFW Overview**  OMAP4430 Technical Reference Manual at 337. *See* other microprocessor user guides at PDSND000001-PDSND012125; PDSND012126-PDSND048086; PDSND085767-PDSND088712.	
an entire oscillator disposed upon said integrated circuit substrate and connected	On information and belief, each Accused Microprocessor in each Accused Product **clocks its CPU** at a clock rate, generating the signals used for timing the operation of	6.c

EXHIBIT G-1 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,809,336 By LG

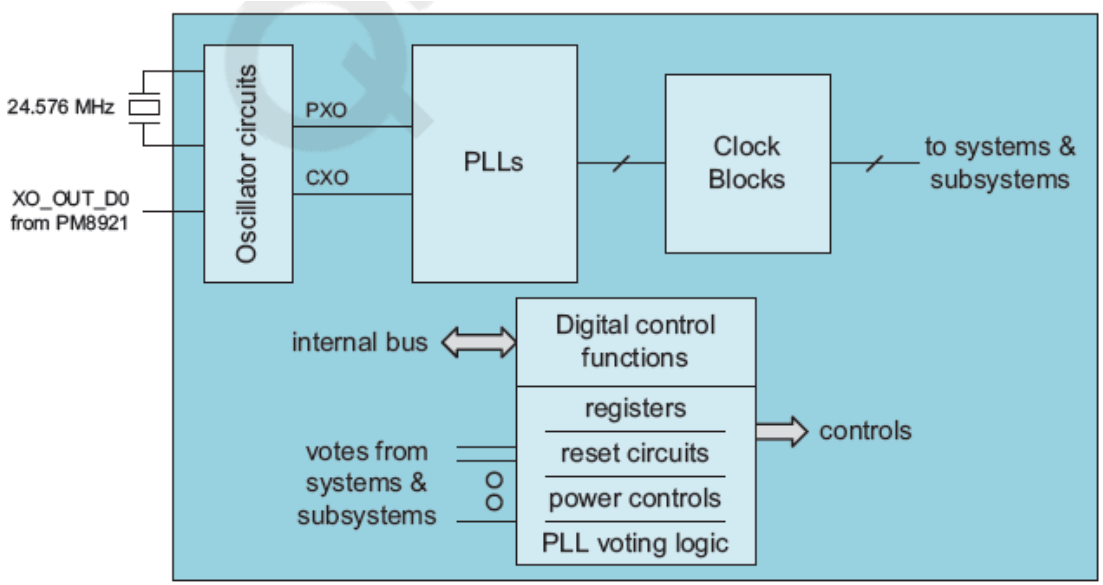
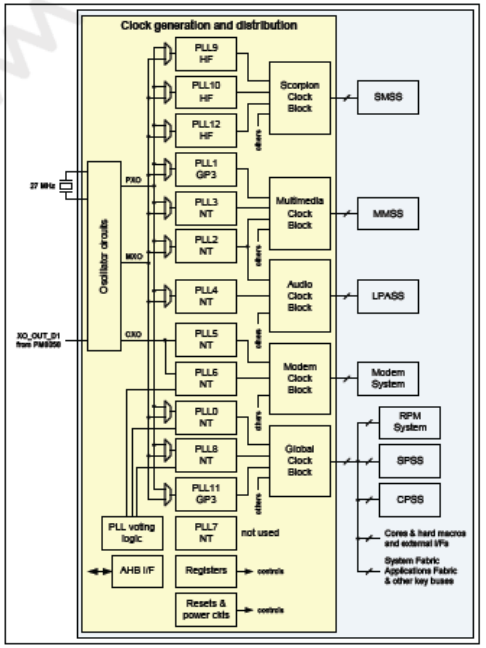
to said central processing unit, said oscillator clocking said central processing unit at a clock rate and being constructed of a second plurality of electronic devices,	the CPU, using PLL circuitry containing an **entire oscillator** disposed upon the integrated circuit substrate, connected to the CPU, and constructed of a second plurality of electronic devices. Each oscillator does not use any external clock to generate the signal used to clock the CPU. For example: • The MSM8960 clocks its CPU using PLL:  MSM8960 Chipset Schematics and Design Guidelines at 103-108. • The MSM8660 clocks its CPU using PLL	

EXHIBIT G-1 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,809,336 By LG

Clock Generation and Distribution

- The on-chip PLLs (PLL0 through PLL12)
  - PLL0 (reserved)
  - PLL1 (multimedia PLL0)
  - PLL2 (multimedia PLL1)
  - PLL3 (multimedia PLL2)
  - PLL4 (LPASS PLL)
  - PLL5 (modem PLL0)
  - PLL6 (modem PLL1)
  - PLL7 (unused)
  - PLL8 (peripheral PLL)
  - PLL9 (SC1 PLL0)
  - PLL10 (SC2 PLL1)
  - PLL11 (EBI1 PLL)
  - PLL12 (SC1/2 L2 PLL)



MSM8660 Mobile Station Modem Architecture and Features at 40; See also MSM8260/MSM8660 Mobile Station Modem User Guide at 72, 73 (Clock generation and distribution).

- The TI OMAP4430 clocks its CPU using PLL. The DPLL_MPU generates a clock for the MPU subsystem.

  To generate high-frequency clocks, the device supports multiple on-chip DPLLs controlled directly by the PRCM module. They are of two types, identified as type A and type B DPLLs.

EXHIBIT G-1 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,809,336 By LG

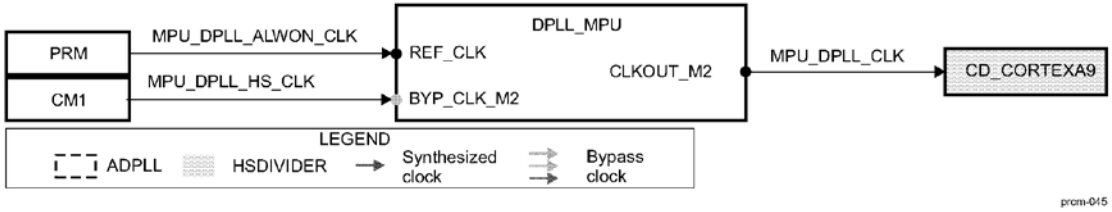
	Following DPLLs belong to type A: • DPLL_MPU • DPLL_IVA • DPLL_CORE • DPLL_PER • DPLL_ABE Following DPLLs belong to type B: • DPLL_USB TPLBN037652 (OMAP 4430 Tech Ref Manual). *See also* TPLBN037641-TPLBN037755 (OMAP 4430 Tech Ref Manual). DPLL_MPU supplies the source clock for the MPU subsystem, which uses this source clock to internally generate all subsystem clock signals within the MPU subsystem. **Figure 3-50. DPLL_MPU Overview**  TPLBN037679, TPLBN038343 (OMAP4430 Tech Ref Manual).	

EXHIBIT G-1 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,809,336 By LG

Table 4-2. Clocks and Resets

Module Instance	Destination Signal Name	Source Signal Name	Source	Description
		Clocks		
CORTEXA9	MPU_DPLL_CLK	MPU_DPLL_CLK	PRCM	Interface and functional clock
		Resets		
CORTEXA9	CORTEXA9_PWRON_RSTN	CORTEXA9_PWRON_RSTN	PRCM	Power-on reset for all the modules inside the MPU system power domain, nonretention
	CORTEXA9_RSTN	CORTEXA9_RSTN	PRCM	Warm reset for all the modules inside the MPU system power domain, nonretention

4.2.1 Clock Distribution

The Cortex-A9 MPU clock generator is fed by the MPU DPLL, which can be gated off by the global power, reset, and clock management (PRCM) module when system power domain is in a low-power state. There is a global clock gating for each CPU. Due to the MPU DPLL, the Cortex-A9 MPU subsystem is asynchronous from the rest of the device.

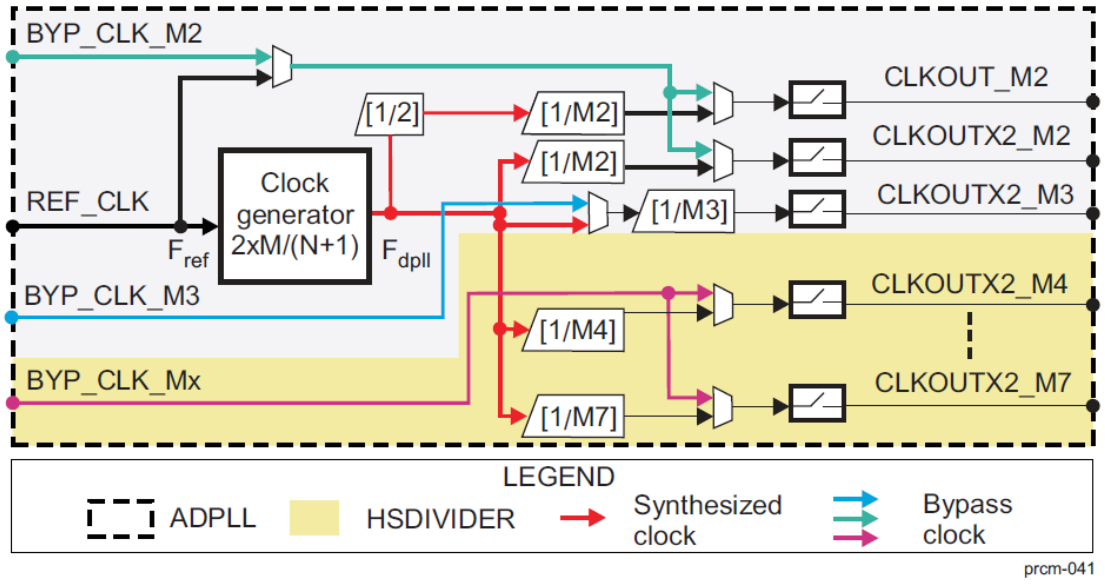
The clock generator generates the following clocks from the MPU DPLL output clock:

- ARM_FCLK – ARM Cortex-A9 MPCore functional clock
- LOCAL_INTCNT_FCLK – Local interconnect functional clock
- CACHE_CTRL_FCLK – PL310 cache controller functional clock

TPLBN038343 (OMAP4430 Tech Ref Manual).

EXHIBIT G-1 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,809,336 By LG

Figure 3-40. Generic DPLL Functional Diagram



TPLBN037653 (OMAP4430 Tech Ref Manual).

EXHIBIT G-1 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,809,336 By LG

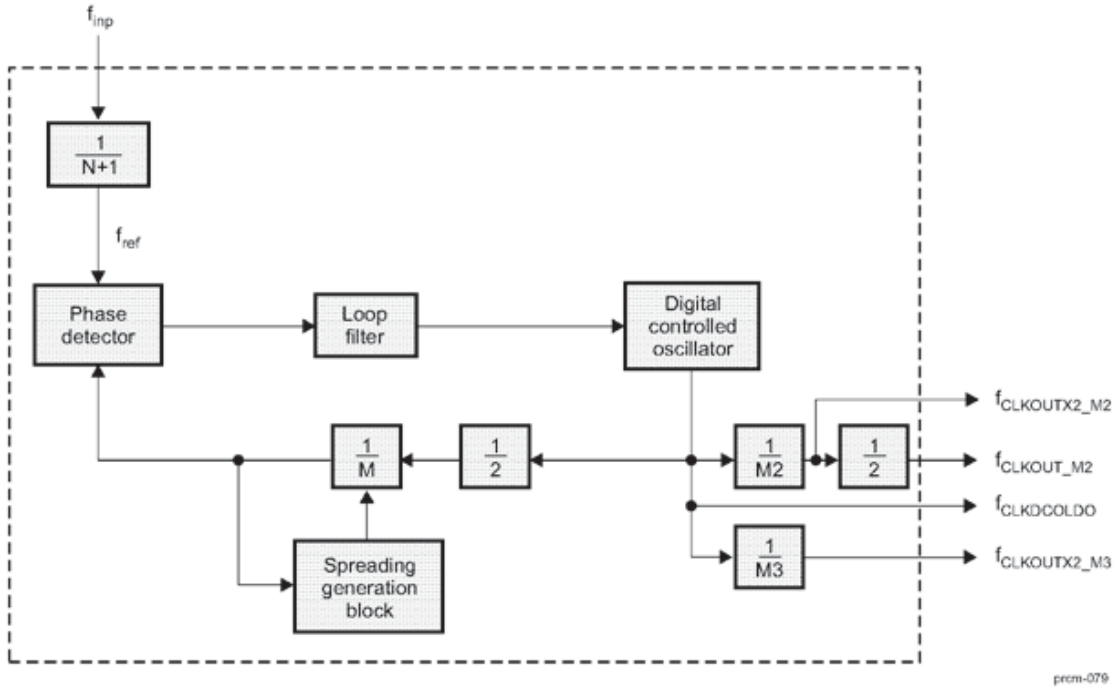
	Figure 3-41. DPLL With SSC Reduction Feature  TPLBN037657 (OMAP4430 Tech Ref Manual). *See* other microprocessor user guides at PDSND000001-PDSND012125; PDSND012126-PDSND048086; PDSND085767-PDSND088712; *see also* http://en.wikipedia.org/wiki/Phase-locked_loop. The presence of a PLL indicates the presence of a voltage or current controlled oscillator, which itself will include the claimed oscillator. The claimed oscillator is disposed upon the integrated circuit substrate, connected to the central processing unit, and is clocking the central processing unit. The oscillator is constructed of a second plurality of electronic devices.	
thus varying the processing frequency of	In each Accused Microprocessor of each Accused Product, because the central	6.d

EXHIBIT G-1 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,809,336 By LG

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,	processing unit and the entire oscillator reside on the same integrated circuit, they were constructed using the same process technology and will have corresponding manufacturing variations. One of ordinary skill in the art would understand this with respect to the Accused Products based on generally accepted principles relating to semiconductor ICs. *Design of High-Performance Microprocessor Circuits* pp. 98, 101 (Anatha Chandrakasan et al. eds., IEEE Press, 2001) [*Models of Process Variations in Device and Interconnect* (Duane Boning and Sani Nassif)] (TPL853_02927444 – TPL853_02927464). **6.1 INTRODUCTION: SOURCES OF VARIATION** Variation is the deviation from intended or designed values for a structure or circuit parameter of concern. The electrical performance of microprocessors or other integrated circuits are impacted by two sources of variation. *Environmental factors* arise during the operation of a circuit, and include variations in power supply, switching activity, and temperature of the chip or across the chip. *Physical factors* during manufacture result in structural device and interconnect variations that are essentially permanent. These variations arise due to processing and masking limitations, and result in random or spatially varying deviations from designed parameter values. In this chapter we focus on *parametric variation* due to continuously varying structural or electrical parameters, as these can significantly impact not only yield but also performance in high-speed microprocessor and other digital circuits. Such parametric variation is becoming a larger concern, as variation and margins for device and interconnect do not appear to be scaling at the same rate. Figure 6.1 6.2 Overview: Statistical Descriptions **101** While a simplified distribution is often assumed to capture the variance die-to-die, it is often possible and helpful to characterize and model systematic trends across the wafer. For example, chip speed may depend on some parameter that varies in a systematic "bowl" fashion across the wafer; accurate speed binning or yield analysis and improvement may depend on understanding such patterns from one die to the next. In typical circuit design, die-to-die variations are the simplest to analyze; a small number of situations may be analyzed in which all device or interconnect parameters on the chip are mean shifted together up or down by some amount (e.g., 2–3σ).	

EXHIBIT G-1 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,809,336 By LG

TI confirms that processing frequency, the speed at which the CPU operates, and the clock rate of the oscillator vary in the same way, increasing and decreasing proportionally, as a function of parameter variation in one or more fabrication or operational parameters, such as temperature or voltage, associated with the integrated circuit substrate:

> Figure 1 below gives a general view of how performance and power dissipation vary with process, temperature and voltage. . . .

	Performance	Power Dissipation
Process	Linear	Linear
Temperature	1/Logarithmic	Exponential
Voltage	Exponential	Exponential

Figure 1. Overview of key dependencies.

> . . . ICs can be manufactured in a variety of silicon processes. Processes in use today include 180-nm, 130-nm, and 90-nm processes. The performance of an IC depends on the characteristics of the underlying process. When manufacturers design an IC, they target the nominal process characteristics. However, variations in the process generate devices that are sometimes weaker ("colder") than desired or sometimes stronger ("hotter") than a nominal device. Hot devices can provide higher levels of performance than cold or nominal devices, and vice-versa. A typical relationship is shown in Figure 2 on the following page.
>
> The performance documented in a data sheet takes into account the worse-case process variation. In other words, the maximum operational frequency given in the data sheet is determined by the weakest, cold devices. As illustrated in Figure 2, these weak devices represent only a fraction of the devices shipped.

EXHIBIT G-1 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,809,336 By LG

Therefore, the maximum performance of most devices exceeds the data sheet specification.

Similarly, the data sheet provides an operational temperature range, say –40°C–105°C. The documented performance is for the worst performance across the temperature limits. Figure 2 shows how the operational frequency of a device tends to increase with decreasing temperatures. A given device can have a significantly higher performance level if it is run below maximum temperature specification, e.g., at 75°C instead of 105°C.

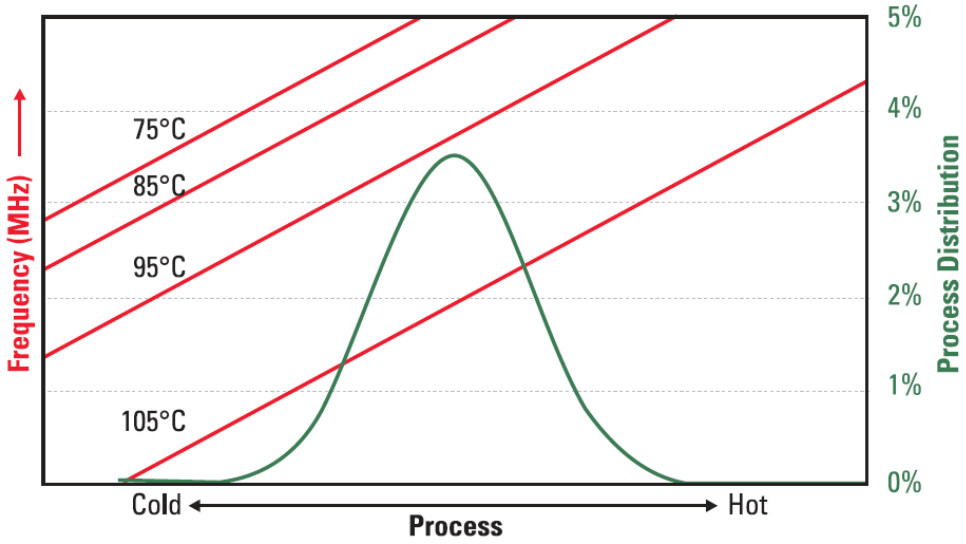


Figure 2. Dependency on process variation and temperature.

There is a similar relationship with respect to voltage. A device provides more performance as the voltage increases. In Figure 3, the minimum voltage Va determines the performance listed in the specification. At higher voltages (Vb, Vc, Vd, respectively), the performance tends to improve.

All of these trends provide the basis for the final specification provided to a

EXHIBIT G-1 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,809,336 By LG

customer. In order to provide some margin, each parameter is guard banded to ensure the specification is met under all voltage, frequency, temperature, and process conditions, for a particular number of power-on hours. If a device does not meet the required performance at the limit of the specification plus guard band, the device is discarded.

From a batch of devices that meet the specification, most are likely able to outperform the data sheet performance limits.

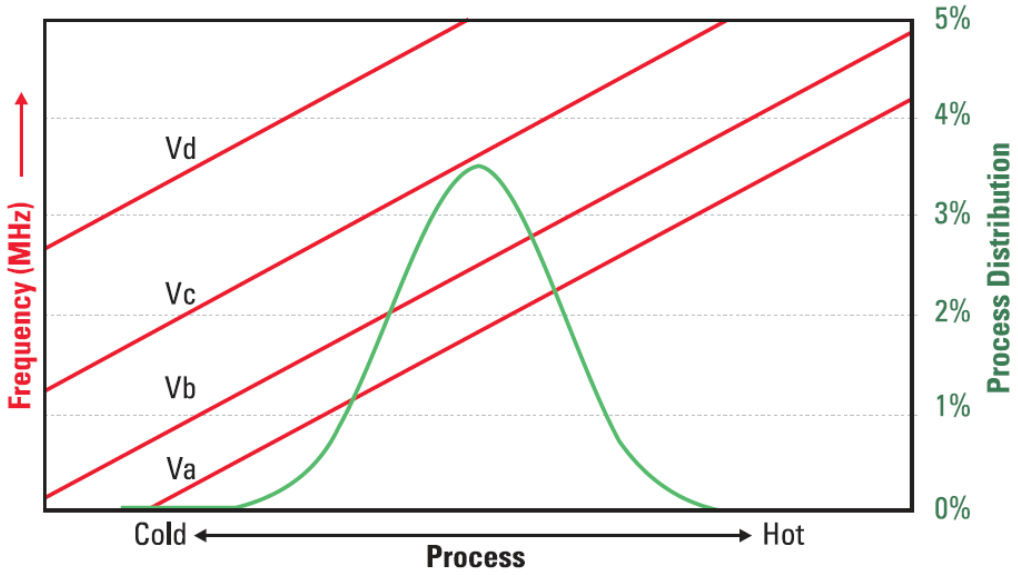


Figure 3. Dependency on voltage.

. . . So, what does all of this mean to the engineer designing the system? It means:

• An application that is performance-centric can find additional performance by lowering the temperature, or by increasing the voltage, or both. . . .

EXHIBIT G-1 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,809,336 By LG

• Most devices have higher performance and lower power dissipation than the data sheet suggests. . . .

By understanding these details behind the data sheet specification, you can create the product you need—even when the data sheet says it is impossible.

TPLBN033411-15 (Frantz, Mai, and Garcia, *Push Performance and Power Beyond the Data Sheet*, Texas Instruments White Paper (July 2007)).

On information and belief, TI's OMAP processors employ a mechanism called "Dynamic Voltage Frequency Scaling," or "DVFS," whereby the ring oscillator and the CPU run at various operating points or "OPP." Making use of DVFS allows the chip to run at a slower frequency to conserve power and battery life when faster processing is not required. According to TI, it is sometimes desirable to change the frequency of the signal output from the ring oscillator so that the chip may run at different speeds depending on the needs of the system. The only way to vary the clock speed output from the ring oscillator is to vary the input current. Even when purposeful variation of the ring oscillator output frequency is not desired, the output frequency will still vary by a few percentage points.

Further, the OMAP processors and first clocks vary together as a result of process conditions. TI employs binning to separate chips with different process capabilities that result from process variations.

Table 4-15. DPLL1 Clock Frequency Ranges

Clock Signal	Description		Max	Unit
ARM_CLK	DPLL1 output clock.	OPP6(1)	720	MHz
		OPP5	600	MHz
		OPP4	550	MHz
		OPP3	500	MHz
		OPP2	250	MHz
		OPP1(2)	125	MHz

(1) OPP6 frequency range is only supported on high-speed grade OMAP3530/25 devices.
(2) Cannot boot in OPP1. If OPP1 is desired, boot in higher OPP then switch to OPP1.

EXHIBIT G-1 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,809,336 By LG

	TPLBN037143 (indicating "speed binning"). *See also* TPL853_02993475-84 (ISSCC 2004 / Session 26 / *Optical and Fast IO 26.10*) (CX-0170); TPL853_02993485-96 (CX-0171). Similarly, on information and belief, Samsung's documents will show that Samsung is concerned with PVT variation in the design and operation of its chips because CPU processing frequency and the clock rate of the oscillator in Samsung microprocessors also vary in the same way, increasing and decreasing proportionally, as a function of parameter variation in one or more fabrication or operational parameters, such as temperature or voltage, associated with the integrated circuit substrate.	
thereby enabling said processing frequency to track said clock rate in response to said parameter variation;	The central processing unit and the oscillator in each Accused Microprocessor of each Accused Product are constructed on the same integrated circuit using the same process technology. Accordingly, the processing frequency of the central processing unit is enabled to track the clock rate, increasing and decreasing proportionally, in response to said parameter variation, including variation in fabrication, operating voltage or temperature of the integrated circuit. One of ordinary skill in the art would understand this with respect to the Accused Products based on generally accepted principles relating to semiconductor ICs. *See Design of High-Performance Microprocessor Circuits* p. 98 (Anatha Chandrakasan et al. eds., IEEE Press, 2001) [*Models of Process Variations in Device and Interconnect* (Duane Boning and Sani Nassif)] (TPL853_02927444 – TPL853_02927464).	6.e

EXHIBIT G-1 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,809,336 By LG

	6.1 INTRODUCTION: SOURCES OF VARIATION Variation is the deviation from intended or designed values for a structure or circuit parameter of concern. The electrical performance of microprocessors or other integrated circuits are impacted by two sources of variation. *Environmental factors* arise during the operation of a circuit, and include variations in power supply, switching activity, and temperature of the chip or across the chip. *Physical factors* during manufacture result in structural device and interconnect variations that are essentially permanent. These variations arise due to processing and masking limitations, and result in random or spatially varying deviations from designed parameter values. In this chapter we focus on *parametric variation* due to continuously varying structural or electrical parameters, as these can significantly impact not only yield but also performance in high-speed microprocessor and other digital circuits.	
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	On information and belief, each Accused Microprocessor in each Accused Product includes on-chip input/output interfaces, which are circuits having logic for input/output communications, located on the same semiconductor substrate as the CPU. The input/output interfaces are connected between the central processor unit and an off-chip external memory bus, which is a group of conductors coupled between the I/O interface and an external storage device, as well as other peripherals, for facilitating exchanging coupling control signals, addresses and data with said central processing unit. For example, the OMAP4430 includes I/O interfaces:	6.f

EXHIBIT G-1 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,809,336 By LG

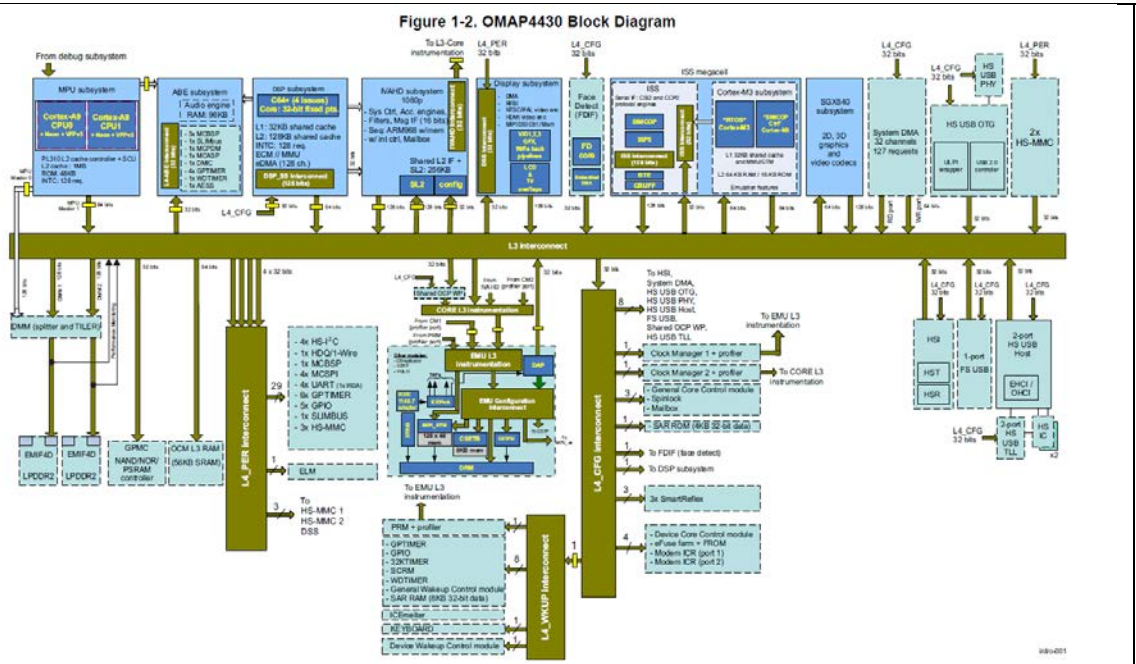
	 TPLBN037538 (OMAP4430 Tech Ref Manual); *see supra* MSM8960 Chipset Training at 9, and MSM8260/MSM8660 Mobile Station Modem User Guide at 16; *see also* other microprocessor user guides at PDSND000001-PDSND012125; PDSND012126-PDSND048086; PDSND085767-PDSND088712. On information and belief, each Accused Microprocessor in each Accused Product includes on-chip USB input/output interfaces connected between the CPU and an off-chip external memory bus, as well as other peripherals.	
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	On information and belief, each Accused Product includes one or more off-chip external clocks, which are not on the integrated circuit substrate, independent of the oscillator, connected to one or more of the input/output interfaces. The off-chip external clocks operate at a frequency independent of the clock frequency of the oscillator: a change in the frequency of either the external clock or oscillator does not affect the frequency of the other.	6.g

EXHIBIT G-1 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,809,336 By LG

	When an Accused Product receives data via its USB connection to a PC or Mac, it receives a clock signal encoded in that data. This clock signal is independently generated on the PC or Mac and is received on the Accused Product's USB data pins. The clock signal sent by the PC or Mac is recovered in the USB PHY (a physical implementation of the USB I/O interface), and is used to clock the data transfer in the Accused Product. *See* http://en.wikipedia.org/wiki/USB*; see also* USB specification at http://www.usb.org/developers/docs/. On information and belief, the schematics of the Accused Products will also show similar pins and operation, for USB, UART, and other interfaces.	
wherein a clock signal from said off-chip external clock originates from a source other than said oscillator.	In each Accused Product, the off-chip external clock generates a clock signal that originates from a source other than the oscillator. Each external second clock originates from an external device, separate from the Accused Product, and thus is both independent from and **asynchronous to** the CPU clock. Therefore, 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. For example, the clock signals encoded in the data stream received by each Accused Product from PC and Mac peripherals via USB originate from the peripherals. The source of these clock signals is therefore independent and asynchronous from any clock source in the Accused Product. In its user manuals and publications, LG instructs its users regarding how to connect each Accused Product to PC or Mac peripherals in order to transfer data and receive system updates. *See e.g.*, LG Escape User Manual at 10 (instructing user to connect phone to computer via USB); *see* PDSND048087-PDSND077820 for other user manuals.	6.h

EXHIBIT G-1 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,809,336 By LG

Claim 7	
7. A microprocessor system of claim 6	*See* Claim Element 6.a.
wherein said one or more operational parameters include operating temperature of said substrate or operating voltage of said substrate.	*See* Claim Elements 6.d, 6.e.

Claim 9	
9. A microprocessor system of claim 6	*See* Claim Element 6.a.
wherein said oscillator comprises a ring oscillator.	*See* Claim Element 6.c. On information and belief, in each Accused Microprocessor, the PLL clocking the CPU contains either a Voltage Controlled Oscillator (VCO) or a Current Controlled Oscillator (ICO), which is a **ring 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. *See* http://en.wikipedia.org/wiki/Ring_oscillator. For example, generally in TI microprocessors, PLLs have voltage-controlled oscillators, which are circuits capable of maintaining an alternating output. TI has published papers regarding its DPLLs confirming the use of ring oscillators.

EXHIBIT G-1 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,809,336 By LG

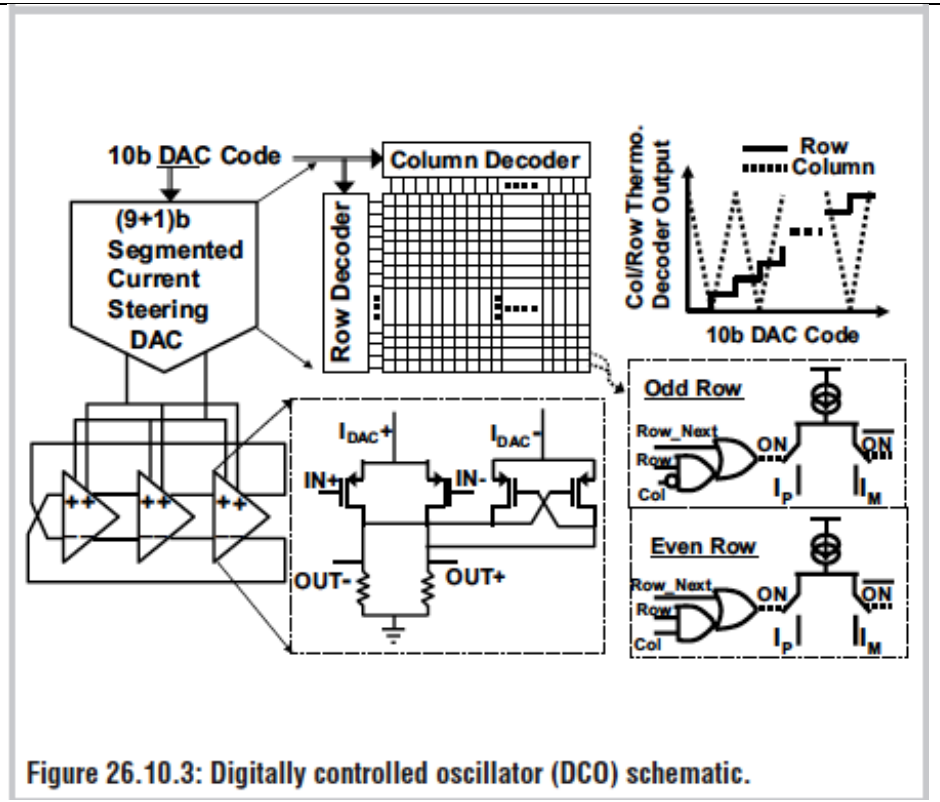
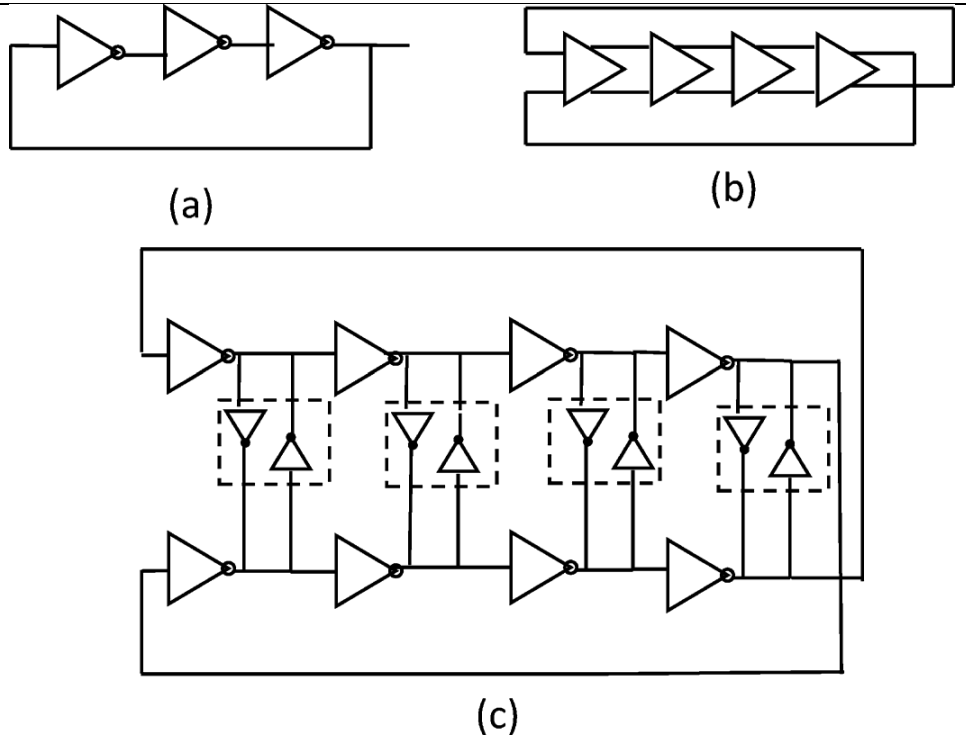


Figure 26.10.3: Digitally controlled oscillator (DCO) schematic.

TPL853_02993475-84, Fig. 26.10.3 (ISSCC 2004 / Session 26 / *Optical and Fast IO 26.10*) (CX-0170).

EXHIBIT G-1 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,809,336 By LG



TPL853_02993485-96, Fig. 3 (CX-0171). "Figs. 17 and 18 show the die micrographs of two Digital PLLs implemented in 65 nm and 45 nm CMOS. . . . Both use a simple three stage ring oscillator shown in Fig. 3(a)." TPL853_02993494.

The TI OMAP4 family of processors is also implemented in 45 nm CMOS technology: "The OMAP4470 high-performance multimedia application device is based on enhanced OMAP architecture and uses 45-nm technology." TPLBN043746 (OMAP4470 Technical Ref Manual). *See also* TPLBN037535 (OMAP4430 Tech Ref Manual).

Furthermore, on information and belief, each of the OMAP3 and OMAP4 chips at issue contains a DPLL that outputs a clock signal for the MPU. Within each of these DPLLs is either a single-ended or differential ring oscillator. This component of the PLL generates an oscillation (*i.e.*, a clock signal). The ring oscillator is able to produce an oscillation due to the presence of an odd number of inversions arranged in a loop. *See also HTC*

EXHIBIT G-1 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,809,336 By LG

	Corporation v. Technology Properties Limited, 08-cv-00882-PSG, Trial Tr. 341:13-347:1, 333:20-374:16 (Haroun testimony). On information and belief, the only input to the ring oscillators in the OMAP3 and OMAP4 chips is a current. In addition, while the bias DAC ("digital to analog converter") in the DPLL may receive a digital word, it does not pass this control signal on to the ring oscillator. Indeed, without the bias component of the PLL, the ring oscillators in the OMAP chips would still output an oscillation. Barring the application of dividers to the ring oscillator's output, the only way to change the frequency of the ring oscillator is to change the input current. In other words, the ring oscillator will always generate a clock signal as long as a current is applied to it.

EXHIBIT G-1 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,809,336 By LG

Claim 13	
13. A microprocessor system comprising:	*See* Claim Element 6.a.
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;	*See* Claim Element 6.b.
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,	*See* Claim Element 6.c.
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,	*See* Claim Element 6.d.
thereby enabling said processing frequency to track said clock rate in response to said parameter variation;	*See* Claim Element 6.e.
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	*See* Claim Element 6.f.
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	*See* Claim Element 6.g.
wherein said central processing unit operates asynchronously to said input/output interface.	*See* Claim Element 6.h.

Claim 14

EXHIBIT G-1 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,809,336 By LG

14. A microprocessor system of claim 13	*See* Claim Element 6.a.
wherein said one or more operational parameters include operating temperature of said substrate or operating voltage of said substrate.	*See* Claim Elements 6.d, 6.e.

Claim 15	
15. A microprocessor system of claim 13	*See* Claim Element 6.a.
wherein said oscillator comprises a ring oscillator.	*See* Claim Element 6.c, and Claim 9.

EXHIBIT G-2

DECLARATION OF OLGA I. MAY IN SUPPORT OF DEFENDANTS LG ELECTRONICS, INC. AND LG ELECTRONICS U.S.A., INC.'S MOTION TO STRIKE INFRINGEMENT CONTENTIONS OR, ALTERNATIVELY, COMPEL SUPPLEMENTAL INFRINGEMENT CONTENTIONS ON CHARTED PROCESSORS

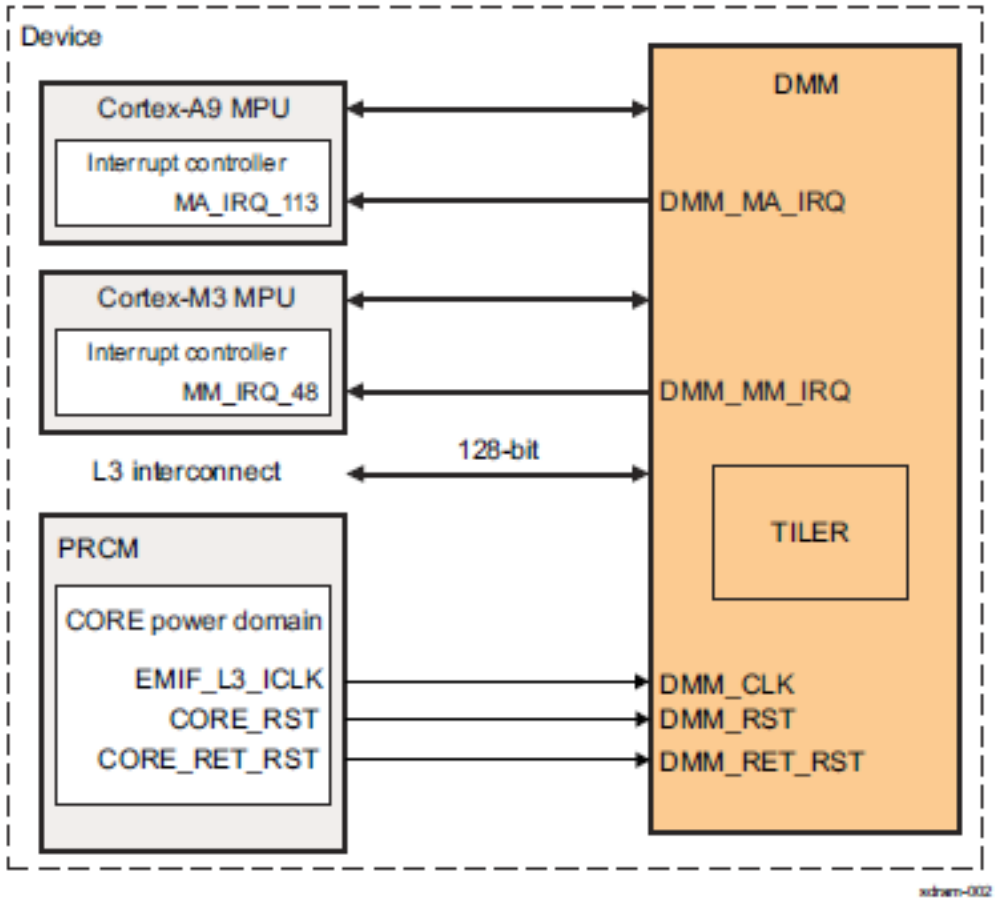
EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Claim 1	
A microprocessor, which comprises	On information and belief, each Accused Product listed in the attached list of Accused Products (Ex. A.6) contains a microprocessor ("Accused Microprocessors").[1] *See* Ex. A for listings of microprocessors in the Accused Products, information obtained from http://www.phonescoop.com/; http://pdadb.net/; http://www.gsmarena.com/. Each microprocessor is an electronic circuit that interprets and executes programmed instructions.
a main central processing unit and	As discussed in the attached list of Accused Products, the Accused Microprocessors contain a first ARM core. *See also*, http://en.wikipedia.org/wiki/Qualcomm_Snapdragon; http://en.wikipedia.org/wiki/OMAP.
a separate direct memory access central processing unit in a single integrated circuit comprising said microprocessor,	On information and belief, the Accused Microprocessors contain a second central processing unit separate from the first ARM core. On information and belief, the second central processing unit is able to access memory and fetch and execute instructions separately and independently of the main central processing unit. For example, many of the Accused Microprocessors are multicore processors that contain more than one ARM core.

[1] Infringement by the Accused Products is largely based on the operation of and implementation of the microprocessors they contain. This chart provides some examples of such operation that, on information and belief, are representative of the operation of the processors in each of the Accused Products. Discovery is in the early stages, and Plaintiffs anticipate receiving additional documents showing the exact operation of the processor in each of the Accused Products with respect to the accused functionality. But because many documents that Plaintiffs would rely on to establish infringement are confidential and have not yet been produced in this litigation, Plaintiffs anticipate receiving additional documents to confirm the operational principles shown in this chart from Defendants and/or third parties. Accordingly, Plaintiffs reserve the right to amend, supplement, or augment their claim charts, infringement contentions, or infringement theories based on documents and information later received through discovery.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

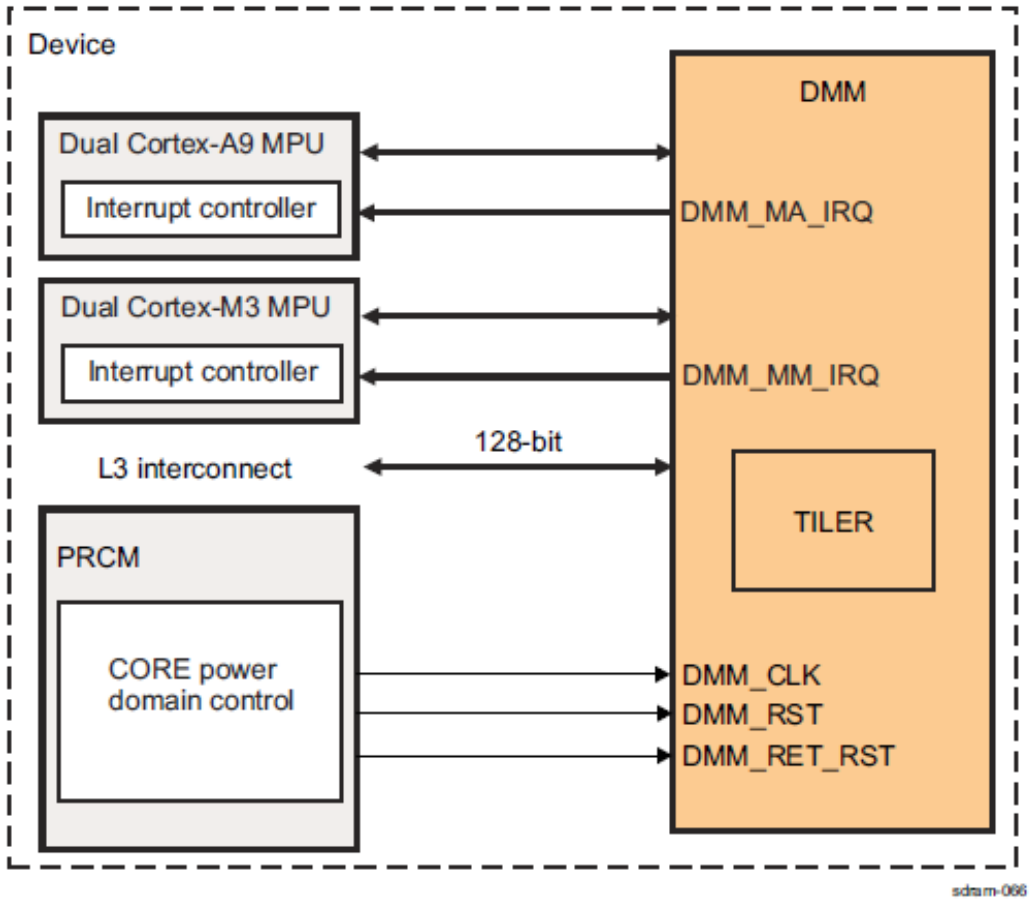
Figure 15-3. DMM Integration



OMAP4430 Technical Reference Manual, p. 3213.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Figure 16-2. DMM and Tiler Overview



OMAP4470 Technical Reference Manual, p. 3308.

In addition, many of the Accused Microprocessors contain one or more DSPs that are able to

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

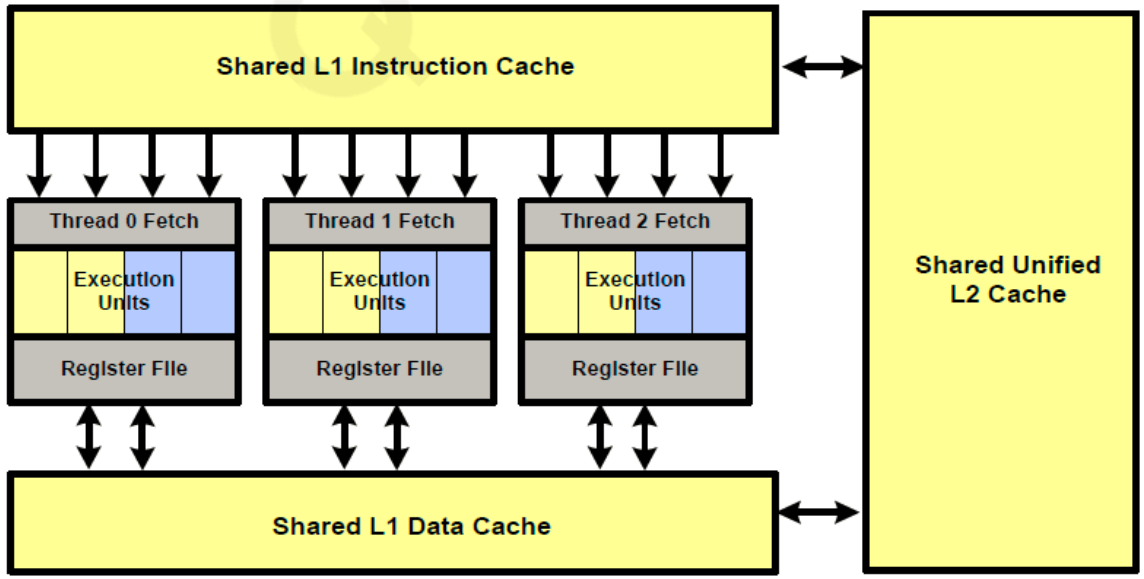
	access memory and fetch and execute instructions separately and independently of the main central processing unit.  Hexagon Programming: Processor Overview, p. 10.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

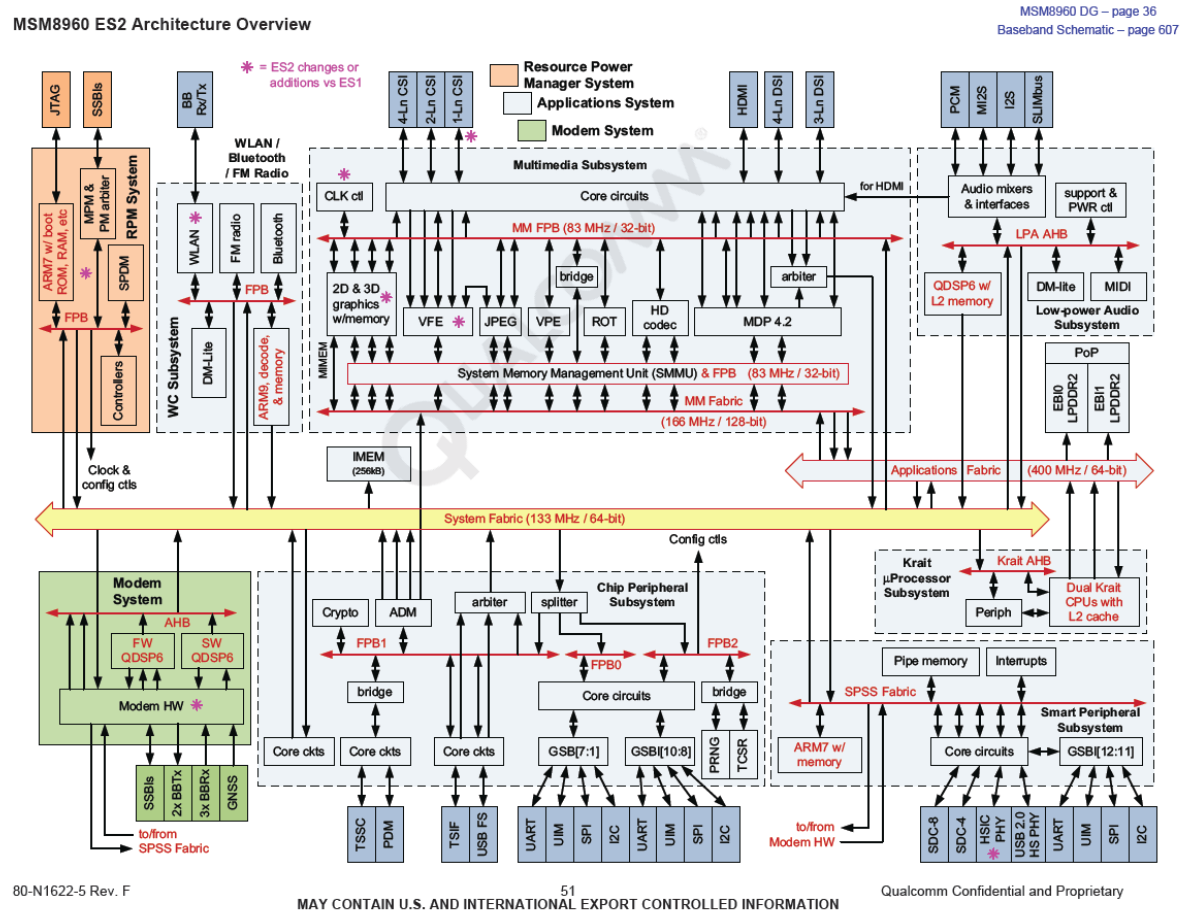
	 MSM8960 Chipset Schematics and Design Guidelines, p. 50. *See also.*, http://en.wikipedia.org/wiki/Qualcomm_Snapdragon; http://en.wikipedia.org/wiki/OMAP.
said main central processing unit	ARM cores, such as those contained within the Accused Microprocessors, contain arithmetic

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

having an arithmetic logic unit,	logic units. The ARM instruction provides control over the Arithmetic Logic Unit (ALU) and confirms that the ALU within the CPU performs arithmetic and logic operations on data. In addition, the ARM architecture provides: • control over both the *Arithmetic Logic Unit* (ALU) and shifter in most data-processing instructions to maximize the use of an ALU and a shifter ARMv7 Architecture Reference Manual [TPLBN051517-TPLBN052654], p. A1-2. **Arithmetic/logic instructions** The following arithmetic/logic instructions share a common instruction format. These perform an arithmetic or logical operation on up to two source operands, and write the result to a destination register. They can also optionally update the condition code flags, based on the result. ARM Architecture Reference Manual, p. A1-7
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,	The ARM architecture defines how an ARM core must operate. ARM Architectures, Processors, and Devices [TPLBN051505-TPLBN051516], pp. 1-3. The ARMv7-A architecture is backwards compatible with the ARMv6 architecture. ARM Architectures, Processors, and Devices, pp. 1-3, 1-4. The ARM instructions provide control over the Arithmetic Logic Unit (ALU) and confirm that the ALU within the CPU performs arithmetic and logic operations on data. In addition, the ARM architecture provides: • control over both the *Arithmetic Logic Unit* (ALU) and shifter in most data-processing instructions to maximize the use of an ALU and a shifter ARM Architecture Reference Manual [TPLBN051517-TPLBN052654], p. A1-2. **Arithmetic/logic instructions** The following arithmetic/logic instructions share a common instruction format. These perform an arithmetic or logical operation on up to two source operands, and write the result to a destination register. They can also optionally update the condition code flags, based on the result.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

ARM Architecture Reference Manual, p. A1-7

The Arithmetic Logic Unit (ALU) derives its two inputs from the 'Top item' (Rn) and the 'Next item' (Rm) of the 'Push Down Stack' from the 'General Purpose Registers' (the holding place for items placed there by stack operations) and directs its output (Rd) back to the 'General Purpose Registers'.

Assembler syntax

ADC{S}<c><q> {<Rd>,} <Rn>, <Rm>, <type> <Rs>

where:

S — If S is present, the instruction updates the flags. Otherwise, the flags are not updated.

<c><q> — See *Standard assembler syntax fields* on page A8-7.

<Rd> — The destination register.

<Rn> — The first operand register.

<Rm> — The register that is shifted and used as the second operand.

<type> — The type of shift to apply to the value read from <Rm>. It must be one of:

- ASR — Arithmetic shift right, encoded as type = 0b10
- LSL — Logical shift left, encoded as type = 0b00
- LSR — Logical shift right, encoded as type = 0b01
- ROR — Rotate right, encoded as type = 0b11.

<Rs> — The register whose bottom byte contains the amount to shift by.

The pre-UAL syntax ADC<c>S is equivalent to ADCS<c>.

ARMv7 Architecture Reference Manual [TPLBN049289-TPLBN051446], p. A8-19.

The Register File includes register R13, also known as SP or the Stack pointer, which is a - pointer to the active stack.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

SP, the Stack Pointer

Register R13 is used as a pointer to the active stack.

In Thumb code, most instructions cannot access SP. The only instructions that can access SP are those designed to use SP as a stack pointer.

The use of SP for any purpose other than as a stack pointer is deprecated.

ARMv7 Architecture Reference Manual, p. A2-11.

The top two items in the stack are connected to provide inputs into the ALU by using a "POP".

A8.6.122 POP

Pop Multiple Registers loads multiple registers from the stack, loading from consecutive memory locations starting at the address in SP, and updates SP to point just above the loaded data.

ARMv7 Architecture Reference Manual, p. A8-246.

The output of the ALU is connected to the top of the stack and may be saved with a "PUSH."

A8.6.123 PUSH

Push Multiple Registers stores multiple registers to the stack, storing to consecutive memory locations ending just below the address in SP, and updates SP to point to the start of the stored data.

ARMv7 Architecture Reference Manual, p. A8-249.

In the alternative, ARM Cortex A9 processor also includes an operand stack for Jazelle DBX instructions.

A2.10.2 Jazelle direct bytecode execution support

From ARMv5TEJ, the architecture requires every system to include an implementation of the Jazelle extension. The Jazelle extension provides architectural support for hardware acceleration of bytecode execution by a *Java Virtual Machine* (JVM).

ARMv7 Architecture Reference Manual, p. A2-73.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Reads

Reads are defined as memory operations that have the semantics of a load.

The memory accesses of the following instructions are reads:

- LDR, LDRB, LDRH, LDRSB, and LDRSH
- LDRT, LDRBT, LDRHT, LDRSBT, and LDRSHT
- LDREX, LDREXB, LDREXD, and LDREXH
- LDM, LDRD, POP, and RFE
- LDC, LDC2, VLDM, VLDR, VLD1, VLD2, VLD3, and VLD4
- the return of status values by STREX, STREXB, STREXD, and STREXH
- in the ARM instruction set only, SWP and SWPB
- in the Thumb instruction set only, TBB and TBH.

Hardware-accelerated opcode execution by the Jazelle extension can cause a number of reads to occur, according to the state of the operand stack and the implementation of the Jazelle hardware acceleration.

Writes

Writes are defined as memory operations that have the semantics of a store.

The memory accesses of the following instructions are Writes:

- STR, STRB, and STRH
- STRT, STRBT, and STRHT
- STREX, STREXB, STREXD, and STREXH
- STM, STRD, PUSH, and SRS
- STC, STC2, VSTM, VSTR, VST1, VST2, VST3, and VST4
- in the ARM instruction set only, SWP and SWPB.

Hardware-accelerated opcode execution by the Jazelle extension can cause a number of writes to occur, according to the state of the operand stack and the implementation of the Jazelle hardware acceleration.

ARMv7 Architecture Reference Manual, p. A3-42.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

	The operand stack contains a top item register and a next item register connected to provide inputs to the ALU and the output of the ALU is connected to the top item register. The first four elements of the stack are held in the register file in registers R0-R3. The stack pointer is held in register R6. In Java State, the processor assigns several ARM registers to functions specific to the Java machine (e.g., R6= stack pointer, R0-R3= top elements of stack, R4=local variable 0). This hardware reuse contributes to the small size of the additional logic (12k gates) required to implement the Java machine, and keeps all of the state required by the Jazelle extension in ARM registers, In addition, it ensures compatibility with existing operating systems, interrupt handlers and exception code. ARM White Paper, Accelerating to Meet the Challenges of Embedded Java™ [TPLBN051478-TPLBN051482] at 3. Plaintiffs contend that this claim element is literally present as described above. In the event that this claim element is not found to be literally present, Plaintiffs contend that the above identified stacks are equivalent to "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" and any differences are insubstantial. In particular, the stacks perform the same function (*i.e.*, provide input and output to the ALU), in substantially the same way (*i.e.*, by providing a last-in, first-out data structure), and have the same result (*i.e.*, the ALU performs operations on inputs and provides output).
said top item register also being connected to provide inputs to an internal data bus,	The top item in the stack of an ARM core is connected to provide input to an internal data bus by using a "POP" which loads it into a register.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

	A8.6.122 POP Pop Multiple Registers loads multiple registers from the stack, loading from consecutive memory locations starting at the address in SP, and updates SP to point just above the loaded data. ARMv7 Architecture Reference Manual, p. A8-246. The top item of the stack is also located in the data cache within the "memory system." On information and belief, the register file provides inputs to and is bidirectionally connected to internal buses (e.g., L3 Interconnect, AXI/AHB Bus, Applications/System Fabric)..

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

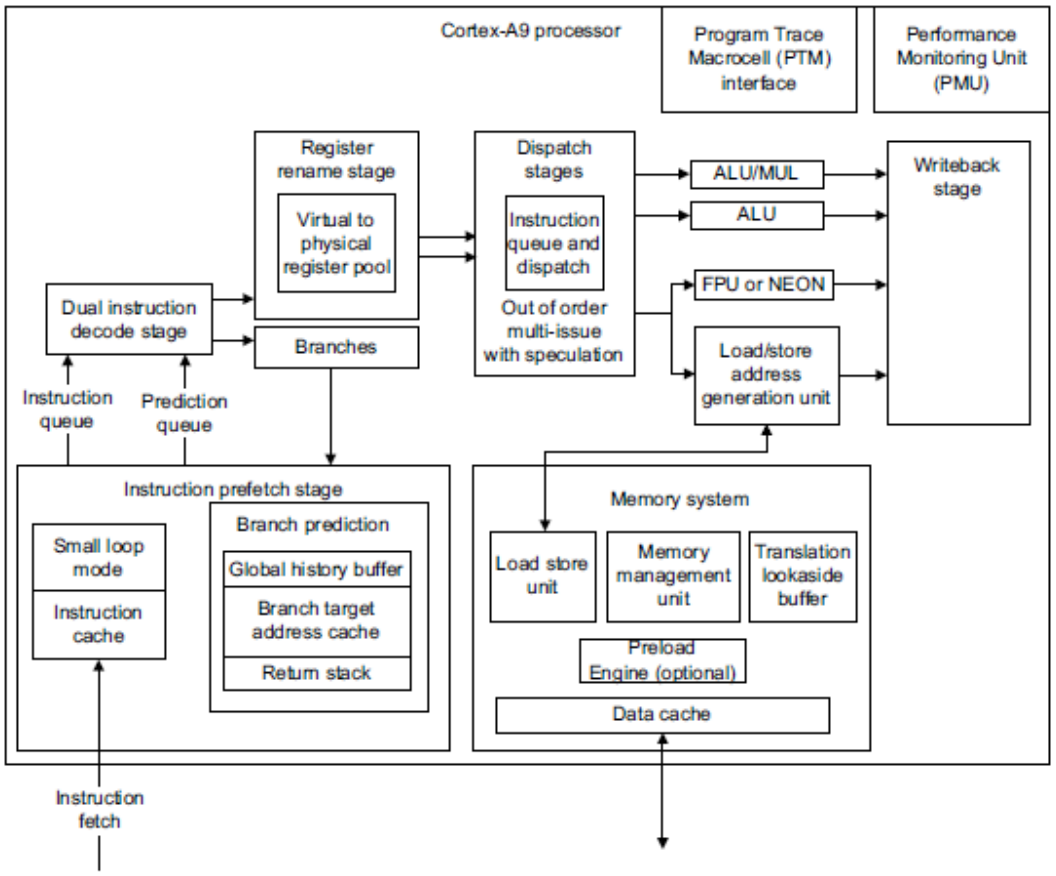
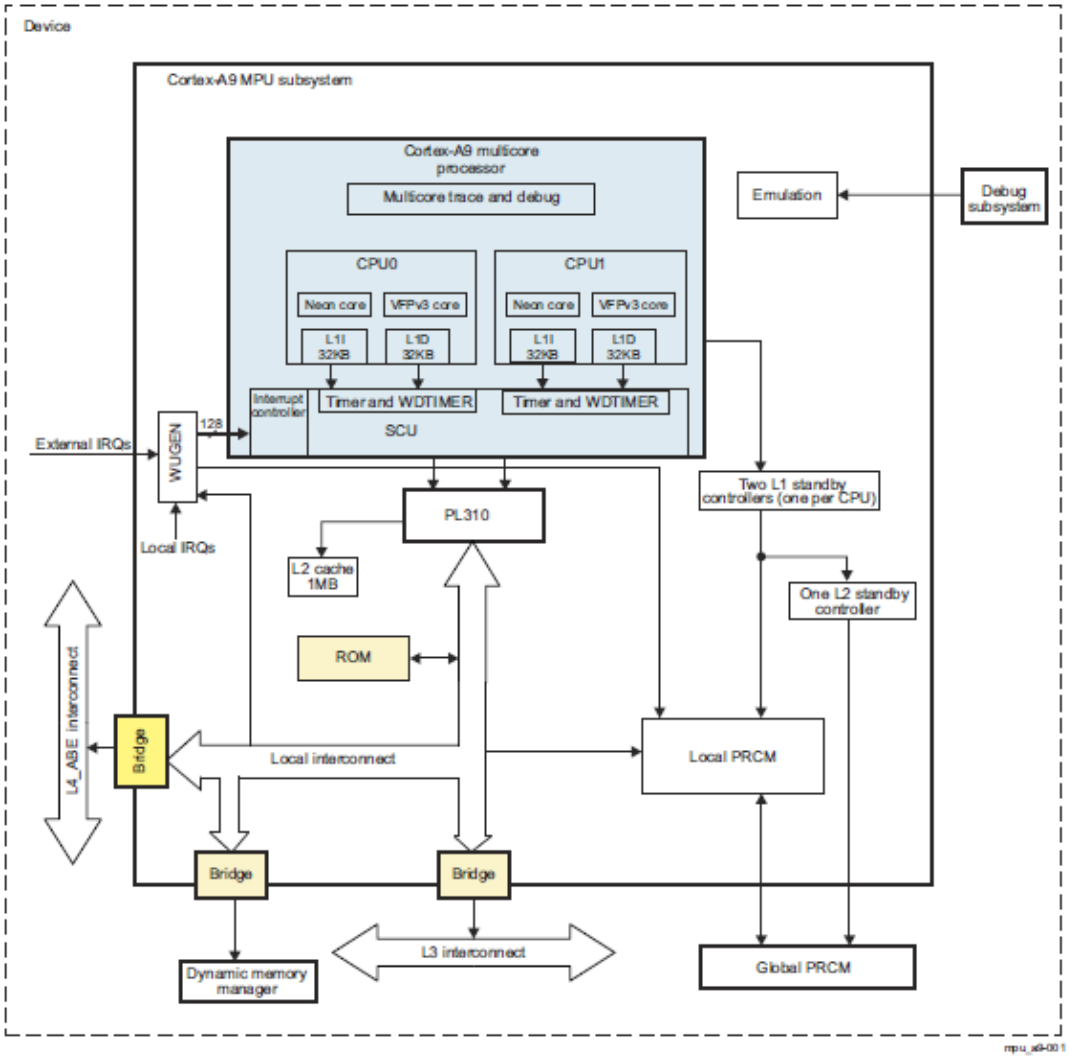


Figure 2-1 Cortex-A9 processor top-level diagram

Cortex-A9 Technical Reference Manual, p.2-2.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

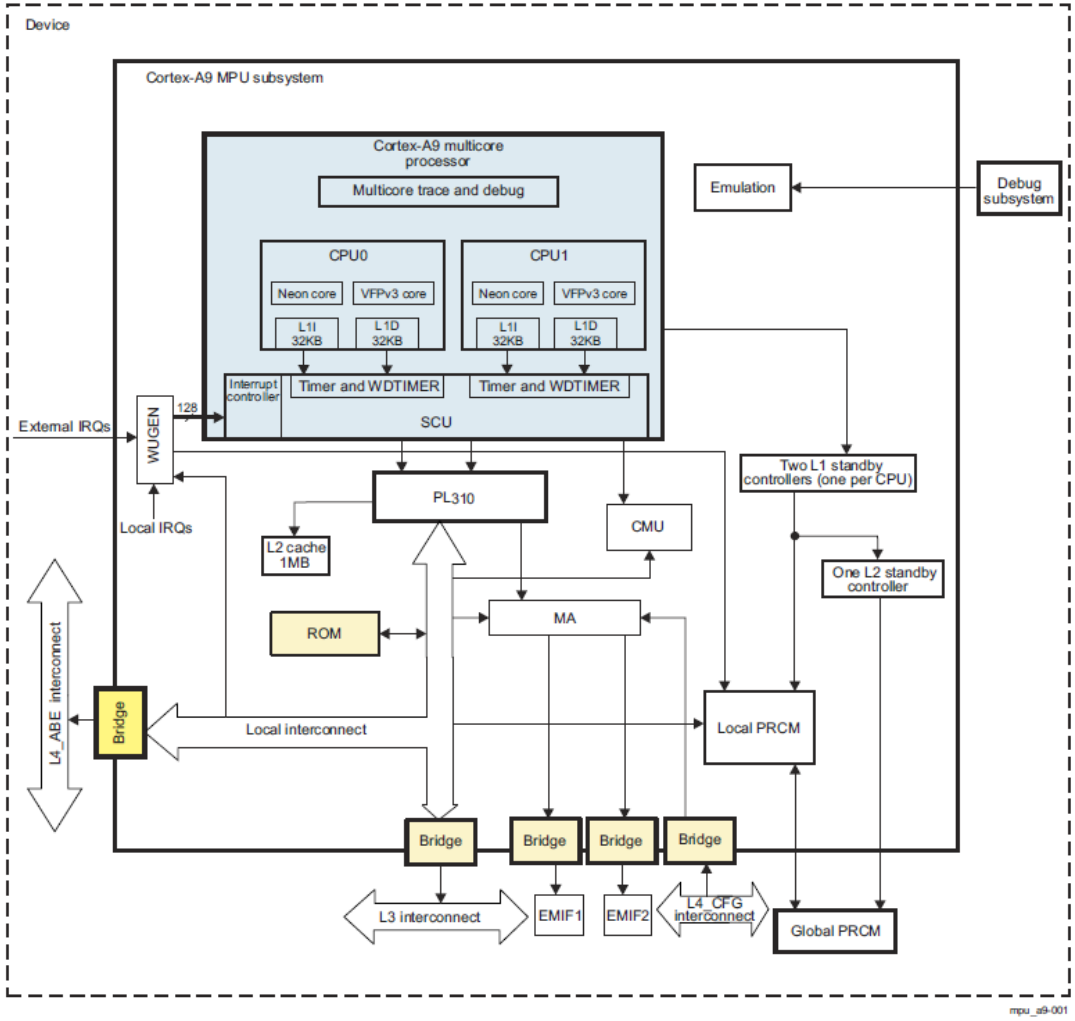
Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4430 Technical Reference Manual, p. 1077.

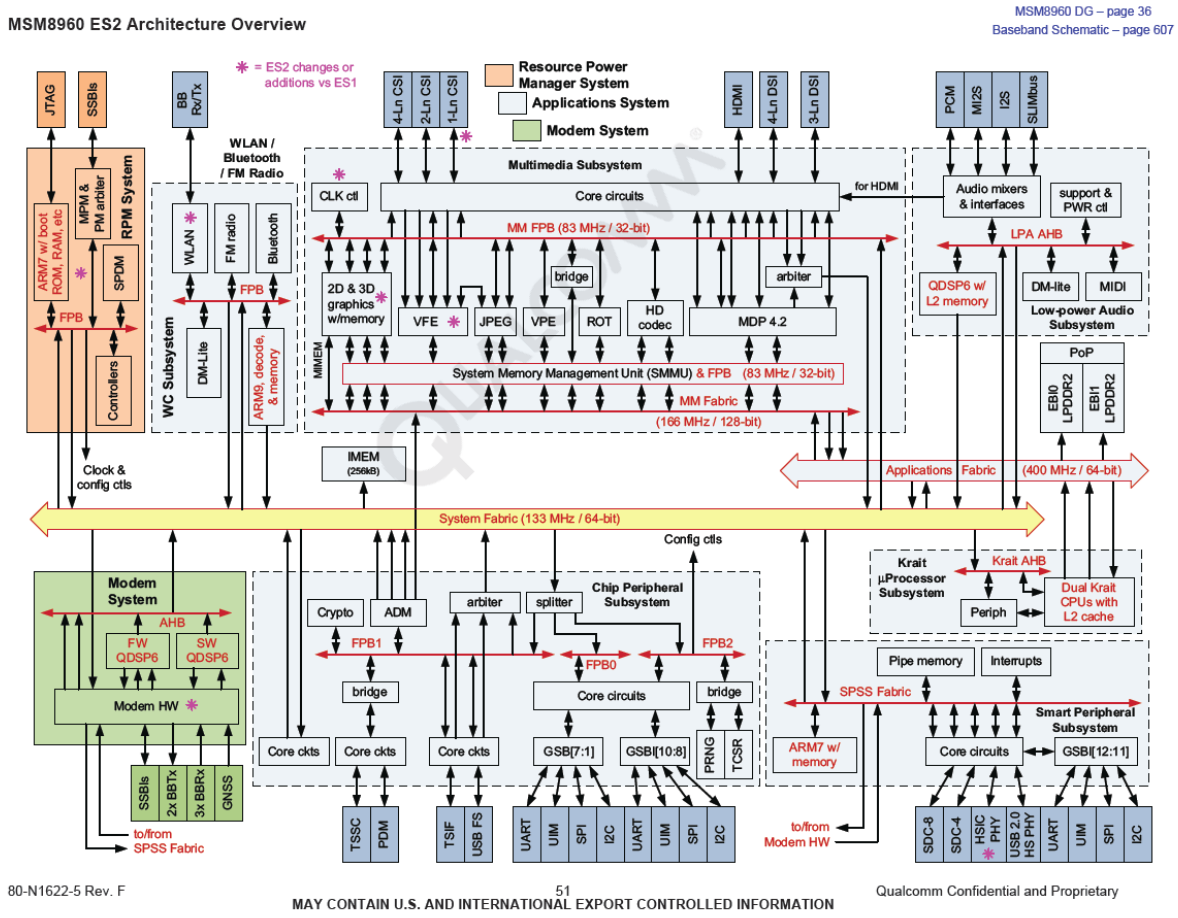
EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4470 Technical Reference Manual, p. 1091.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG



MSM8960 Chipset Schematics and Design Guidelines, p. 50.

In the alternative, when executing Jazelle code, the top item and next item, located in registers R0-R4 are bidirectionally connected to the internal buses.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

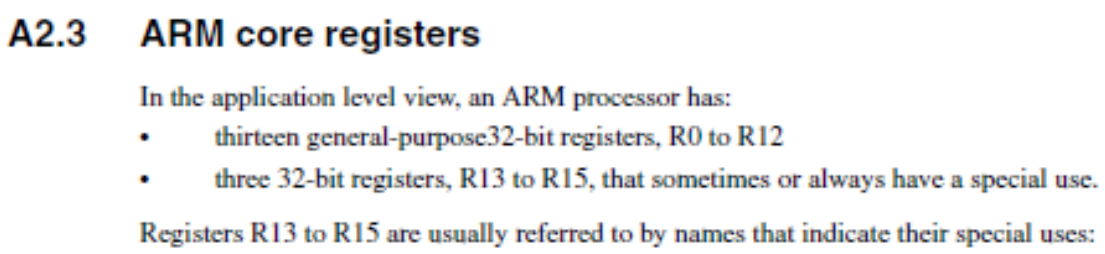
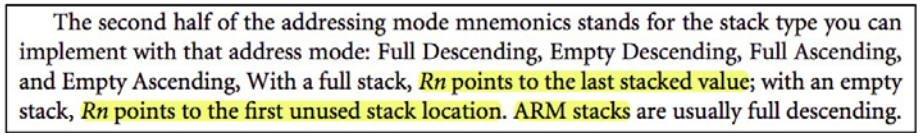
	Plaintiffs contend that this claim element is literally present as described above. In the event that this claim element is not found to be literally present, Plaintiffs contend that the above identified buses, either alone or in combination, are equivalent to "an internal data bus" and any differences are insubstantial. In particular, the buses perform the same function (*i.e.*, provide data connections between the ALU, memory, and registers), in substantially the same way (*i.e.*, by connecting them electrically), and have the same result (*i.e.*, the various connected components transmit data between them).
said internal data bus being bidirectionally connected to a loop counter,	The ARM core contains general purpose registers. For example, in ARMv7 cores there are thirteen general purpose registers R0-R12.  **A2.3 ARM core registers** In the application level view, an ARM processor has: • thirteen general-purpose32-bit registers, R0 to R12 • three 32-bit registers, R13 to R15, that sometimes or always have a special use. Registers R13 to R15 are usually referred to by names that indicate their special uses: ARMv7 Architecture Reference Manual, p. A8-246. Any of the general purpose registers is a loop counter when the register is incremented or decremented and is combined with a conditional branch instruction.  The second half of the addressing mode mnemonics stands for the stack type you can implement with that address mode: Full Descending, Empty Descending, Full Ascending, and Empty Ascending, With a full stack, *Rn* points to the last stacked value; with an empty stack, *Rn* points to the first unused stack location. ARM stacks are usually full descending. ARM System Developer's Guide, A. N. Sloss, Morgan Kaufmann [TPLBN052655-TPLBN053357 p. 585.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

The register file provides inputs to and is bidirectionally connected to internal buses (e.g., L3 Interconnect, AXI/AHB Bus, Applications/System Fabric).

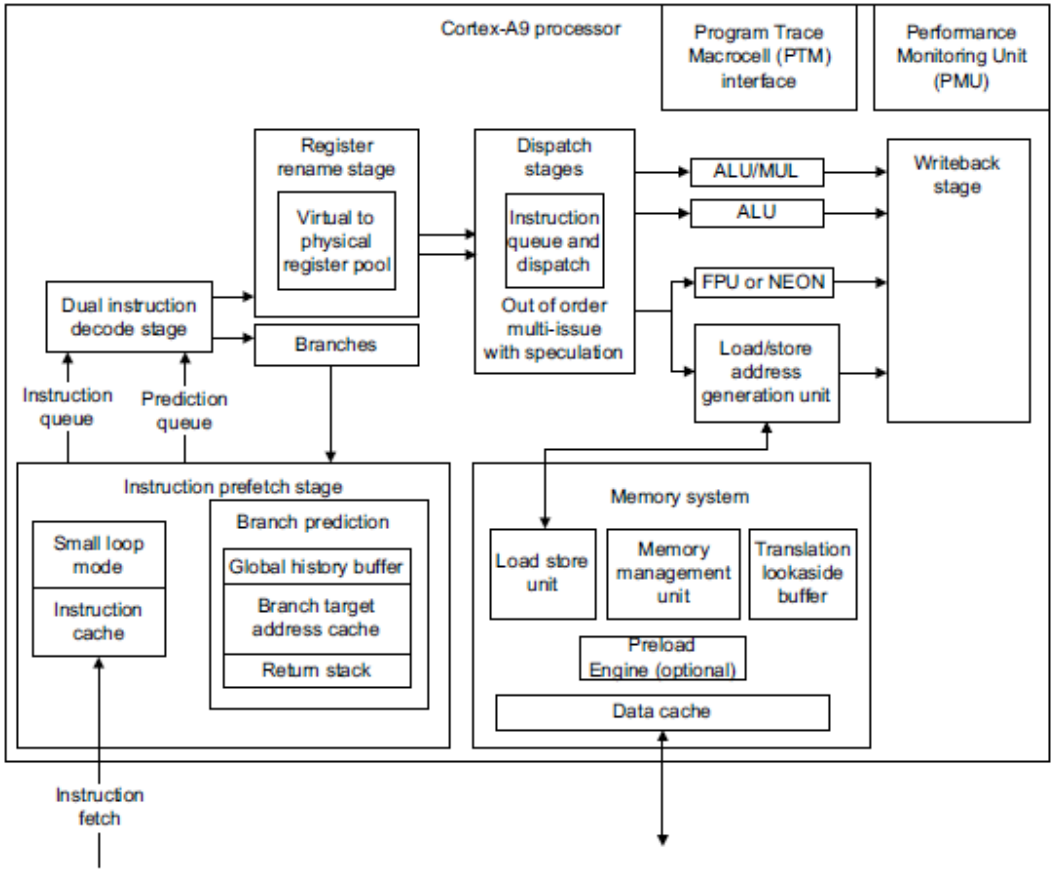
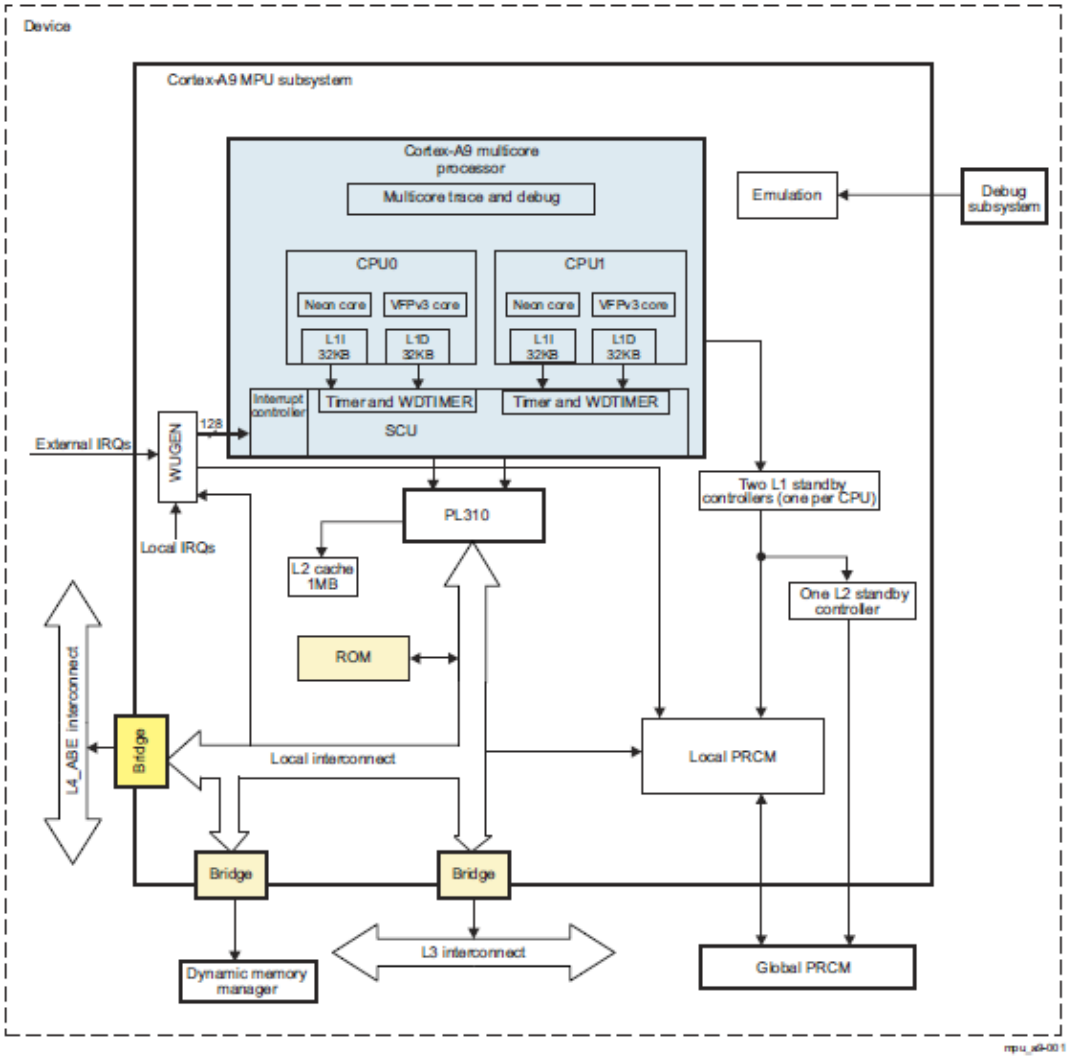


Figure 2-1 Cortex-A9 processor top-level diagram

Cortex-A9 Technical Reference Manual, p.2-2.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

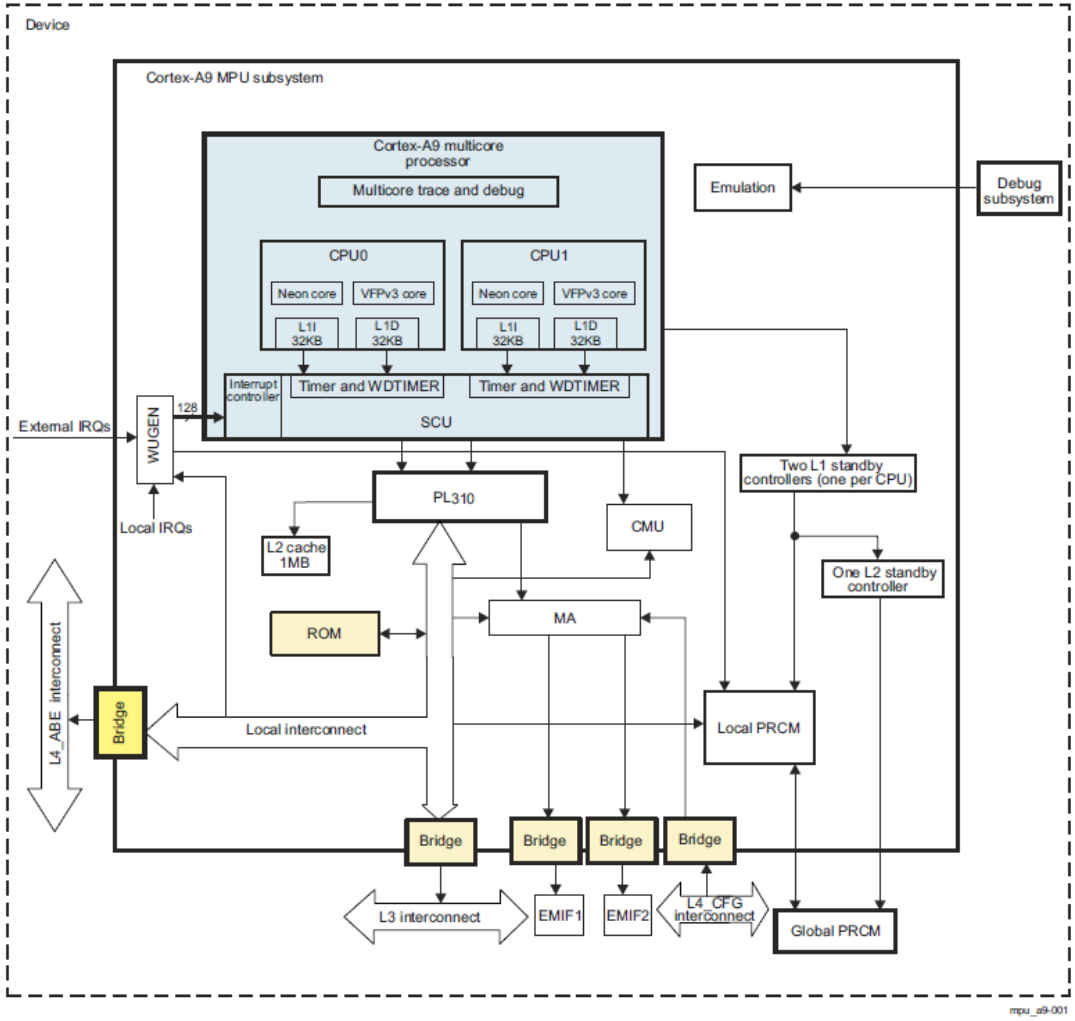
Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4430 Technical Reference Manual, p. 1077.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4470 Technical Reference Manual, p. 1091.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

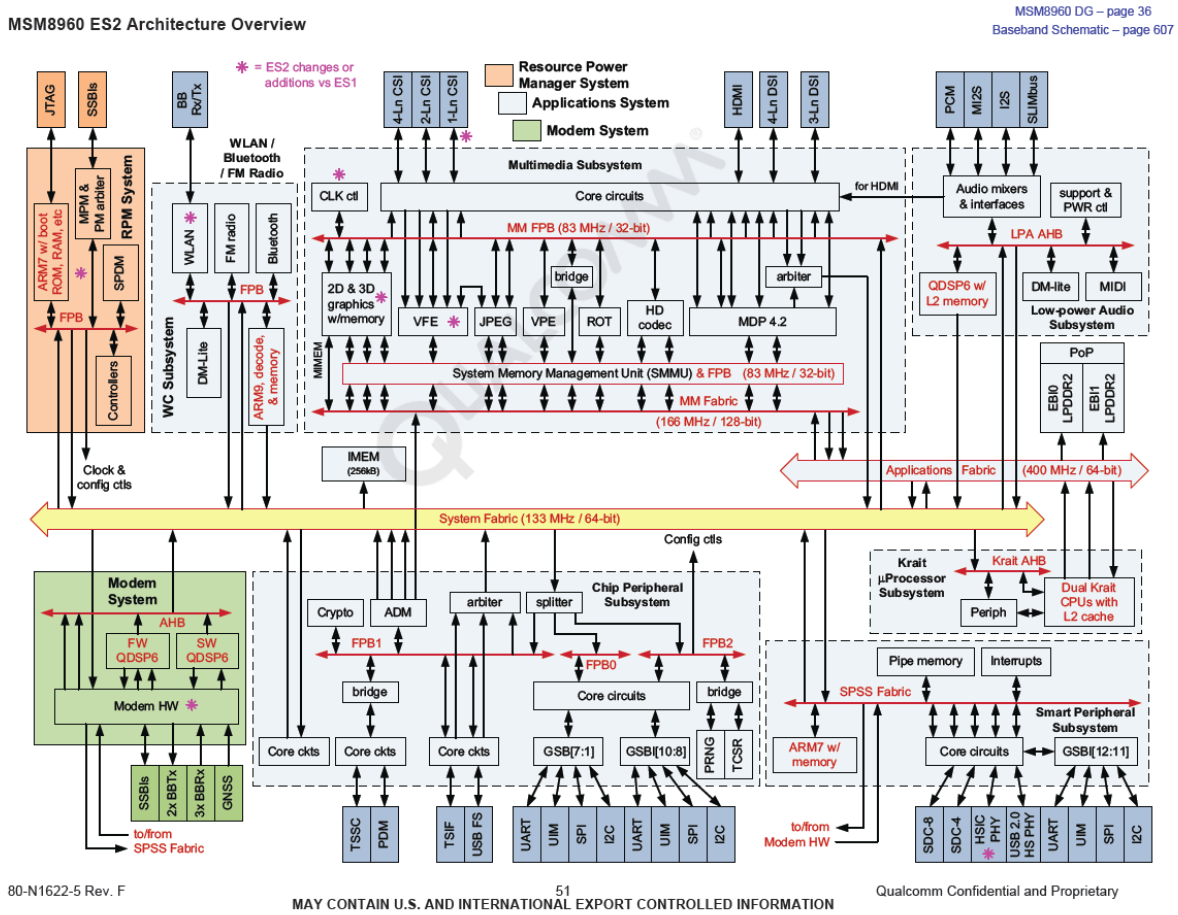
	 MSM8960 Chipset Schematics and Design Guidelines, p. 50. On information and belief, when executing Jazelle instructions within a loop, the processor maintains a loop counter that is bidirectionally connected to the Accused Microprocessor's internal buses.
said loop counter being connected to	Each ARM core contains a general purpose register or other hardware is used as a loop

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

a decrementer,	counter, the loop counter is connected to an ALU or an adder, which serves as a decrementer. For example, in ARMv7 cores there are thirteen general purpose registers R0-R12. **A2.3 ARM core registers** In the application level view, an ARM processor has: • thirteen general-purpose32-bit registers, R0 to R12 • three 32-bit registers, R13 to R15, that sometimes or always have a special use. Registers R13 to R15 are usually referred to by names that indicate their special uses: ARMv7 Architecture Reference Manual, p. A8-246. Any of the general purpose registers is a loop counter when the register is incremented or decremented and is combined with a conditional branch instruction. The second half of the addressing mode mnemonics stands for the stack type you can implement with that address mode: Full Descending, Empty Descending, Full Ascending, and Empty Ascending, With a full stack, *Rn* points to the last stacked value; with an empty stack, *Rn* points to the first unused stack location. ARM stacks are usually full descending. ARM System Developer's Guide, A. N. Sloss, Morgan Kaufmann [TPLBN052655-TPLBN053357 p. 585. The register file provides inputs to and is bidirectionally connected to internal buses (e.g., L3 Interconnect, AXI/AHB Bus, Applications/System Fabric).

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

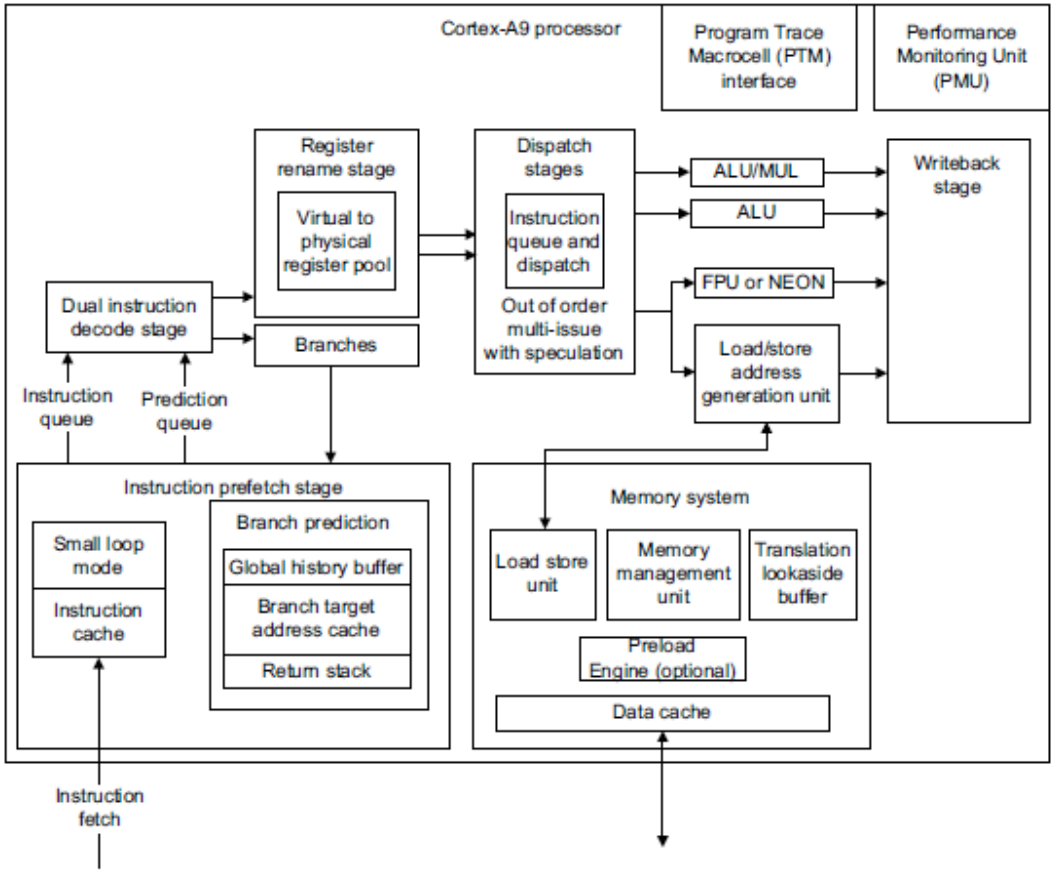
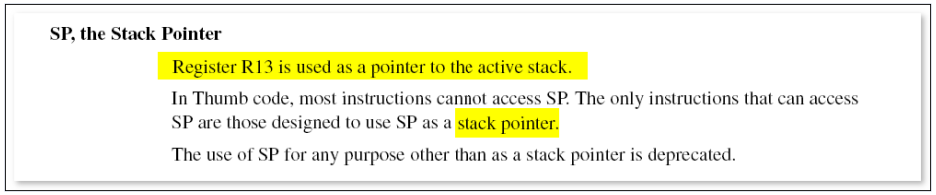
	 **Figure 2-1 Cortex-A9 processor top-level diagram** Cortex-A9 Technical Reference Manual, p.2-2.
said internal data bus being bidirectionally connected to a stack pointer, return stack pointer, mode register and instruction register,	Each ARM core has a stack pointer.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG



SP, the Stack Pointer

Register R13 is used as a pointer to the active stack.

In Thumb code, most instructions cannot access SP. The only instructions that can access SP are those designed to use SP as a stack pointer.

The use of SP for any purpose other than as a stack pointer is deprecated.

ARM Architecture Reference Manual - ARMv7-A and ARMv7-R p. A2-11

On information and belief, the Register File of each Accused Microprocessor is bidirectionally connected to the data bus. The Register File includes Register R13, also known as SP or the stack pointer, which points to the active stack and Register R14, also known as LR, the Link Register, which contains an identical memory address as the top of the return stack.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

SP, the Stack Pointer

Register R13 is used as a pointer to the active stack.

In Thumb code, most instructions cannot access SP. The only instructions that can access SP are those designed to use SP as a stack pointer.

The use of SP for any purpose other than as a stack pointer is deprecated.

Note

Using SP for any purpose other than as a stack pointer is likely to break the requirements of operating systems, debuggers, and other software systems, causing them to malfunction.

LR, the Link Register

Register R14 is used to store the return address from a subroutine. At other times, LR can be used for other purposes.

When a `BL` or `BLX` instruction performs a subroutine call, LR is set to the subroutine return address. To perform a subroutine return, copy LR back to the program counter. This is typically done in one of two ways, after entering the subroutine with a `BL` or `BLX` instruction:

- Return with a `BX LR` instruction.
- On subroutine entry, store LR to the stack with an instruction of the form:
  `PUSH {<registers>,LR}`
  and use a matching instruction to return:
  `POP {<registers>,PC}`

ThumbEE checks and handler calls use LR in a similar way. For details see Chapter A9 *ThumbEE*.

ARMv7 Architecture Reference Manual, p. A2-11.

On information and belief, one or more instruction registers receive instructions from the instruction cache.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

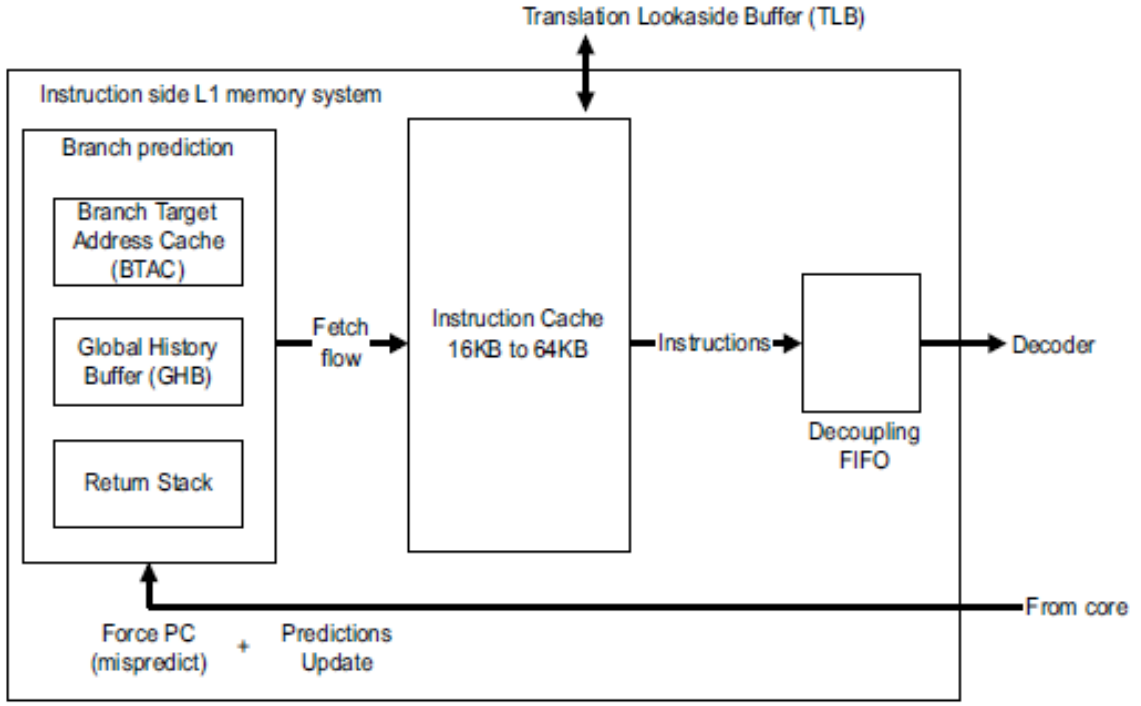


Figure 6-1 Branch prediction and instruction cache

Cortex-A9 Technical Reference Manual p. 6-5.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Figure 2-1 shows a top-level functional diagram of the Cortex-A15 MPCore processor.

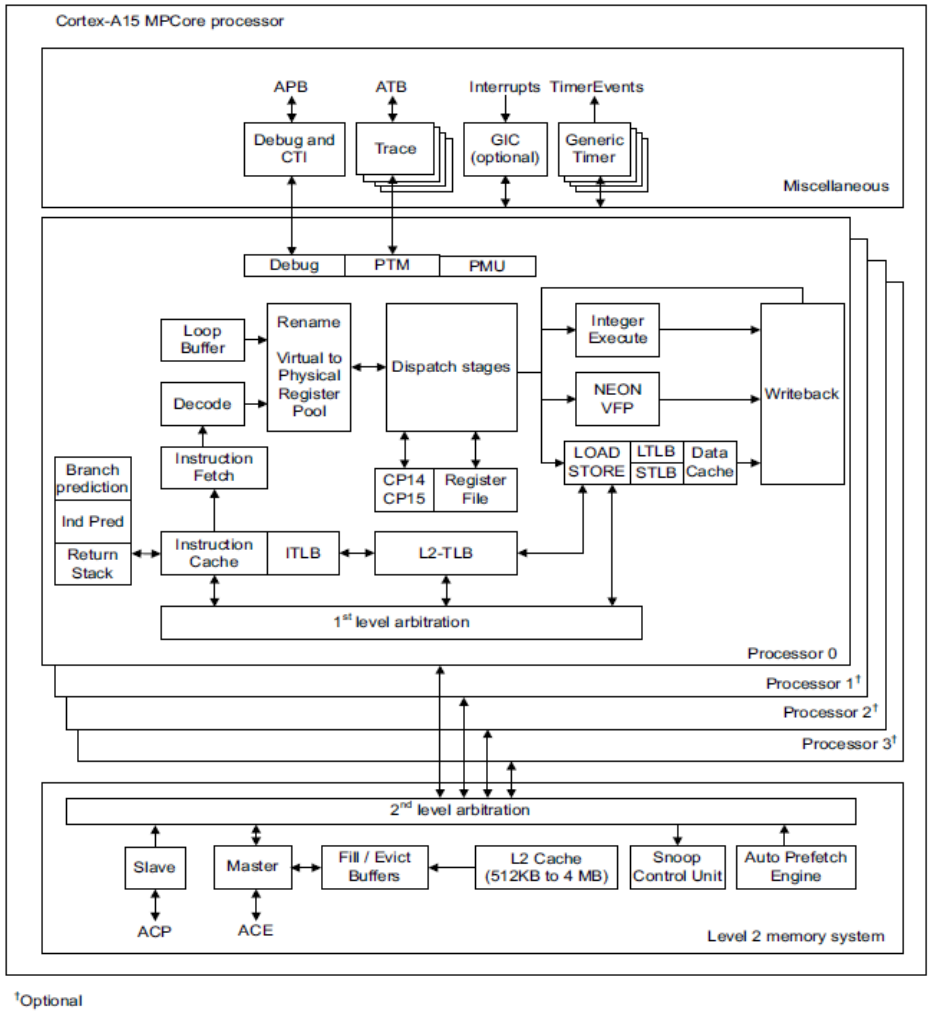


Figure 2-1 Block diagram

Cortex-A15 MPCore Processor Technical Reference Manual p. 2-2.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

	Instruction fetch The instruction fetch unit fetches instructions from the L1 instruction cache and delivers up to three instructions per cycle to the instruction decode unit. It supports dynamic and static branch prediction. The instruction fetch unit includes: • L1 instruction cache that is a 32KB 2-way set-associative cache with 64 bytes cache line and optional parity protection per 16-bits. • 2-level dynamic predictor with BTB for fast target generation. • Return stack. • Static branch predictor. • Indirect predictor. • 32-entry fully-associative L1 instruction TLB. Cortex-A15 MPCore Processor Technical Reference Manual p. 2-3. The ARM core contains a number of mode registers within its system control coprocessors. *See, e.g.*, Cortex-A9 Technical Reference Manual Ch. 4; Cortex-A15 MPCore Processor Technical Reference Manual Ch. 4. On information and belief, the register file provides inputs to and is bidirectionally connected to internal buses (e.g., L3 Interconnect, AXI/AHB Bus, Applications/System Fabric.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

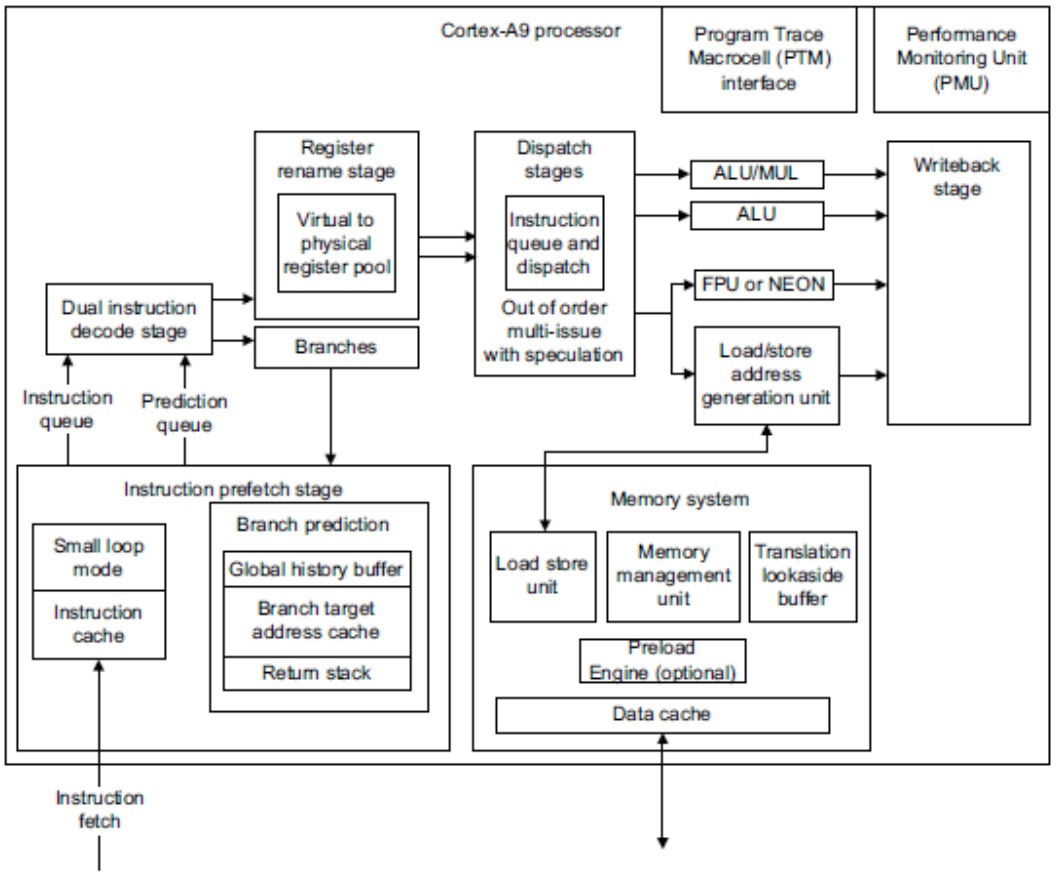
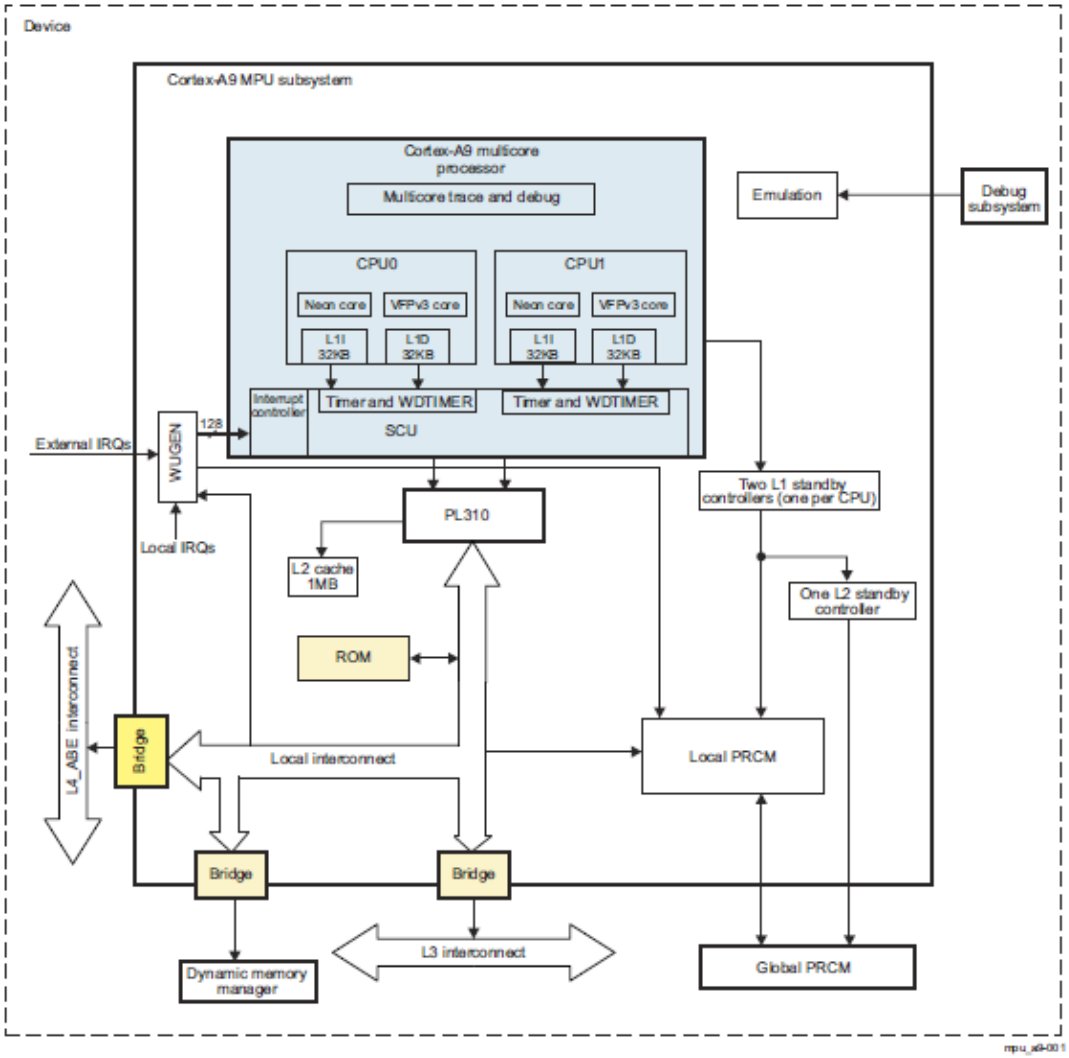


Figure 2-1 Cortex-A9 processor top-level diagram

Cortex-A9 Technical Reference Manual, p.2-2.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

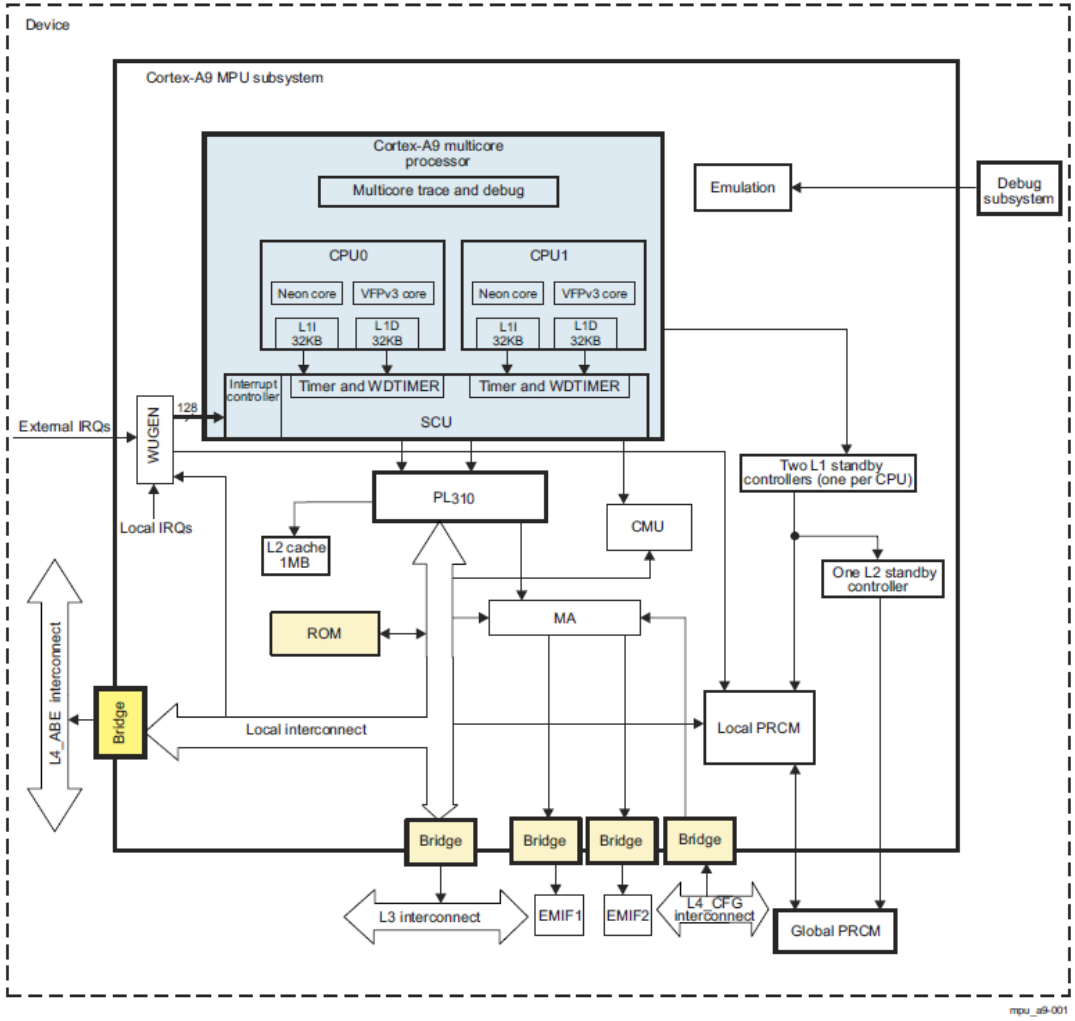
Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4430 Technical Reference Manual, p. 1077.

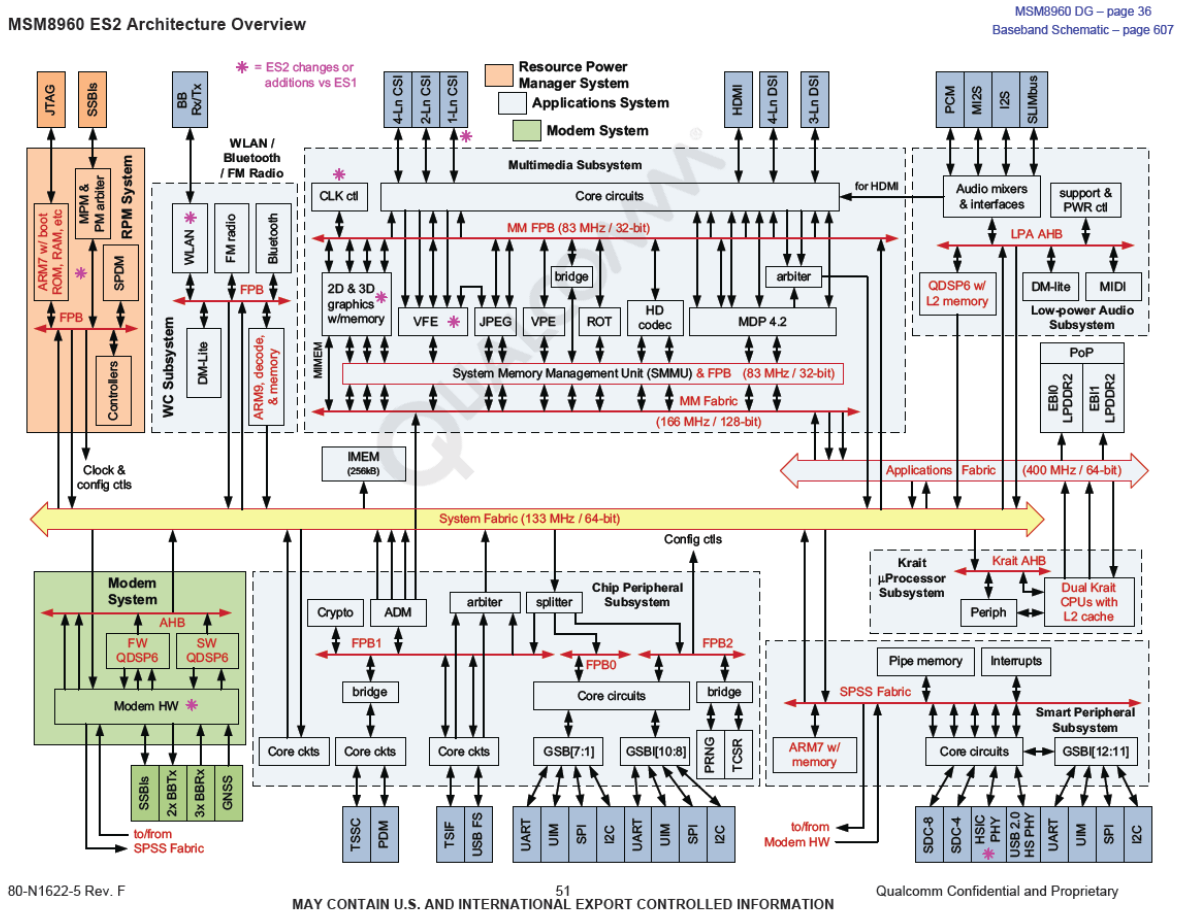
EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4470 Technical Reference Manual, p. 1091.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG



MSM8960 Chipset Schematics and Design Guidelines, p. 50.

When executing Jazelle instructions, the stack pointer is located in register R6.

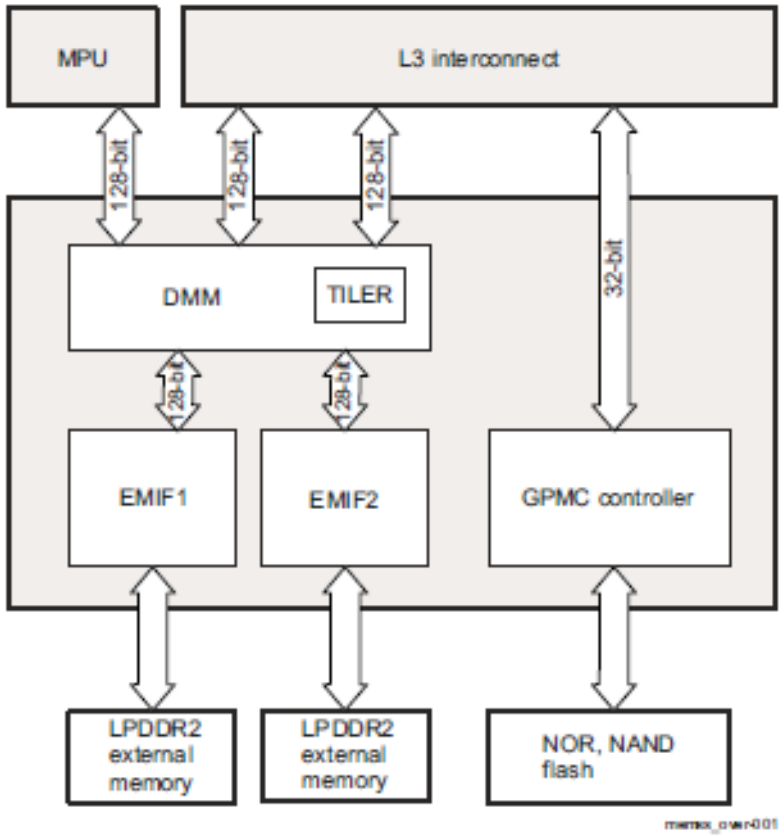
EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

	In Java State, the processor assigns several ARM registers to functions specific to the Java machine (e.g., R6= stack pointer, R0-R3= top elements of stack, R4=local variable 0). This hardware reuse contributes to the small size of the additional logic (12k gates) required to implement the Java machine, and keeps all of the state required by the Jazelle extension in ARM registers, In addition, it ensures compatibility with existing operating systems, interrupt handlers and exception code. ARM White Paper, Accelerating to Meet the Challenges of Embedded Java™ at 3.
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,	On information and belief, the internal buses of the Accused Microprocessors are connected to a memory controller.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Figure 15-1 shows a functional diagram of all memory subsystems in the device.

Figure 15-1. Memory Subsystem Functional Diagram

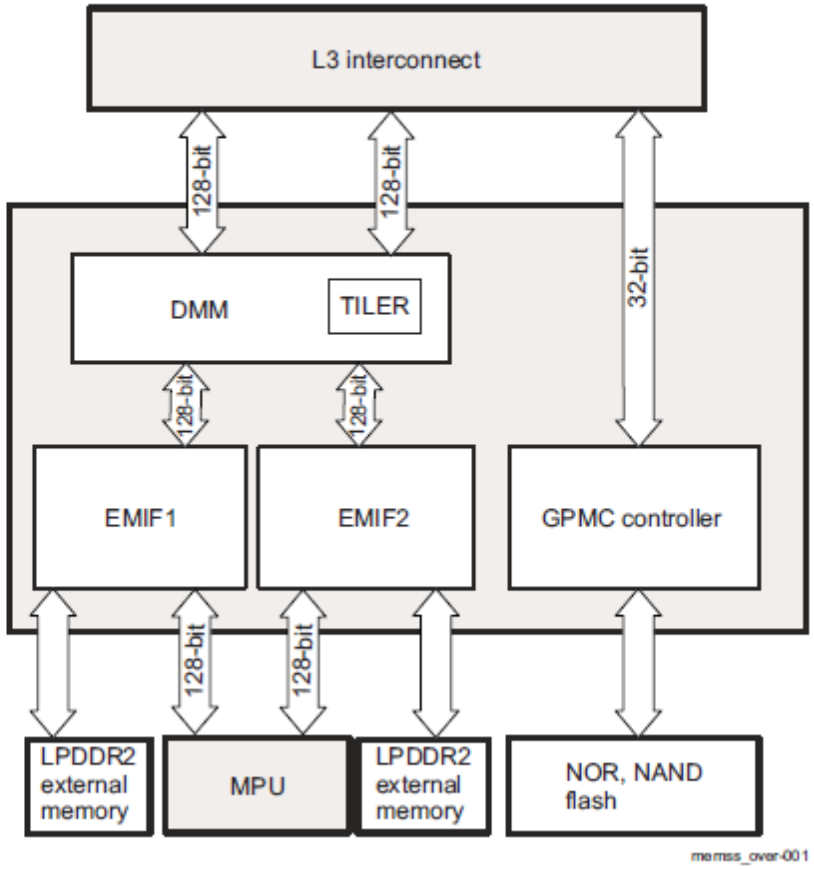


OMAP4430 Technical Reference Manual, p. 3207.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

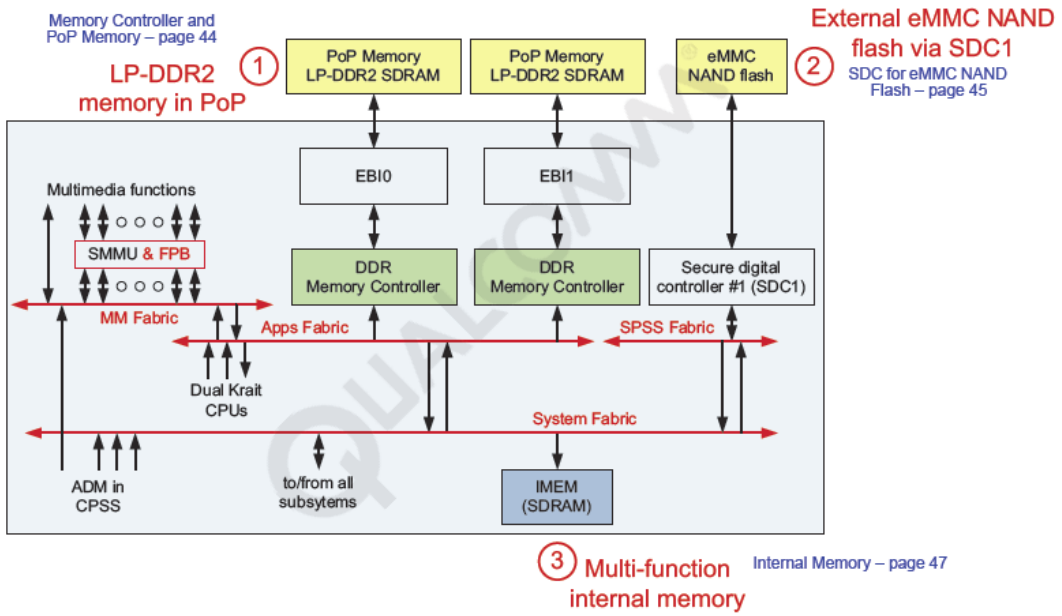
Figure 16-1 shows a functional diagram of all memory subsystems in the device.

Figure 16-1. Memory Subsystem Functional Diagram



OMAP4470 Technical Reference Manual, p. 3302.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG



MSM8960 Chipset Schematics and Design Guidelines, p. 41.

On information and belief, the internal buses are connected to a register of the return stack which serves as a Y register. The internal buses are also connected to the Register File which contains several general purpose registers that may serve as an X register and the program counter.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

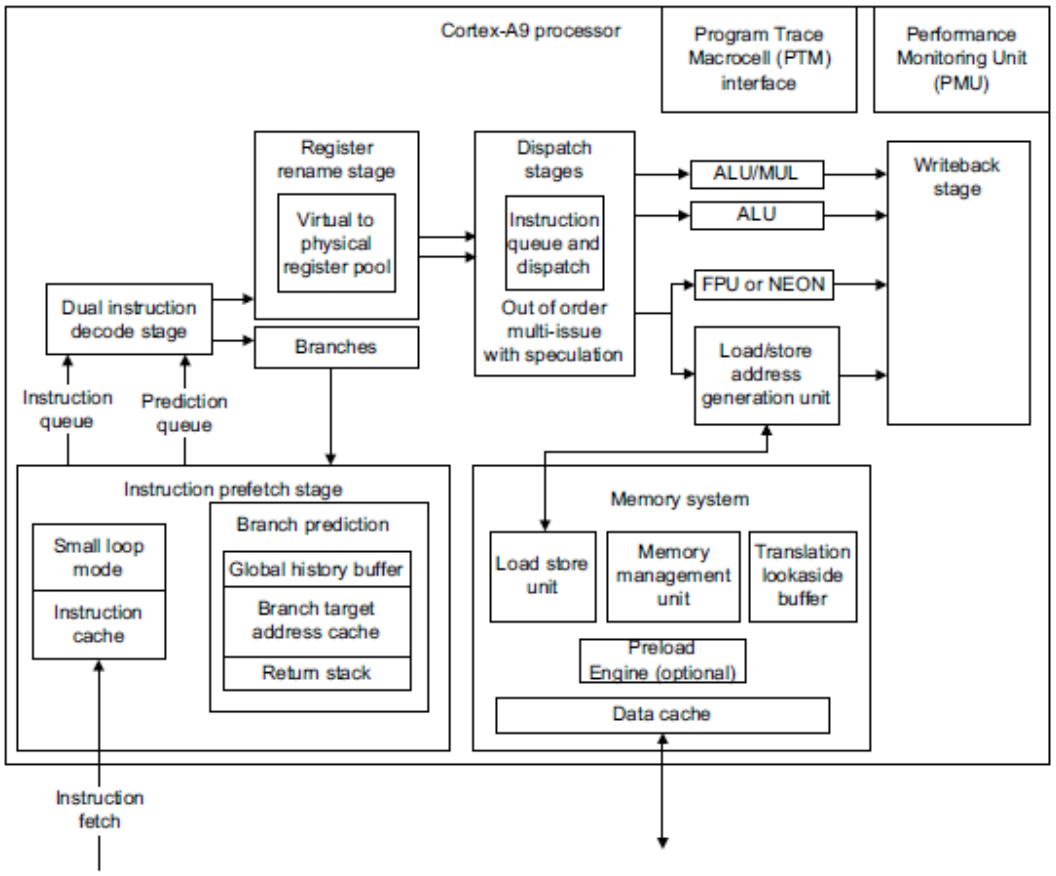


Figure 2-1 Cortex-A9 processor top-level diagram

Cortex-A9 Technical Reference Manual, p.2-2.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

	PC, the Program Counter Register R15 is the program counter: • When executing an ARM instruction, PC reads as the address of the current instruction plus 8. • When executing a Thumb instruction, PC reads as the address of the current instruction plus 4. • Writing an address to PC causes a branch to that address. In Thumb code, most instructions cannot access PC. ARMv7 Architecture Reference Manual, p. A2-11.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Figure 2-1 shows a top-level functional diagram of the Cortex-A15 MPCore processor.

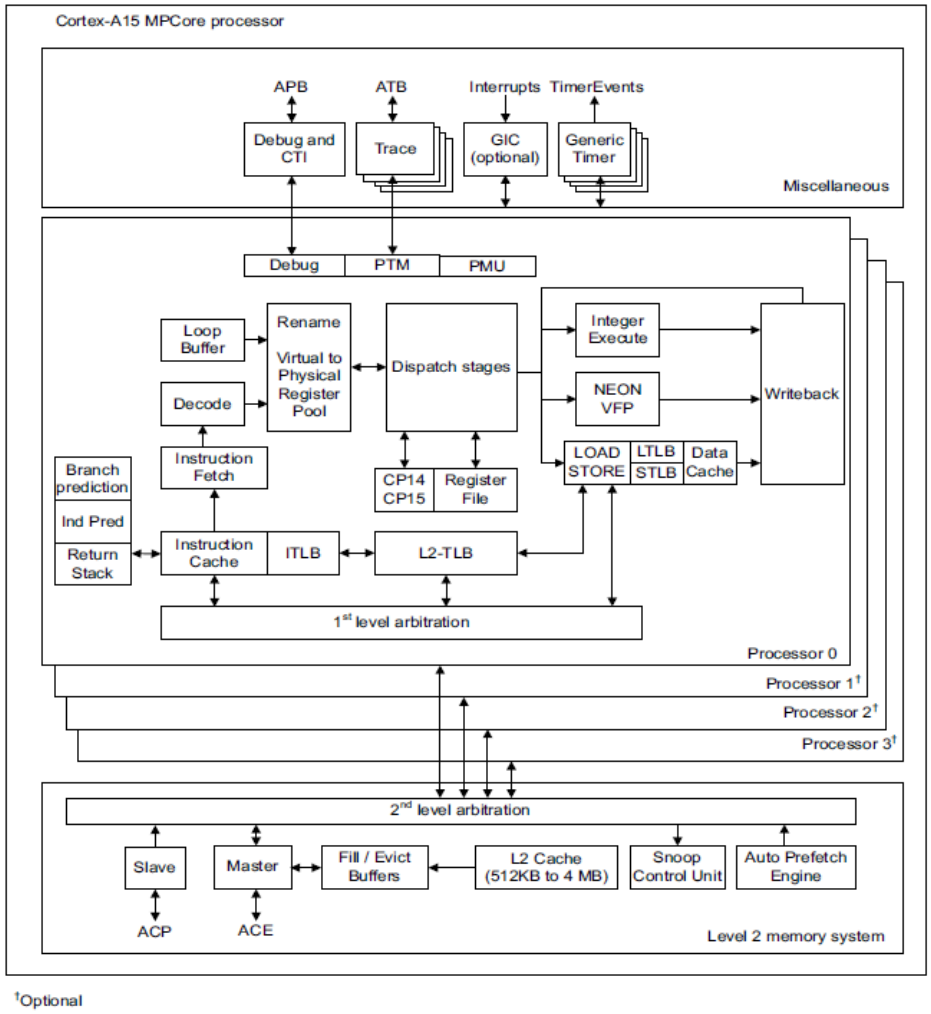
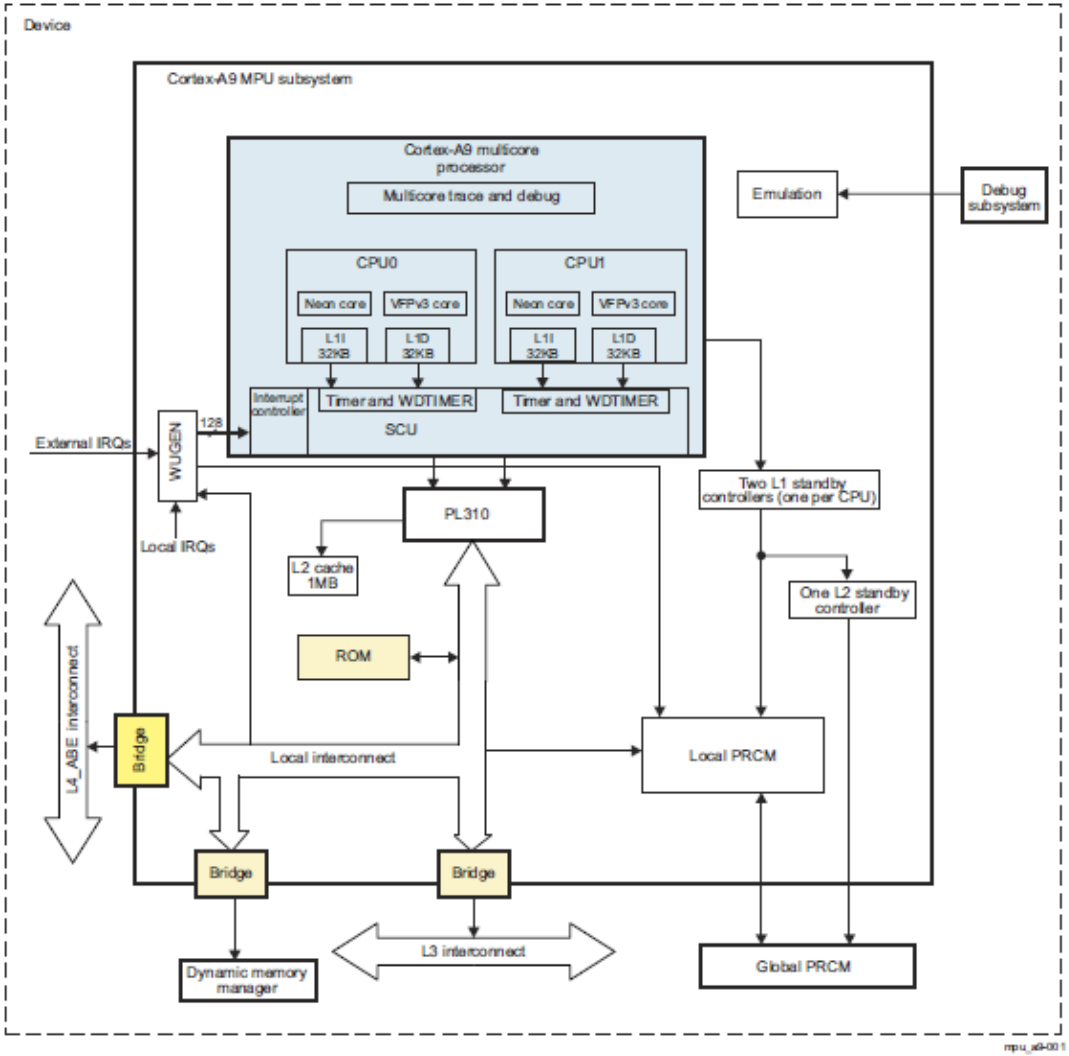


Figure 2-1 Block diagram

Cortex-A15 MPCore Processor Technical Reference Manual p. 2-2.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

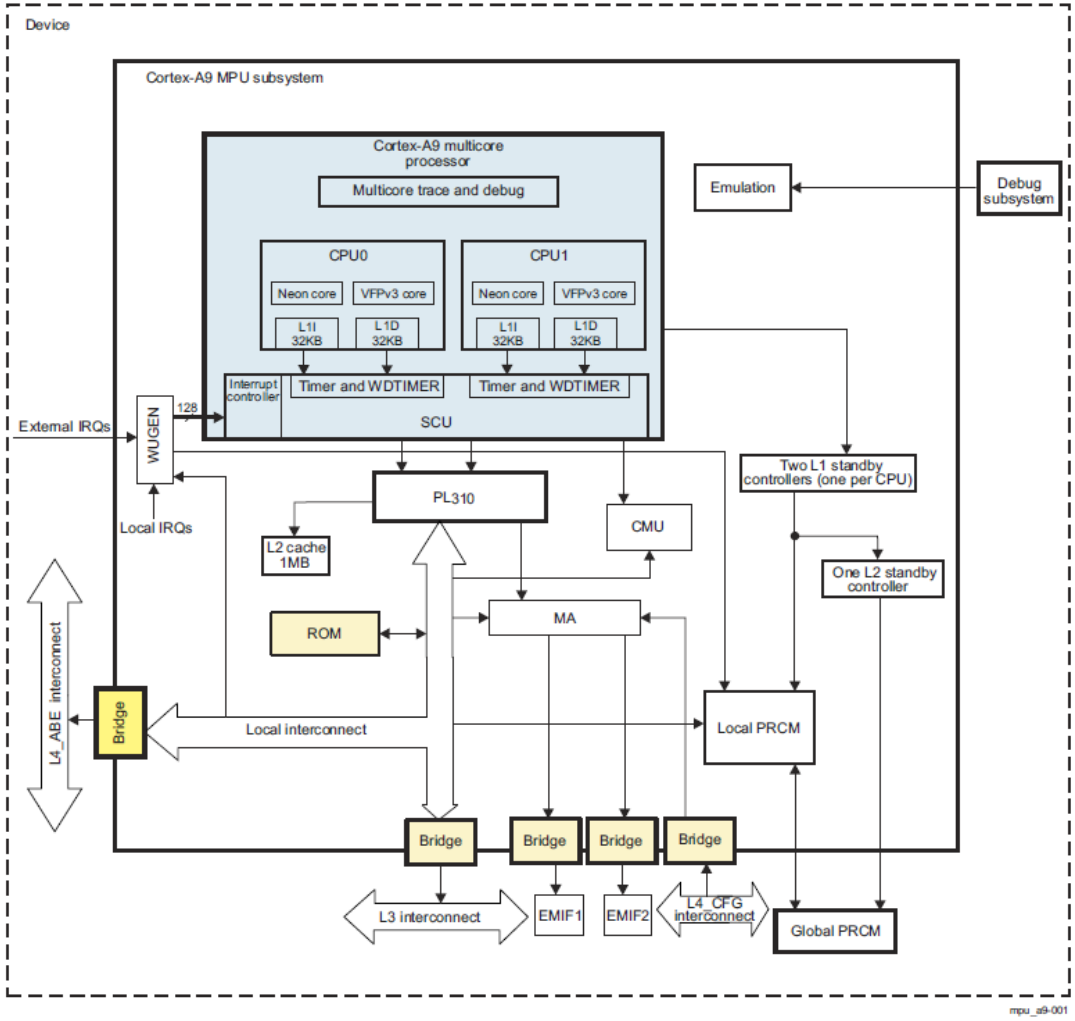
Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4430 Technical Reference Manual, p. 1077.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4470 Technical Reference Manual, p. 1091.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

	When executing Jazelle instructions, register R4 may be used as an X register. In Java State, the processor assigns several ARM registers to functions specific to the Java machine (e.g., R6= stack pointer, R0-R3= top elements of stack, R4=local variable 0). This hardware reuse contributes to the small size of the additional logic (12k gates) required to implement the Java machine, and keeps all of the state required by the Jazelle extension in ARM registers. In addition, it ensures compatibility with existing operating systems, interrupt handlers and exception code. ARM White Paper, Accelerating to Meet the Challenges of Embedded Java™ at 3.
said Y register, X register and program counter providing outputs to an internal address bus,	On information and belief, the program counter, X register, and Y register provide outputs to the internal buses of the Accused Microprocessors. Plaintiff contends that this claim element is literally present. However, in the event that this claim element is not found to be literally present, Plaintiffs contend that the above identified buses, either alone or in combination, are equivalent to "an internal address bus" and any differences are insubstantial. In particular, the buses perform the same function (*i.e.*, provide a means to communicate addresses between the ALU, memory, and registers), in substantially the same way (*i.e.*, by connecting them electrically), and have the same result (*i.e.*, the various connected components transmit addresses to each other).
said internal address bus providing inputs to said memory controller and to an incrementer,	On information and belief, the internal buses of the Accused Microprocessors provide inputs to the memory controller to provide access to RAM. In the case that the Program Counter, X register, or Y register holds an address pointing to external memory, the internal buses provides the address to the memory controller. The internal buses also provide input to an incrementer.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

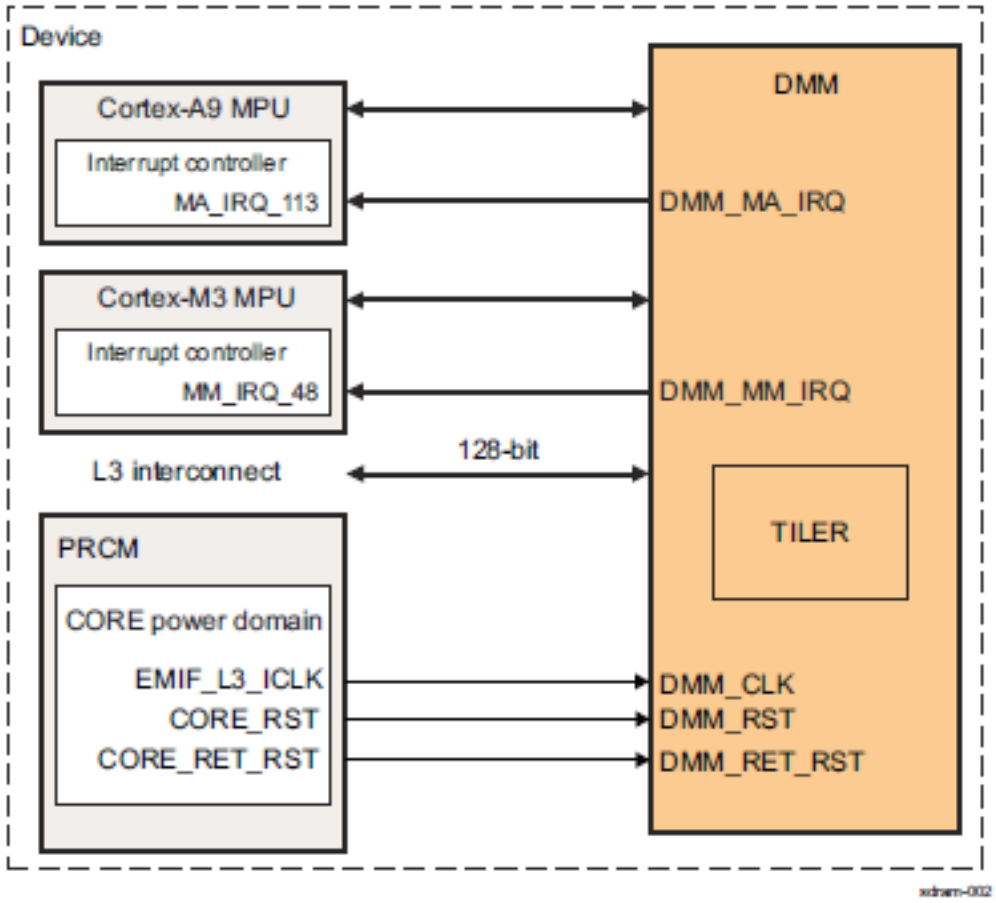
	T == 0 32-bit instructions are fetched (and the PC is incremented by four) and are executed as ARM instructions. **T == 1** 16-bit instructions are fetched (and the PC is incremented by two) and are executed as Thumb instructions. See ARM Architecture Reference Manual at A6-2 . **A3.1.1 Address incrementing and address space overflow** When a processor performs normal sequential execution of instructions, it effectively calculates: `(address_of_current_instruction) + (size_of_executed_instruction)` after each instruction to determine which instruction to execute next. **Note** The size of the executed instruction depends on the current instruction set, and might depend on the instruction executed. If this address calculation overflows the top of the address space, the result is UNPREDICTABLE. In other words, a program must not rely on sequential execution of the instruction at address `0x00000000` after the instruction at address: • `0xFFFFFFFC`, when a 4-byte instruction is executed • `0xFFFFFFFE`, when a 2-byte instruction is executed • `0xFFFFFFFF`, when a single byte instruction is executed. ARMv7 Technical Reference Manual p. A3-2
said incrementer being connected to said internal data bus,	On information and belief, the incrementer is connected to the internal buses in order to increment the instruction address. The internal buses also provide input to an incrementer.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

	T == 0 32-bit instructions are fetched (and the PC is incremented by four) and are executed as ARM instructions. **T == 1** 16-bit instructions are fetched (and the PC is incremented by two) and are executed as Thumb instructions. See ARM Architecture Reference Manual at A6-2 . **A3.1.1 Address incrementing and address space overflow** When a processor performs normal sequential execution of instructions, it effectively calculates: `(address_of_current_instruction) + (size_of_executed_instruction)` after each instruction to determine which instruction to execute next. **Note** The size of the executed instruction depends on the current instruction set, and might depend on the instruction executed. If this address calculation overflows the top of the address space, the result is UNPREDICTABLE. In other words, a program must not rely on sequential execution of the instruction at address `0x00000000` after the instruction at address: • `0xFFFFFFFC`, when a 4-byte instruction is executed • `0xFFFFFFFE`, when a 2-byte instruction is executed • `0xFFFFFFFF`, when a single byte instruction is executed. ARMv7 Technical Reference Manual p. A3-2
said direct memory access central processing unit providing inputs to said memory controller,	On information and belief, the other central processing units of the Accused Microprocessors provide inputs to the memory controller.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

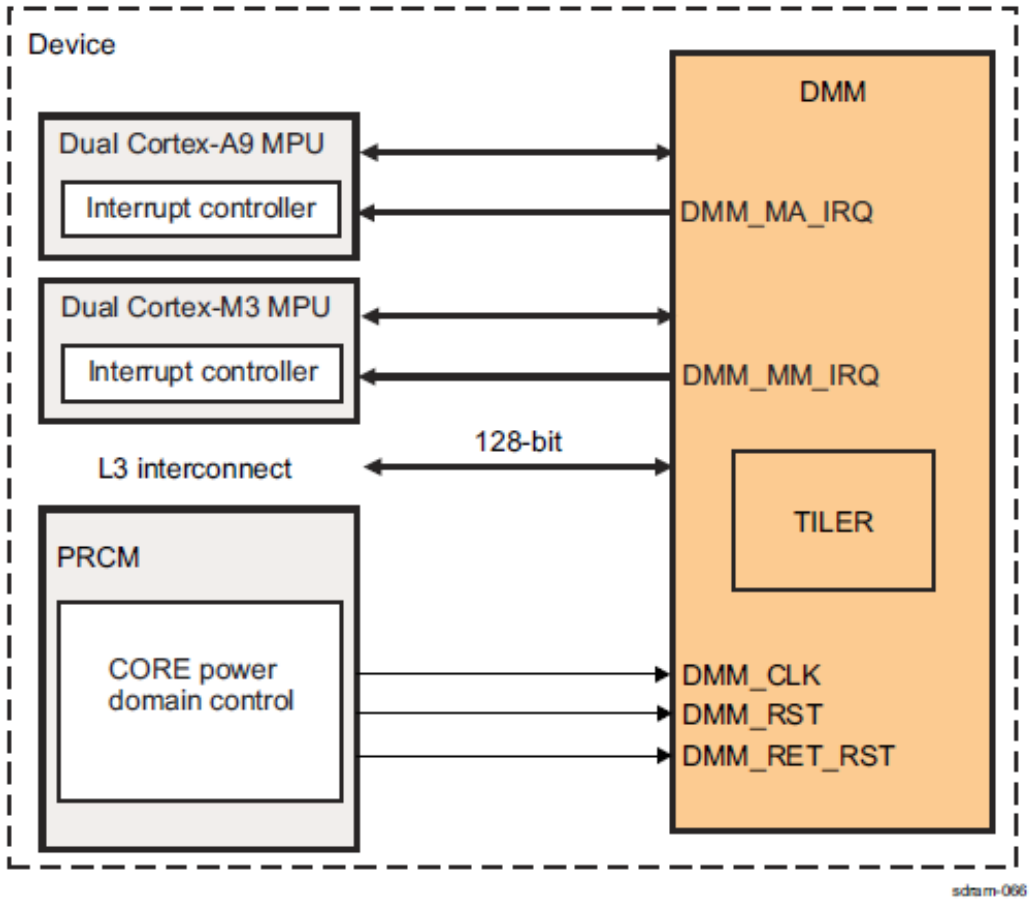
Figure 15-3. DMM Integration



OMAP4430 Technical Reference Manual, p. 3213.

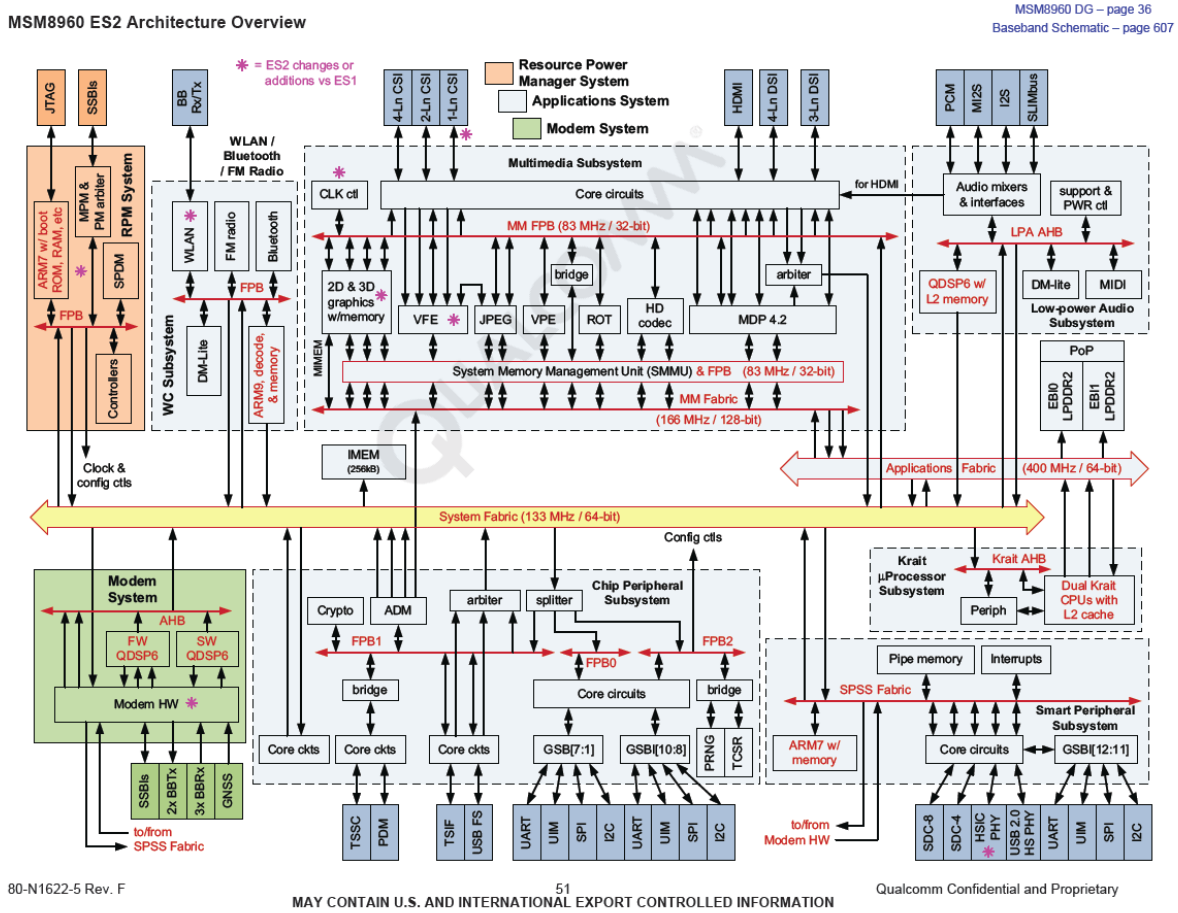
EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Figure 16-2. DMM and Tiler Overview



OMAP4470 Technical Reference Manual, p. 3308.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG



MSM8960 Chipset Schematics and Design Guidelines, p. 50.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

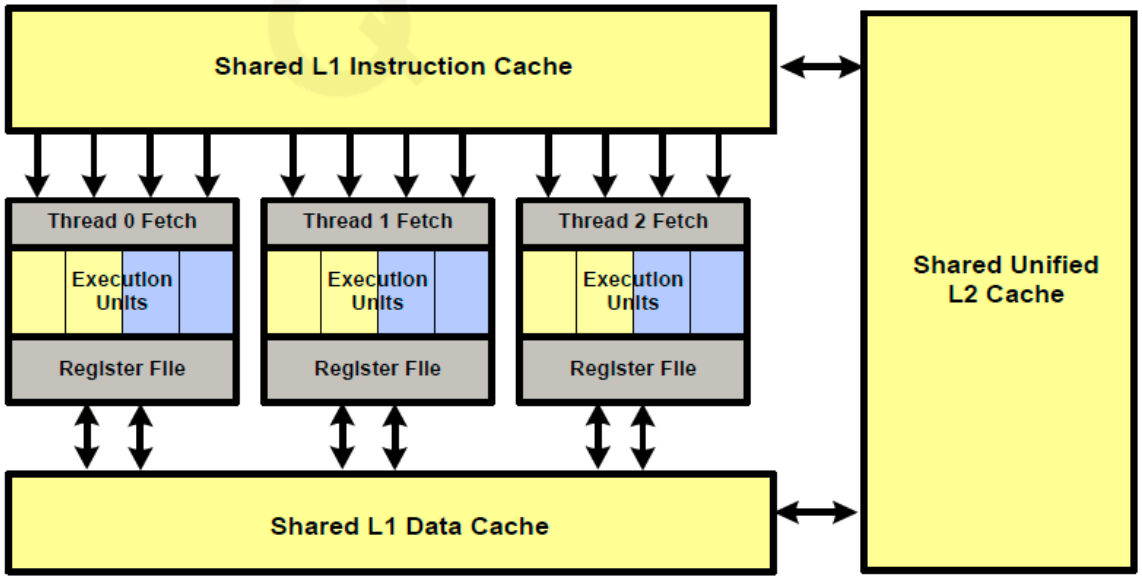
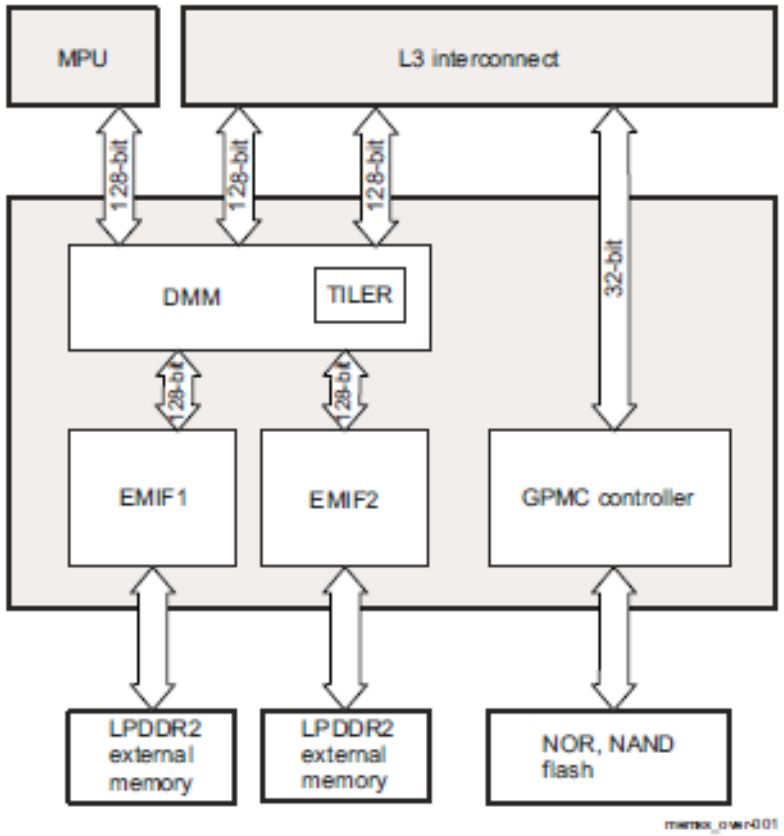
	 Hexagon Programming: Processor Overview, p.10.
said memory controller having an address/data bus and a plurality of control lines for connection to a random access memory.	On information and belief the memory controller of each Accused Microprocessor has an address/data bus and a plurality of control lines for connection to LPDDR2 random access memory.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Figure 15-1 shows a functional diagram of all memory subsystems in the device.

Figure 15-1. Memory Subsystem Functional Diagram

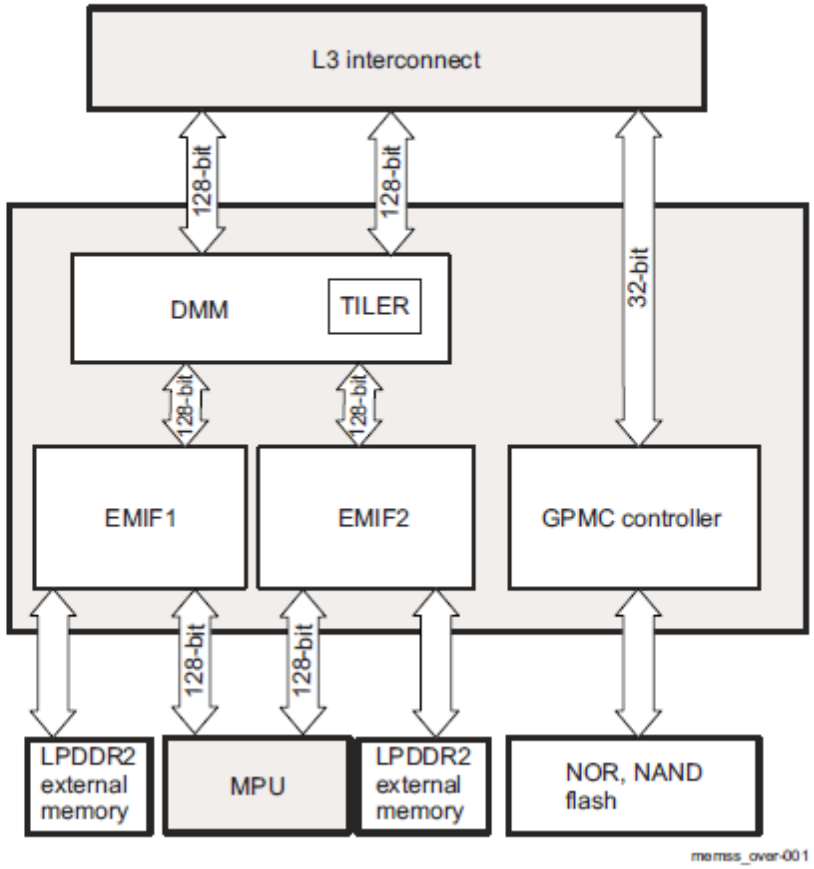


OMAP4430 Technical Reference Manual, p. 3207.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

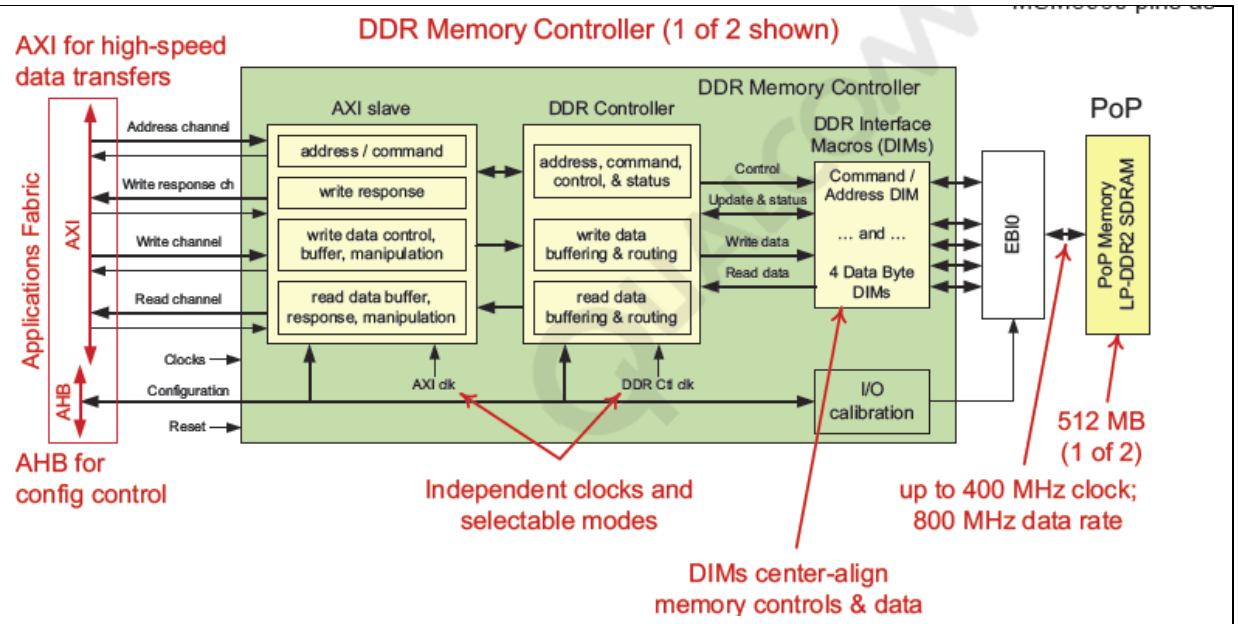
Figure 16-1 shows a functional diagram of all memory subsystems in the device.

Figure 16-1. Memory Subsystem Functional Diagram



OMAP4470 Technical Reference Manual, p. 3302.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

	 MSM8960 Chipset Schematics and Design Guidelines, p. 44.

Claim 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	On information and belief, in each Accused Microprocessor, the PLL clocking the CPU includes a Voltage Controlled Oscillator (VCO) or a Current Controlled Oscillator (ICO), which is a **ring 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. *See* http://en.wikipedia.org/wiki/Ring_oscillator.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

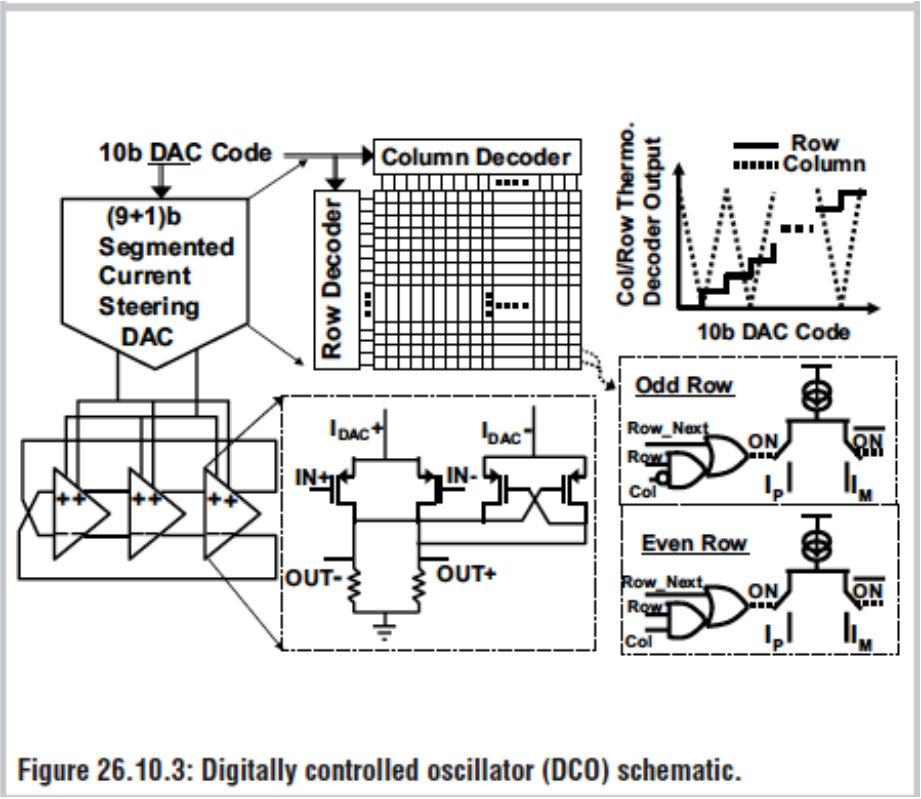
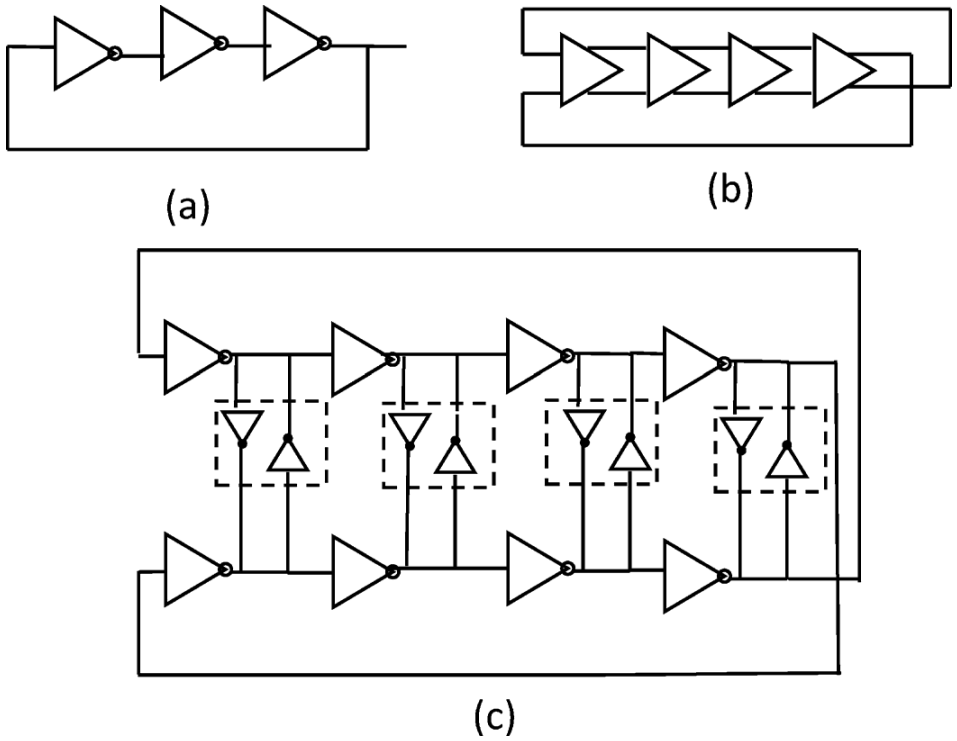
integrated circuit.	For example, generally in TI microprocessors, PLLs have voltage-controlled oscillators, which are circuits capable of maintaining an alternating output. TI has published papers regarding its DPLLs confirming the use of ring oscillators.  Figure 26.10.3: Digitally controlled oscillator (DCO) schematic. TPL853_02993475-84, Fig. 26.10.3 (ISSCC 2004 / Session 26 / *Optical and Fast IO 26.10*)

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

(CX-0170).



TPL853_02993485-96, Fig. 3 (CX-0171). "Figs. 17 and 18 show the die micrographs of two Digital PLLs implemented in 65 nm and 45 nm CMOS. . . . Both use a simple three stage ring oscillator shown in Fig. 3(a)." TPL853_02993494.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

	The TI OMAP4 family of processors is also implemented in 45 nm CMOS technology: "The OMAP4470 high-performance multimedia application device is based on enhanced OMAP architecture and uses 45-nm technology." TPLBN043746 (OMAP4470 Technical Ref Manual). *See also* TPLBN037535 (OMAP4430 Tech Ref Manual). Furthermore, on information and belief, each of the Accused Microprocessors at issue contains a DPLL that outputs a clock signal for the MPU. Within each of these DPLLs is either a single-ended or differential ring oscillator. This component of the PLL generates an oscillation (*i.e.*, a clock signal). The ring oscillator is able to produce an oscillation due to the presence of an odd number of inversions arranged in a loop. *See also HTC Corporation v. Technology Properties Limited*, 08-cv-00882-PSG, Trial Tr. 341:13-347:1, 333:20-374:16 (Haroun testimony). On information and belief, the only input to the ring oscillators in the Accused Microprocessors is a current. In addition, while the bias DAC ("digital to analog converter") in the DPLL may receive a digital word, it does not pass this control signal on to the ring oscillator. Indeed, without the bias component of the PLL, the ring oscillators in the Accused Microprocessors would still output an oscillation. Barring the application of dividers to the ring oscillator's output, the only way to change the frequency of the ring oscillator is to change the input current. In other words, the ring oscillator will always generate a clock signal as long as a current is applied to it.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

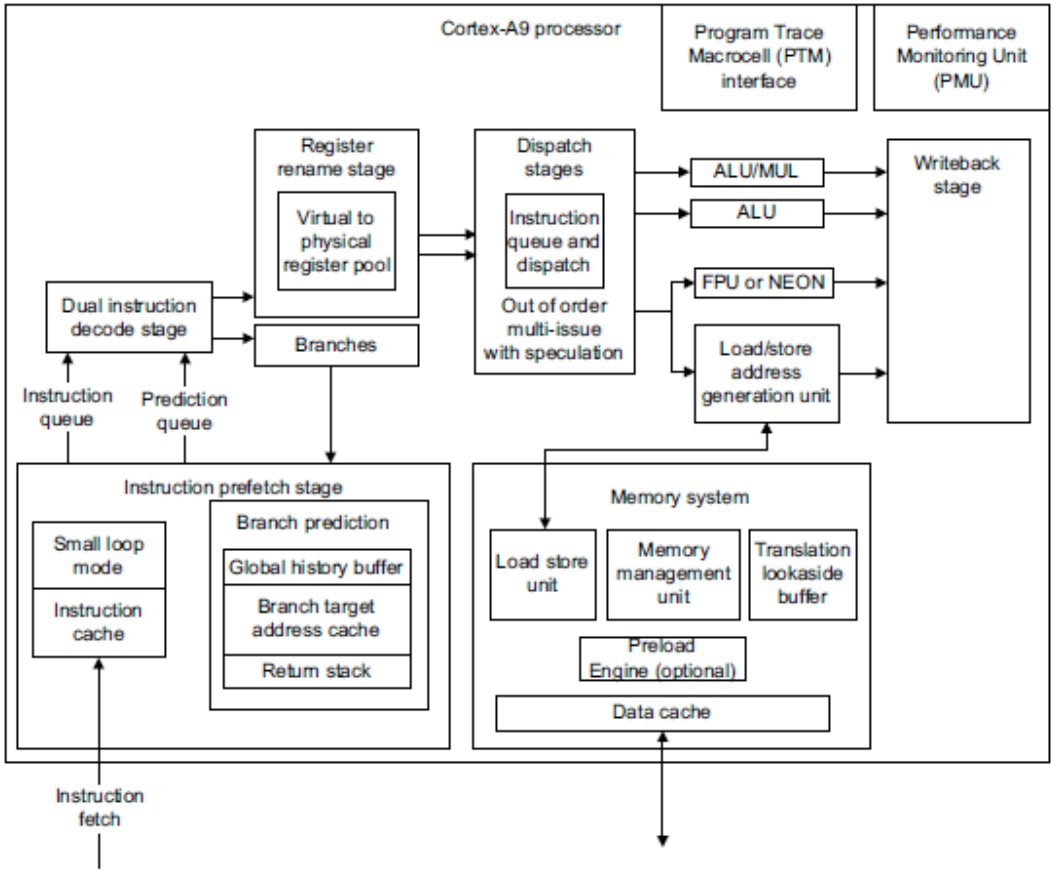
Claim 9	
The microprocessor of claim 1	
in which said first push down stack has a first plurality of stack elements configured as latches,	On information and belief, the first plurality of stack elements is configured as latches (register file or L1 data cache).  **Figure 2-1 Cortex-A9 processor top-level diagram** Cortex-A9 Technical Reference Manual, p.2-2.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Figure 2-1 shows a top-level functional diagram of the Cortex-A15 MPCore processor.

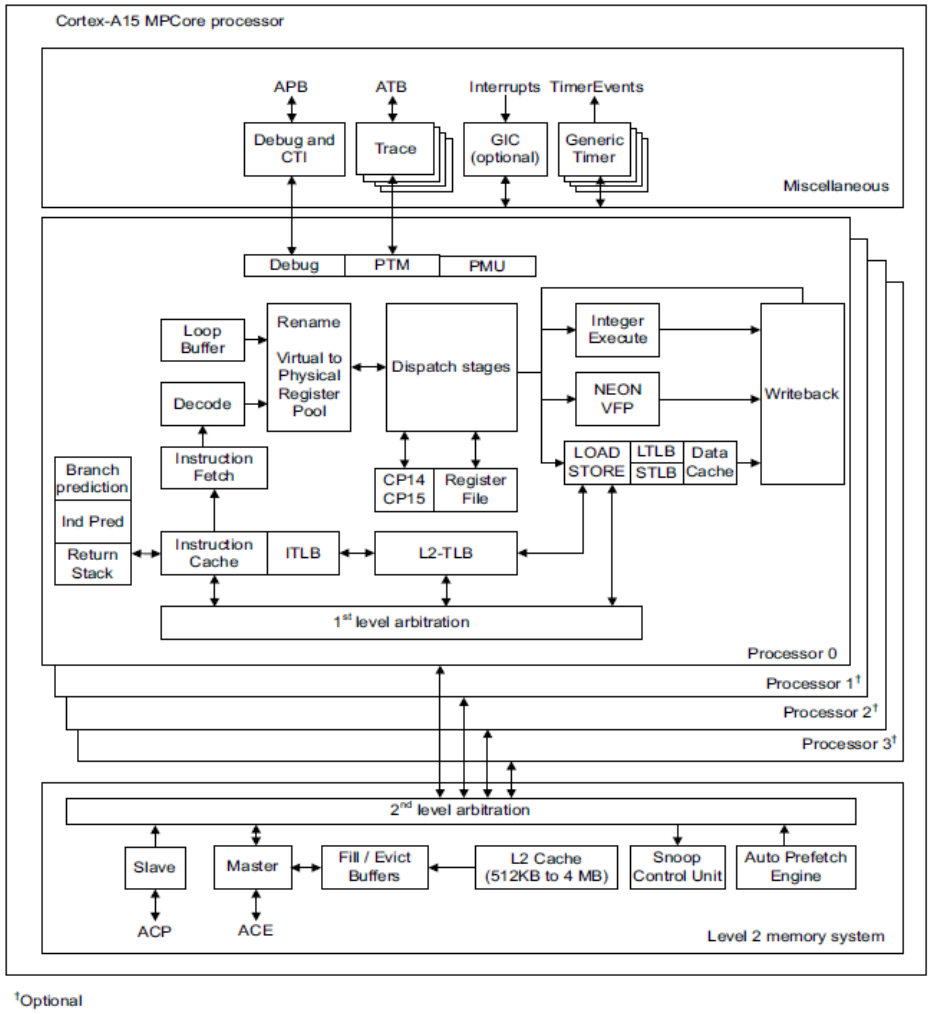


Figure 2-1 Block diagram

Cortex-A15 MPCore Processor Technical Reference Manual p. 2-2.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

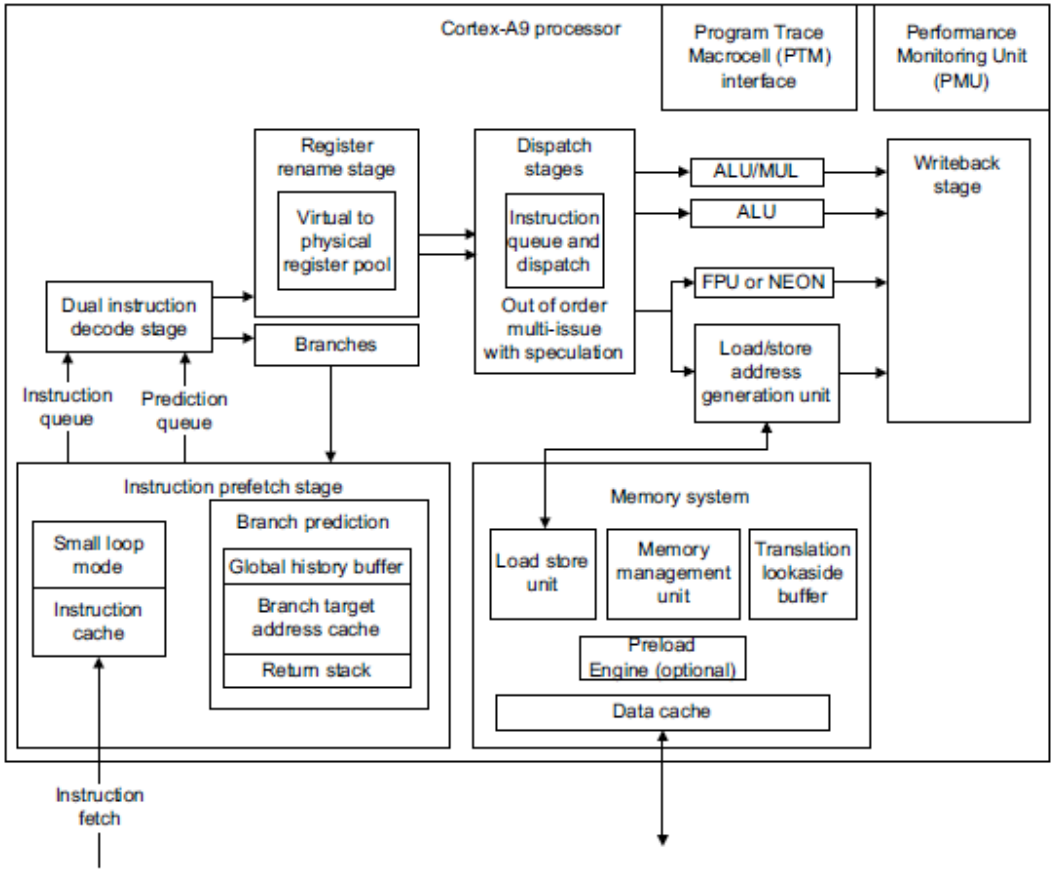
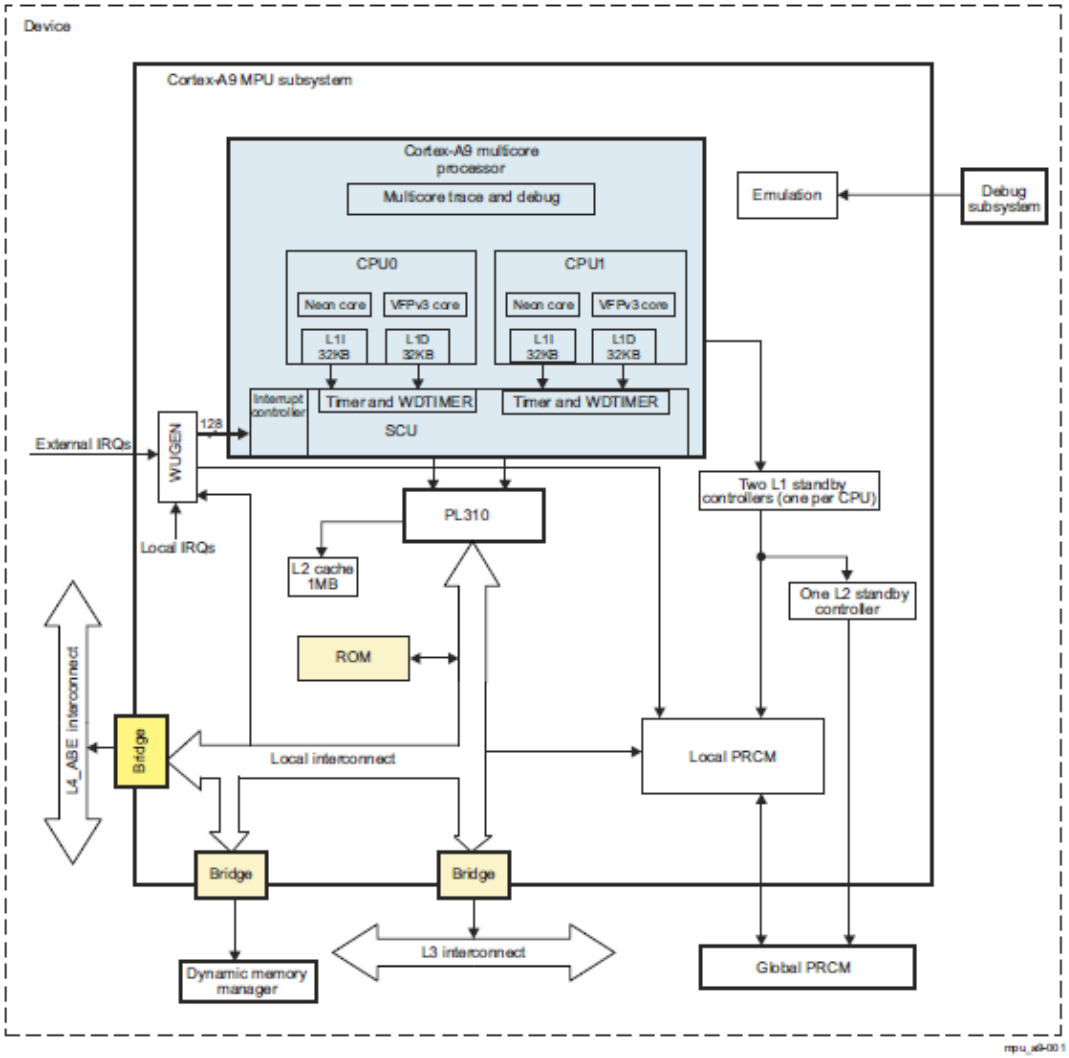
a second plurality of stack elements configured as a random access memory,	On information and belief, the second plurality of stack elements is configured as random access memory (L1 cache or L2 cache).  **Figure 2-1 Cortex-A9 processor top-level diagram** Cortex-A9 Technical Reference Manual, p.2-2.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

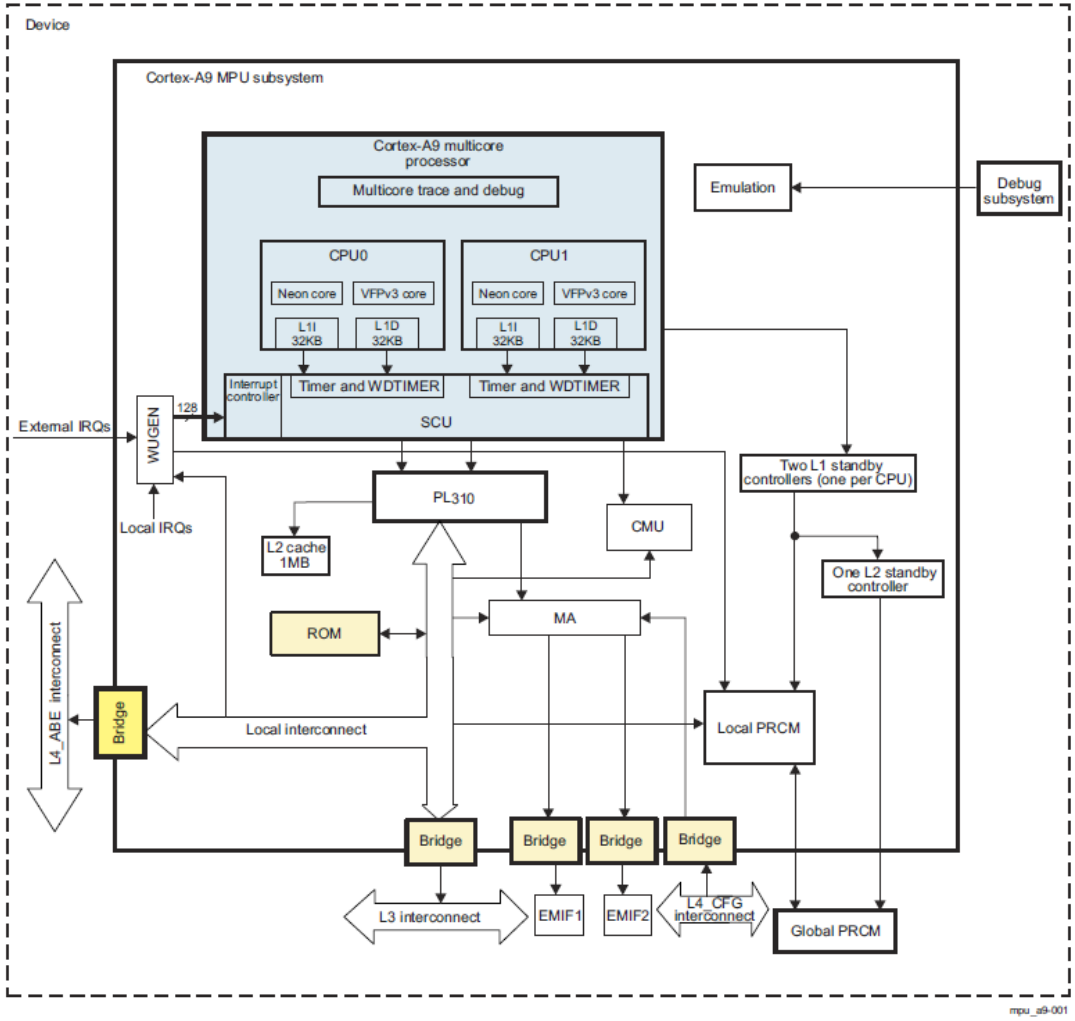
Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4430 Technical Reference Manual, p. 1077.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4470 Technical Reference Manual, p. 1091.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

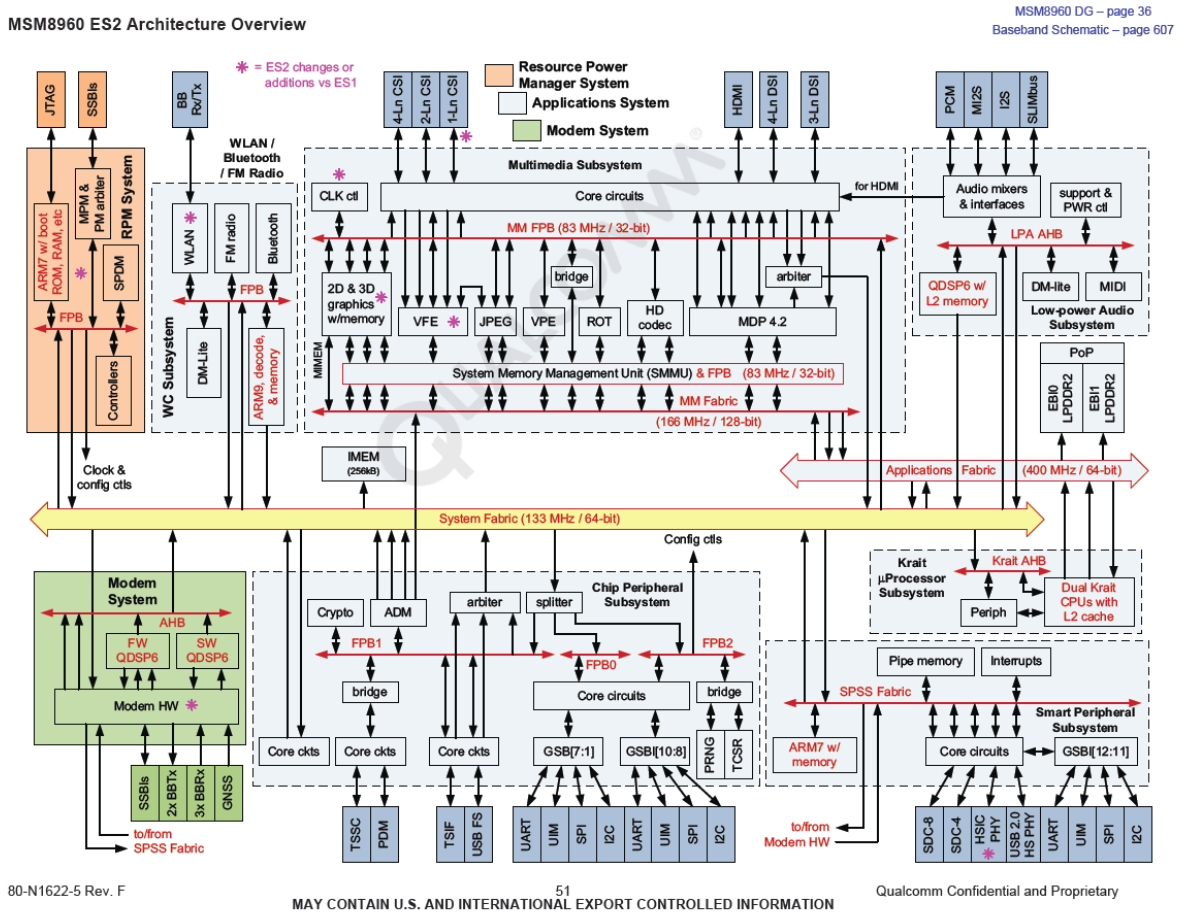
	 MSM8960 Chipset Schematics and Design Guidelines, p. 50.
said first and second plurality of stack elements and said central processing unit being provided in a single integrated circuit,	On information and belief, both the first and the second plurality of stack elements are the central processing unit are provided on the single integrated circuit in the Accused Microprocessors.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

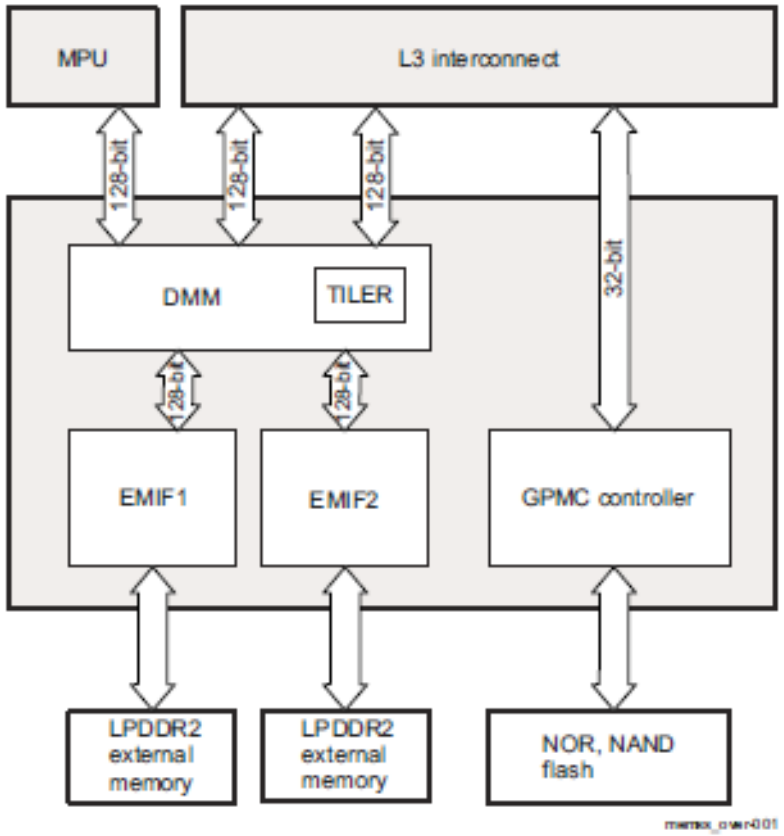
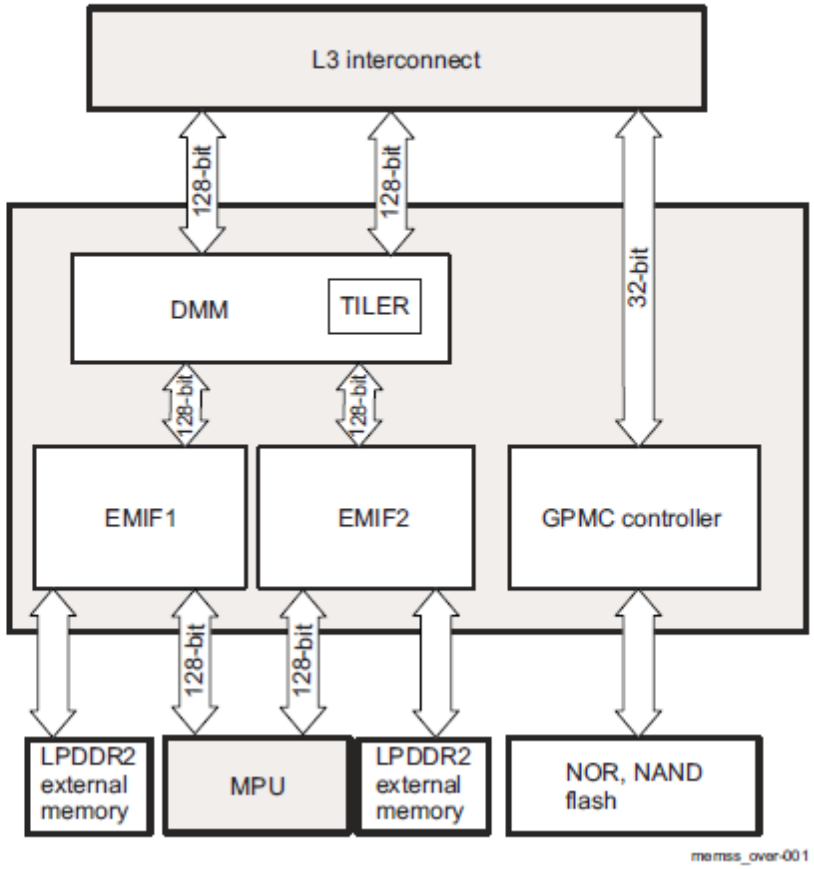
and a third plurality of stack elements configured as a random access memory external to said single integrated circuit.	On information and belief, a third plurality of stack elements is configured as external RAM (e.g, SDRAM). Figure 15-1 shows a functional diagram of all memory subsystems in the device. **Figure 15-1. Memory Subsystem Functional Diagram**  OMAP4430 Technical Reference Manual, p. 3207.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

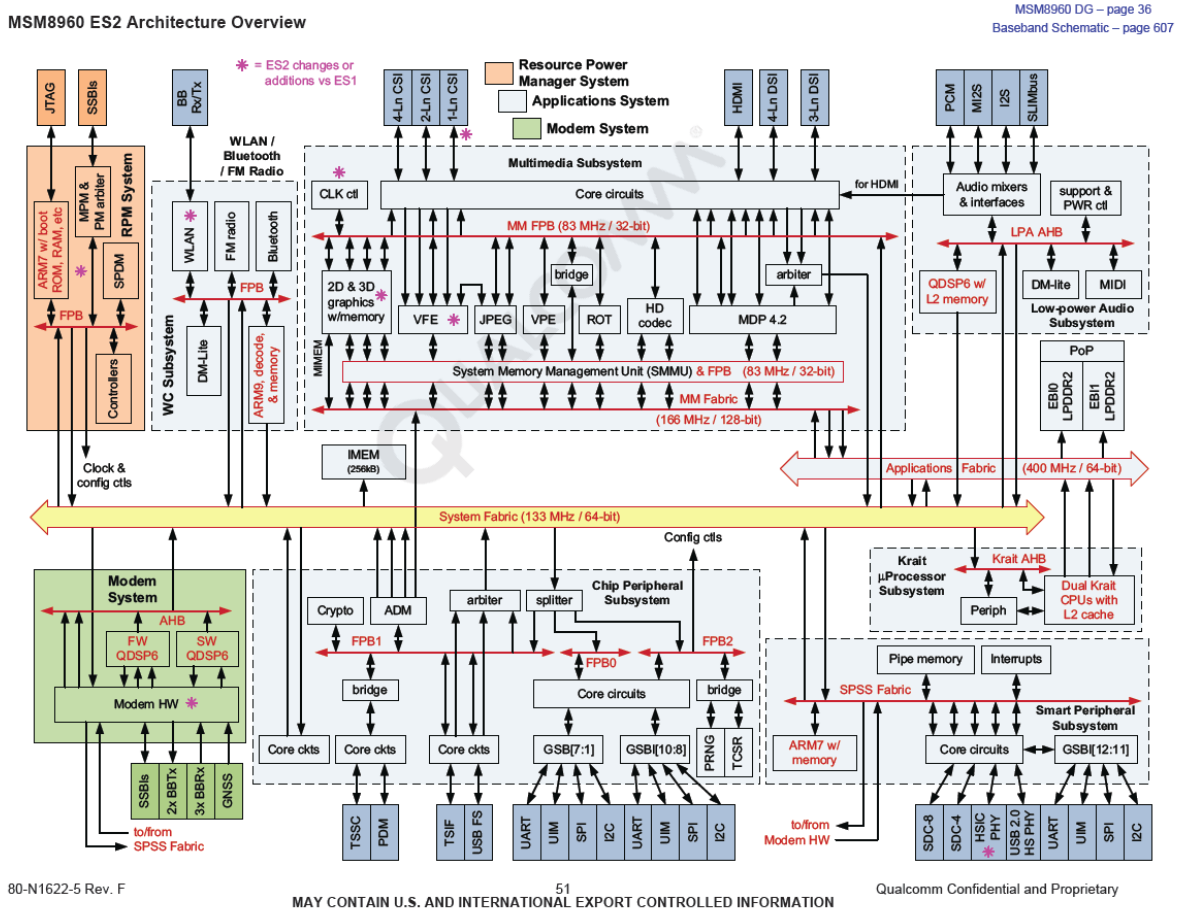
Figure 16-1 shows a functional diagram of all memory subsystems in the device.

Figure 16-1. Memory Subsystem Functional Diagram



OMAP4470 Technical Reference Manual, p. 3302.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG



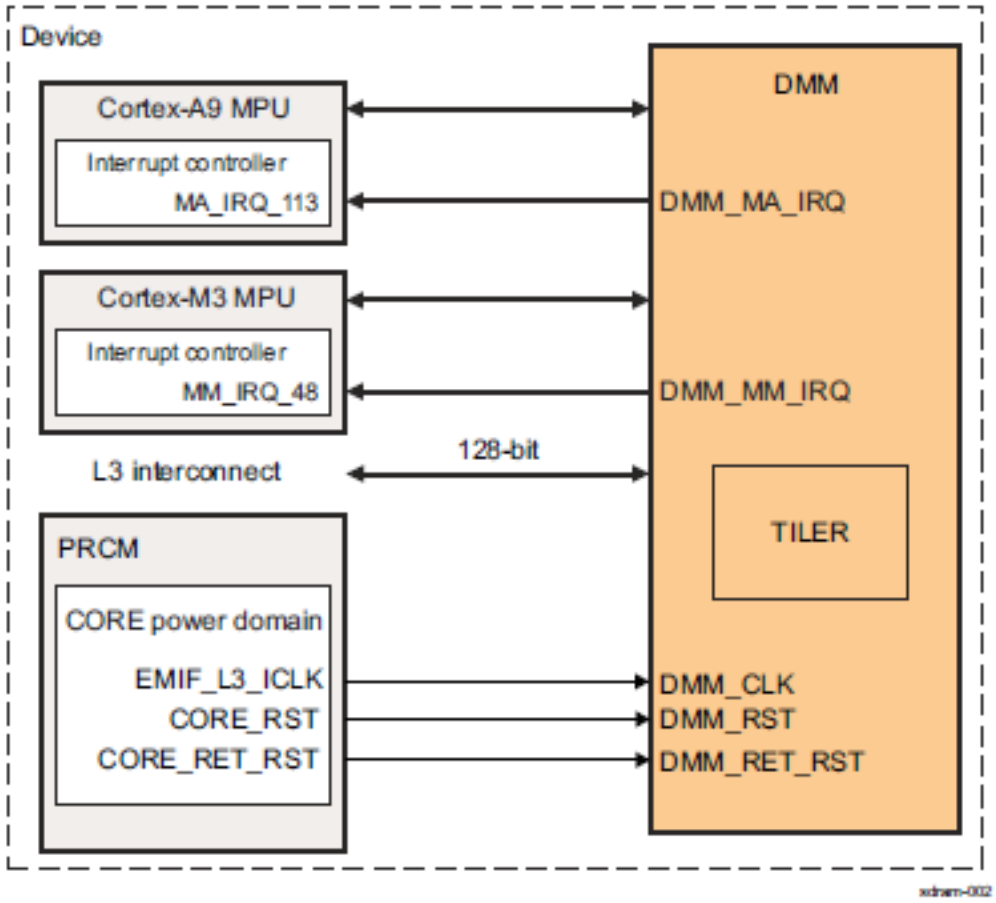
MSM8960 Chipset Schematics and Design Guidelines, p. 50.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Claim 11	
A microprocessor, which comprises	On information and belief, each Accused Product listed in the attached list of Accused Products (Ex. A) contains a microprocessor ("Accused Microprocessors"). *See* Ex. A for listings of microprocessors in the Accused Products, information obtained from http://www.phonescoop.com/; http://pdadb.net/; http://www.gsmarena.com/. Each microprocessor is an electronic circuit that interprets and executes programmed instructions.
a main central processing unit and	As discussed in the attached list of Accused Products, the Accused Microprocessors contain a first ARM core. *See also*, http://en.wikipedia.org/wiki/Qualcomm_Snapdragon; http://en.wikipedia.org/wiki/OMAP.
a separate direct memory access central processing unit in a single integrated circuit comprising said microprocessor,	On information and belief, the Accused Microprocessors contain a second central processing unit separate from the first ARM core. On information and belief, the second central processing unit is able to access memory and fetch and execute instructions separately and independently of the main central processing unit. For example, many of the Accused Microprocessors are multicore processors that contain more than one ARM core.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

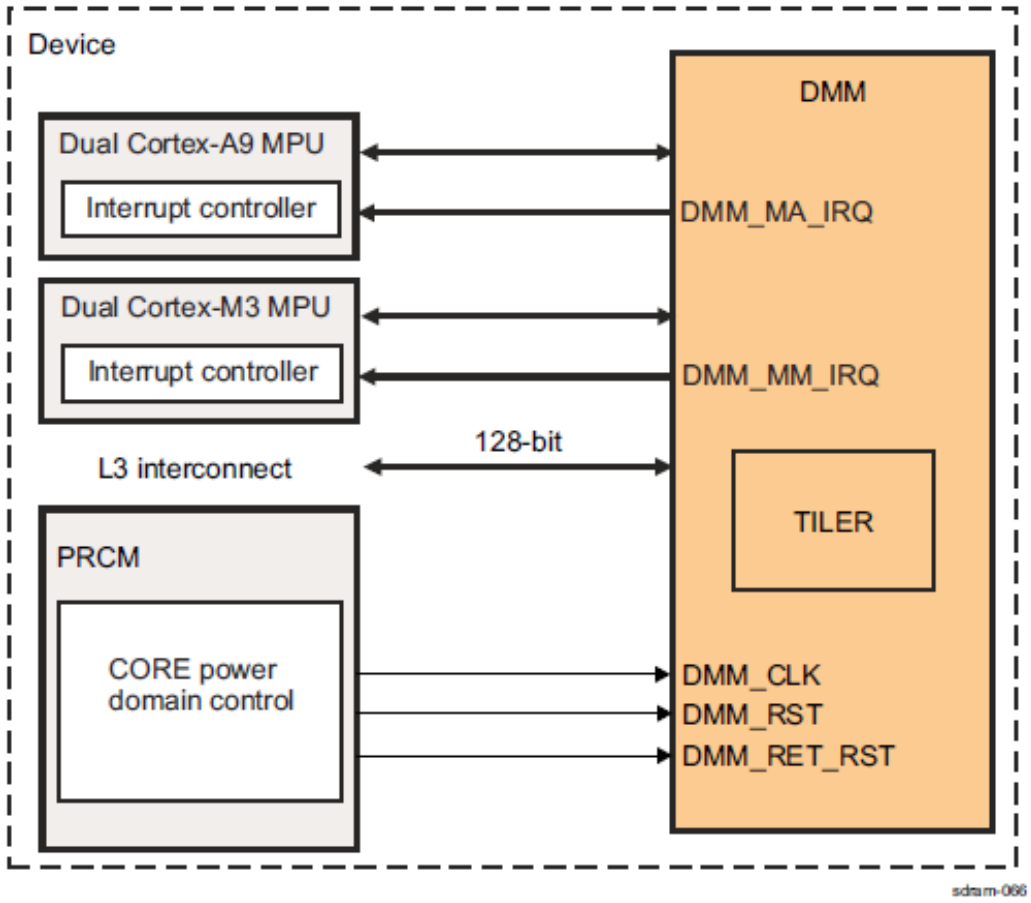
Figure 15-3. DMM Integration



OMAP4430 Technical Reference Manual, p. 3213.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

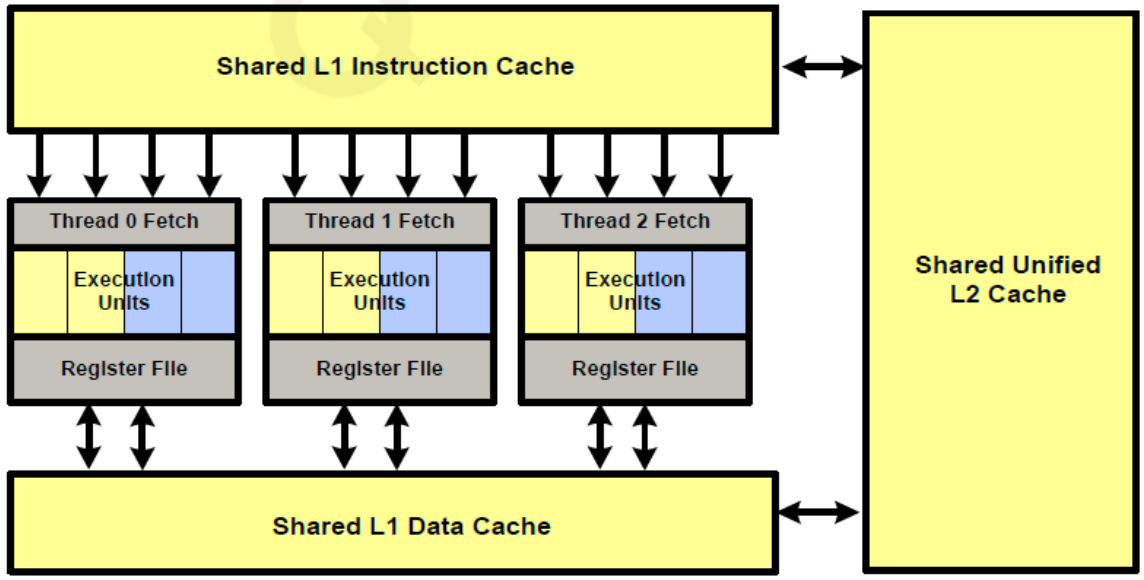
Figure 16-2. DMM and Tiler Overview



OMAP4470 Technical Reference Manual, p. 3308.

In addition, many of the Accused Microprocessors contain one or more DSPs that are able to access memory and fetch and execute instructions separately and independently of the main central processing unit.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG



Hexagon Programming: Processor Overview, p. 10.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

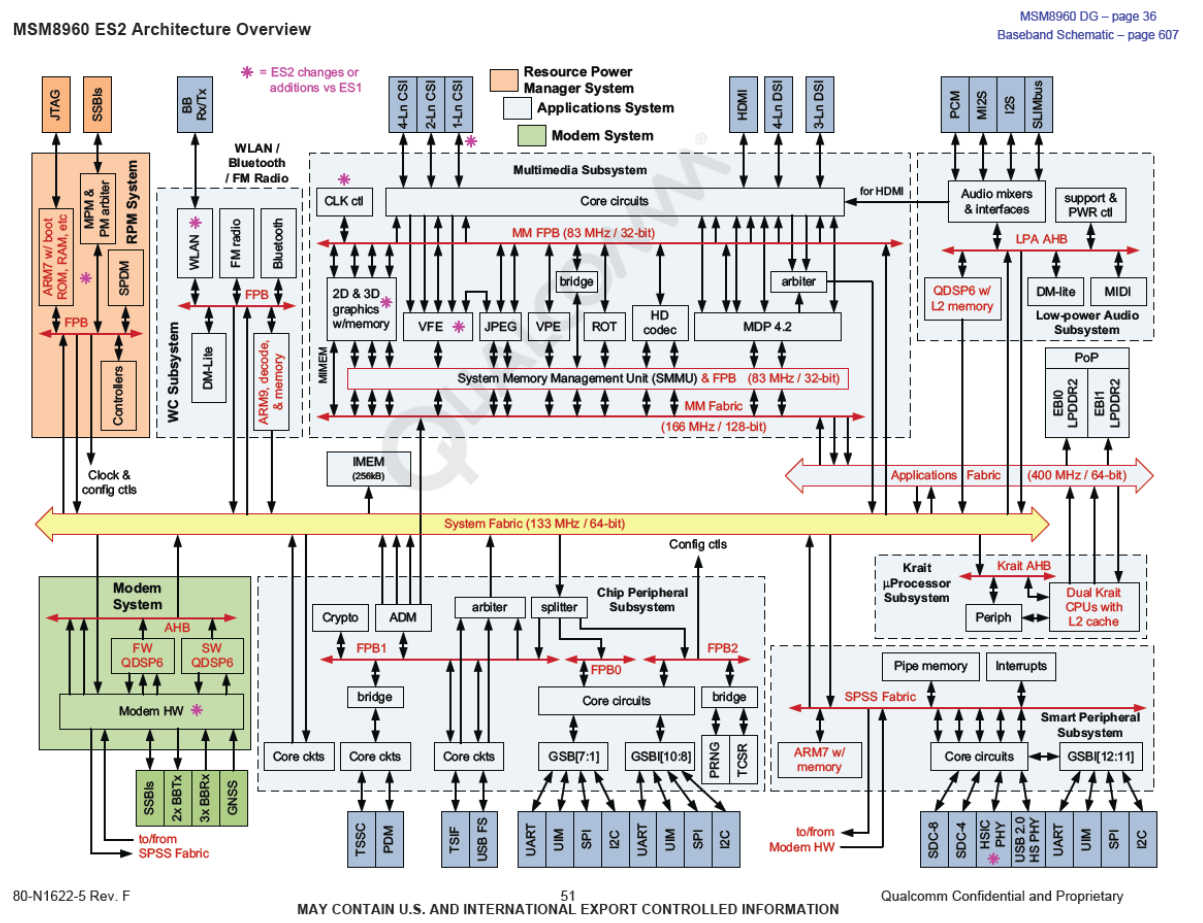
	 MSM8960 Chipset Schematics and Design Guidelines, p. 50. *See also.*, http://en.wikipedia.org/wiki/Qualcomm_Snapdragon; http://en.wikipedia.org/wiki/OMAP.
said main central processing unit having an arithmetic logic unit,	ARM cores, such as those contained within the Accused Microprocessors, contain arithmetic logic units. The ARM instruction provides control over the Arithmetic Logic Unit (ALU) and confirms

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

	that the ALU within the CPU performs arithmetic and logic operations on data. In addition, the ARM architecture provides: • control over both the *Arithmetic Logic Unit* (ALU) and shifter in most data-processing instructions to maximize the use of an ALU and a shifter ARMv7 Architecture Reference Manual [TPLBN051517-TPLBN052654], p. A1-2. **Arithmetic/logic instructions** The following arithmetic/logic instructions share a common instruction format. These perform an arithmetic or logical operation on up to two source operands, and write the result to a destination register. They can also optionally update the condition code flags, based on the result. ARM Architecture Reference Manual, p. A1-7
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,	The ARM architecture defines how an ARM core must operate. ARM Architectures, Processors, and Devices, pp. 1-3 The ARMv7-A architecture is backwards compatible with the ARMv6 architecture. ARM Architectures, Processors, and Devices, p. 1-3, 1-4. The ARM instruction provides control over the Arithmetic Logic Unit (ALU) and confirms that the ALU within the CPU performs arithmetic and logic operations on data. In addition, the ARM architecture provides: • control over both the *Arithmetic Logic Unit* (ALU) and shifter in most data-processing instructions to maximize the use of an ALU and a shifter ARM Architecture Reference Manual, p. A1-2. **Arithmetic/logic instructions** The following arithmetic/logic instructions share a common instruction format. These perform an arithmetic or logical operation on up to two source operands, and write the result to a destination register. They can also optionally update the condition code flags, based on the result. ARM Architecture Reference Manual, p. A1-7. The Arithmetic Logic Unit (ALU) derives its two inputs from the 'Top item' (Rn) and the 'Next item' (Rm) of the 'Push Down Stack' from the 'General Purpose Registers' (the holding place for items placed there by stack operations) and directs its output (Rd) back to the 'General

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Purpose Registers'.

Assembler syntax

ADC{S}<c><q> {<Rd>,} <Rn>, <Rm>, <type> <Rs>

where:

S — If S is present, the instruction updates the flags. Otherwise, the flags are not updated.

<c><q> — See *Standard assembler syntax fields* on page A8-7.

<Rd> — The destination register.

<Rn> — The first operand register.

<Rm> — The register that is shifted and used as the second operand.

<type> — The type of shift to apply to the value read from <Rm>. It must be one of:

- ASR — Arithmetic shift right, encoded as type = 0b10
- LSL — Logical shift left, encoded as type = 0b00
- LSR — Logical shift right, encoded as type = 0b01
- ROR — Rotate right, encoded as type = 0b11.

<Rs> — The register whose bottom byte contains the amount to shift by.

The pre-UAL syntax ADC<c>S is equivalent to ADCS<c>.

ARMv7 Architecture Reference Manual, p. A8-19.

The Register File includes register R13, also known as SP or the Stack pointer, which is a - pointer to the active stack.

SP, the Stack Pointer

Register R13 is used as a pointer to the active stack.

In Thumb code, most instructions cannot access SP. The only instructions that can access SP are those designed to use SP as a stack pointer.

The use of SP for any purpose other than as a stack pointer is deprecated.

ARMv7 Architecture Reference Manual, p. A2-11.

The top two items in the stack are connected to provide inputs into the ALU by using a "POP".

A8.6.122 POP

Pop Multiple Registers loads multiple registers from the stack, loading from consecutive memory locations starting at the address in SP, and updates SP to point just above the loaded data.

ARMv7 Architecture Reference Manual, p. A8-246.

The output of the ALU is connected to the top of the stack and may be saved with a "PUSH."

A8.6.123 PUSH

Push Multiple Registers stores multiple registers to the stack, storing to consecutive memory locations ending just below the address in SP, and updates SP to point to the start of the stored data.

ARMv7 Architecture Reference Manual, p. A8-249.

In the alternative, ARM Cortex A9 processor also includes an operand stack for Jazelle DBX instructions.

A2.10.2 Jazelle direct bytecode execution support

From ARMv5TEJ, the architecture requires every system to include an implementation of the Jazelle extension. The Jazelle extension provides architectural support for hardware acceleration of bytecode execution by a *Java Virtual Machine* (JVM).

ARMv7 Architecture Reference Manual, p. A2-73.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Reads

Reads are defined as memory operations that have the semantics of a load.

The memory accesses of the following instructions are reads:

- LDR, LDRB, LDRH, LDRSB, and LDRSH
- LDRT, LDRBT, LDRHT, LDRSBT, and LDRSHT
- LDREX, LDREXB, LDREXD, and LDREXH
- LDM, LDRD, POP, and RFE
- LDC, LDC2, VLDM, VLDR, VLD1, VLD2, VLD3, and VLD4
- the return of status values by STREX, STREXB, STREXD, and STREXH
- in the ARM instruction set only, SWP and SWPB
- in the Thumb instruction set only, TBB and TBH.

Hardware-accelerated opcode execution by the Jazelle extension can cause a number of reads to occur, according to the state of the operand stack and the implementation of the Jazelle hardware acceleration.

Writes

Writes are defined as memory operations that have the semantics of a store.

The memory accesses of the following instructions are Writes:

- STR, STRB, and STRH
- STRT, STRBT, and STRHT
- STREX, STREXB, STREXD, and STREXH
- STM, STRD, PUSH, and SRS
- STC, STC2, VSTM, VSTR, VST1, VST2, VST3, and VST4
- in the ARM instruction set only, SWP and SWPB.

Hardware-accelerated opcode execution by the Jazelle extension can cause a number of writes to occur, according to the state of the operand stack and the implementation of the Jazelle hardware acceleration.

ARMv7 Architecture Reference Manual, p. A3-42.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

	The operand stack contains a top item register and a next item register connected to provide inputs to the ALU and the output of the ALU is connected to the top item register. The first four elements of the stack are held in the register file in registers R0-R3. The stack pointer is held in register R6. In Java State, the processor assigns several ARM registers to functions specific to the Java machine (e.g., R6= stack pointer, R0-R3= top elements of stack, R4=local variable 0). This hardware reuse contributes to the small size of the additional logic (12k gates) required to implement the Java machine, and keeps all of the state required by the Jazelle extension in ARM registers, In addition, it ensures compatibility with existing operating systems, interrupt handlers and exception code. ARM White Paper, Accelerating to Meet the Challenges of Embedded Java™ at 3. Plaintiffs contend that this claim element is literally present as described above. In the event that this claim element is not found to be literally present, Plaintiffs contend that the above identified stacks are equivalent to "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" and any differences are insubstantial. In particular, the stacks perform the same function (*i.e.*, provide input and output to the ALU), in substantially the same way (*i.e.*, by providing a last-in, first-out data structure), and have the same result (*i.e.*, the ALU performs operations on inputs and provides output).
said top item register also being connected to provide inputs to an internal data bus,	The top item in the stack of an ARM core is connected to provide input to an internal data bus by using a "POP" which loads it into a register.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

	A8.6.122 POP Pop Multiple Registers loads multiple registers from the stack, loading from consecutive memory locations starting at the address in SP, and updates SP to point just above the loaded data. ARMv7 Architecture Reference Manual, p. A8-246. The top item of the stack is also located in the data cache within the "memory system." On information and belief, the register file provides inputs to and is bidirectionally connected to internal buses (e.g., L3 Interconnect, AXI/AHB Bus, Applications/System Fabric)..

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

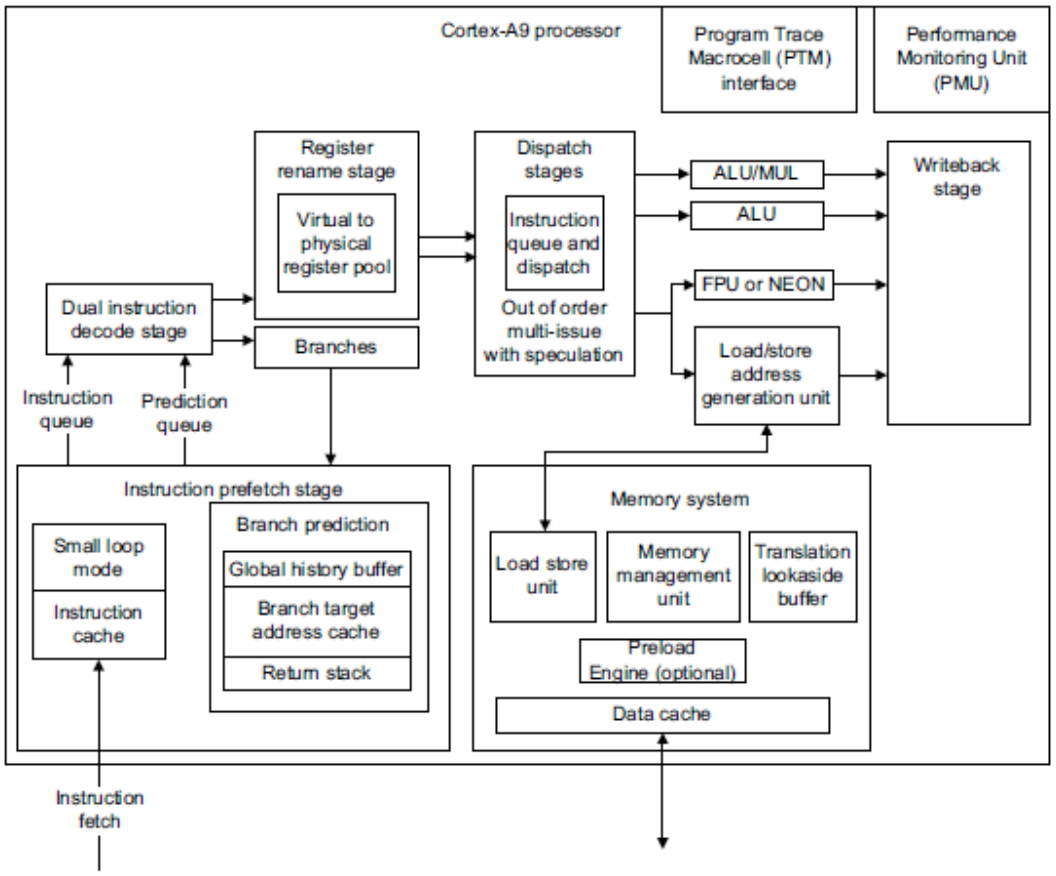
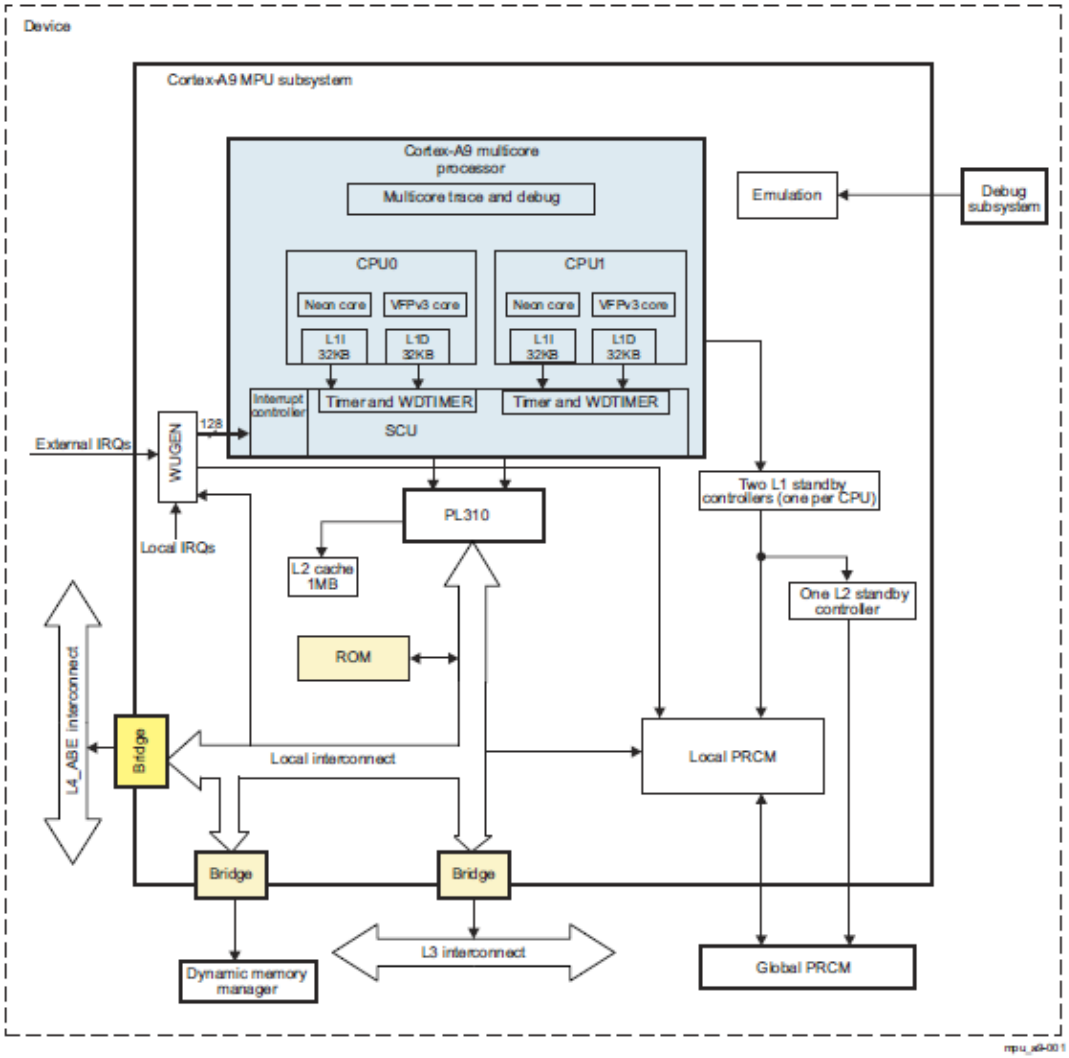


Figure 2-1 Cortex-A9 processor top-level diagram

Cortex-A9 Technical Reference Manual, p.2-2.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

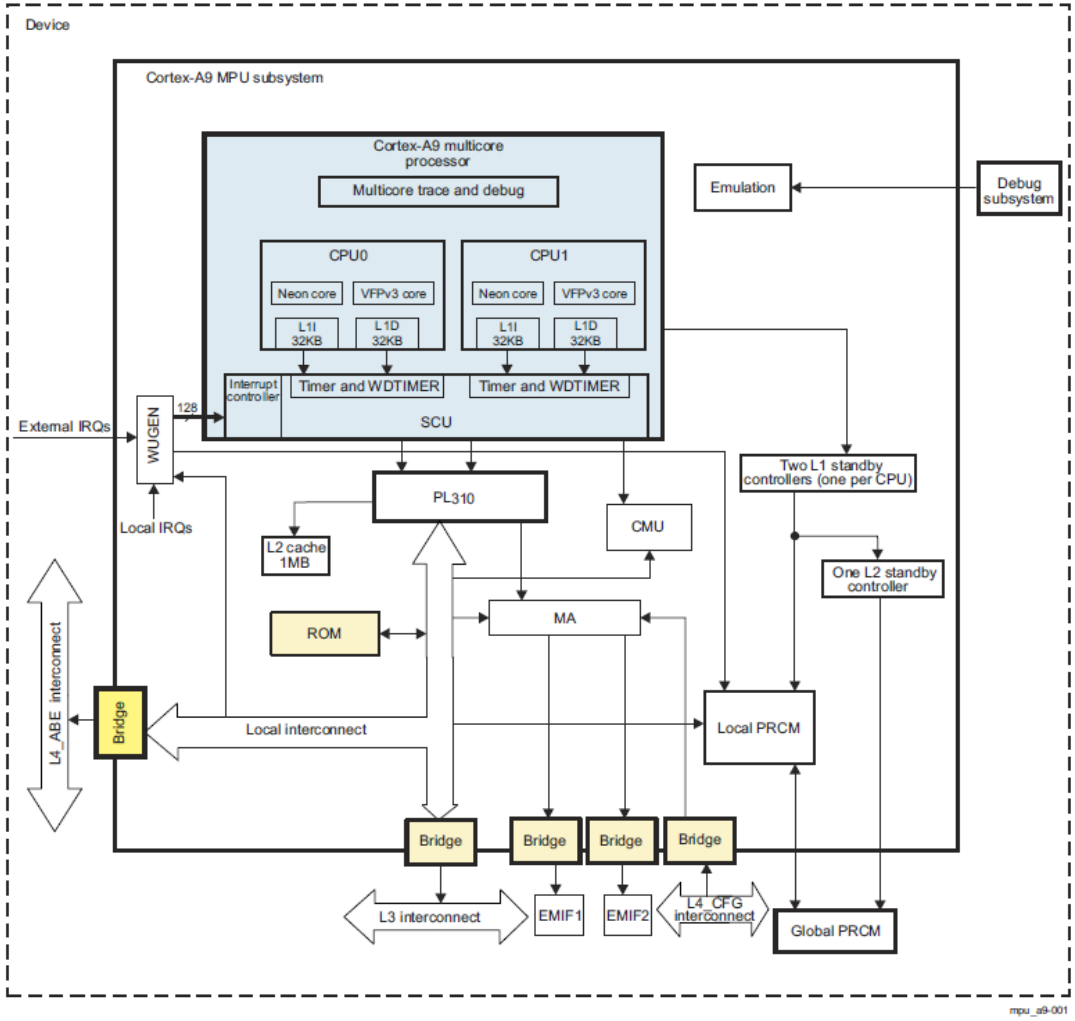
Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4430 Technical Reference Manual, p. 1077.

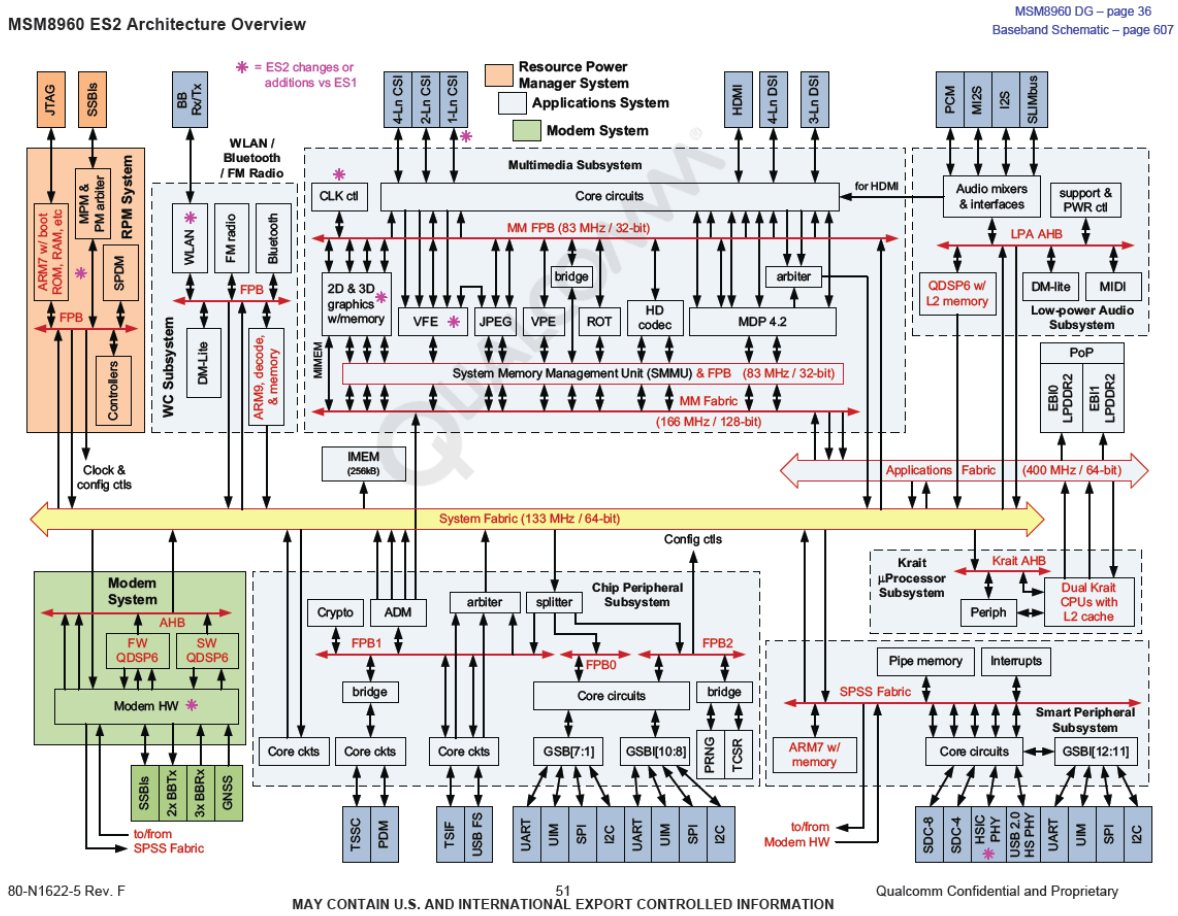
EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4470 Technical Reference Manual, p. 1091.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG



MSM8960 Chipset Schematics and Design Guidelines, p. 50.

In the alternative, when executing Jazelle code, the top item and next item, located in registers R0-R4 are bidirectionally connected to the internal buses.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

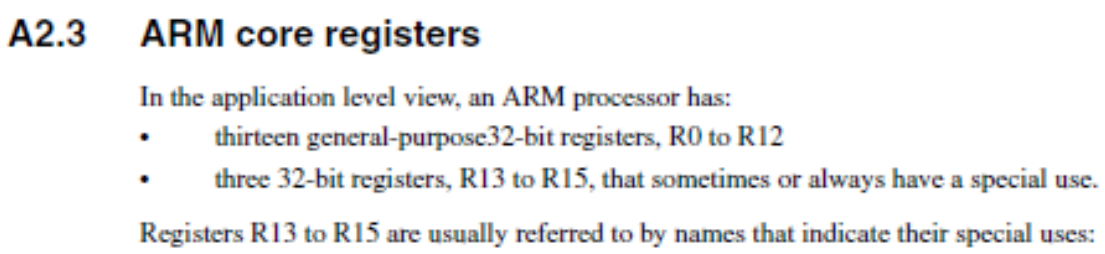
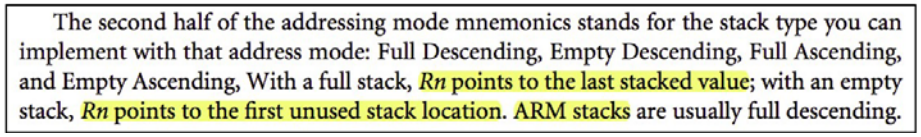
	Plaintiffs contend that this claim element is literally present as described above. In the event that this claim element is not found to be literally present, Plaintiffs contend that the above identified buses, either alone or in combination, are equivalent to "an internal data bus" and any differences are insubstantial. In particular, the buses perform the same function (*i.e.*, provide data connections between the ALU, memory, and registers), in substantially the same way (*i.e.*, by connecting them electrically), and have the same result (*i.e.*, the various connected components transmit data between them).
said internal data bus being bidirectionally connected to a loop counter,	The ARM core contains general purpose registers. For example, in ARMv7 cores there are thirteen general purpose registers R0-R12.  **A2.3 ARM core registers** In the application level view, an ARM processor has: • thirteen general-purpose32-bit registers, R0 to R12 • three 32-bit registers, R13 to R15, that sometimes or always have a special use. Registers R13 to R15 are usually referred to by names that indicate their special uses: ARMv7 Architecture Reference Manual, p. A8-246. Any of the general purpose registers is a loop counter when the register is incremented or decremented and is combined with a conditional branch instruction.  The second half of the addressing mode mnemonics stands for the stack type you can implement with that address mode: Full Descending, Empty Descending, Full Ascending, and Empty Ascending, With a full stack, *Rn* points to the last stacked value; with an empty stack, *Rn* points to the first unused stack location. ARM stacks are usually full descending. ARM System Developer's Guide, A. N. Sloss, Morgan Kaufmann [TPLBN052655-TPLBN053357 p. 585.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

The register file provides inputs to and is bidirectionally connected to internal buses (e.g., L3 Interconnect, AXI/AHB Bus, Applications/System Fabric).

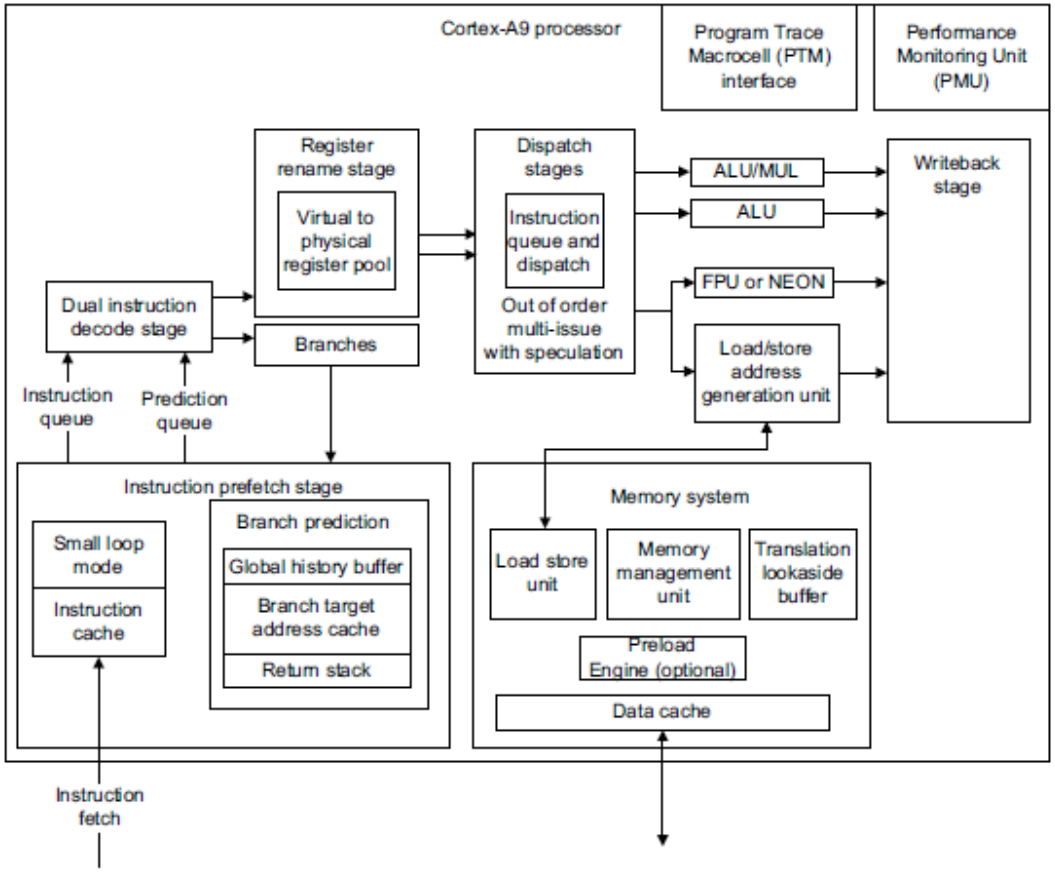
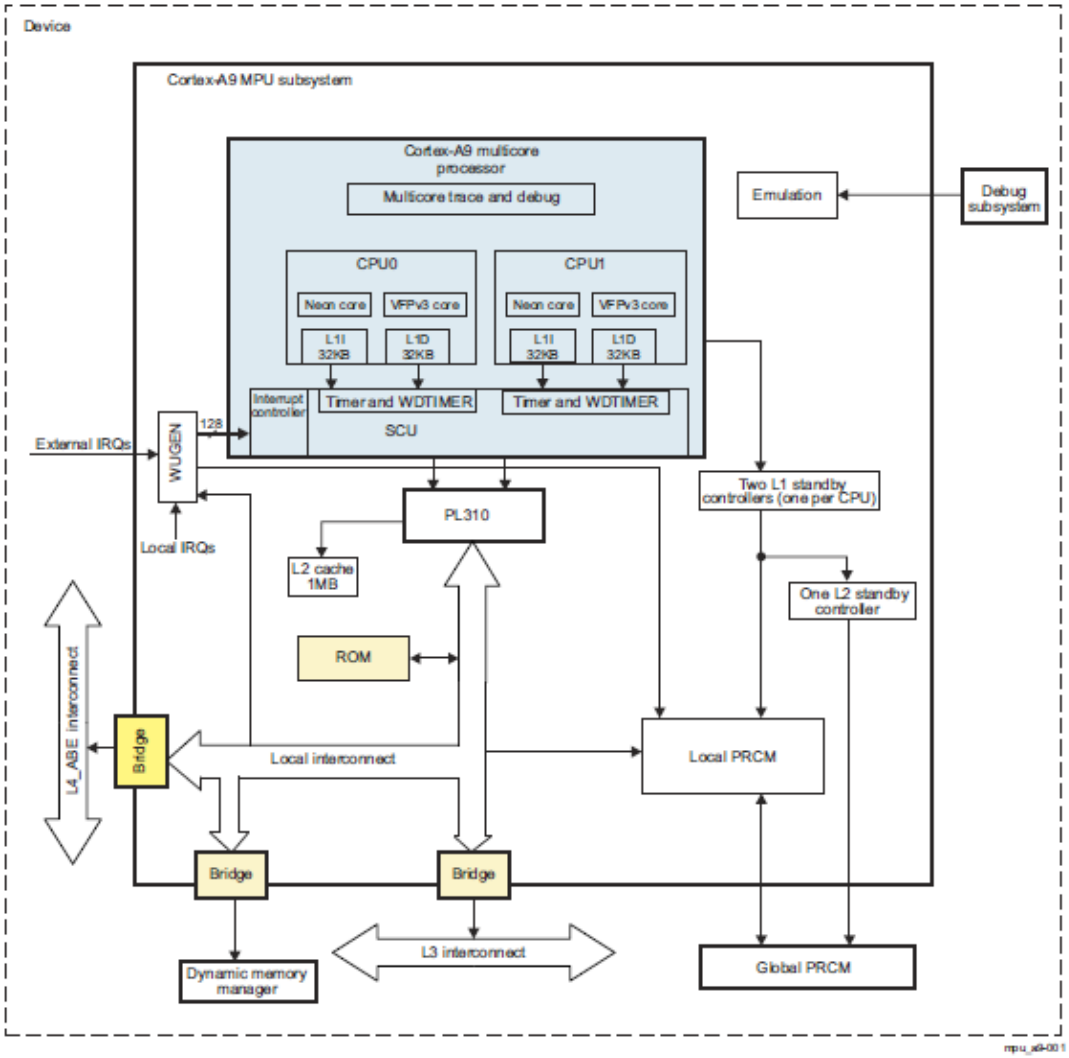


Figure 2-1 Cortex-A9 processor top-level diagram

Cortex-A9 Technical Reference Manual, p.2-2.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

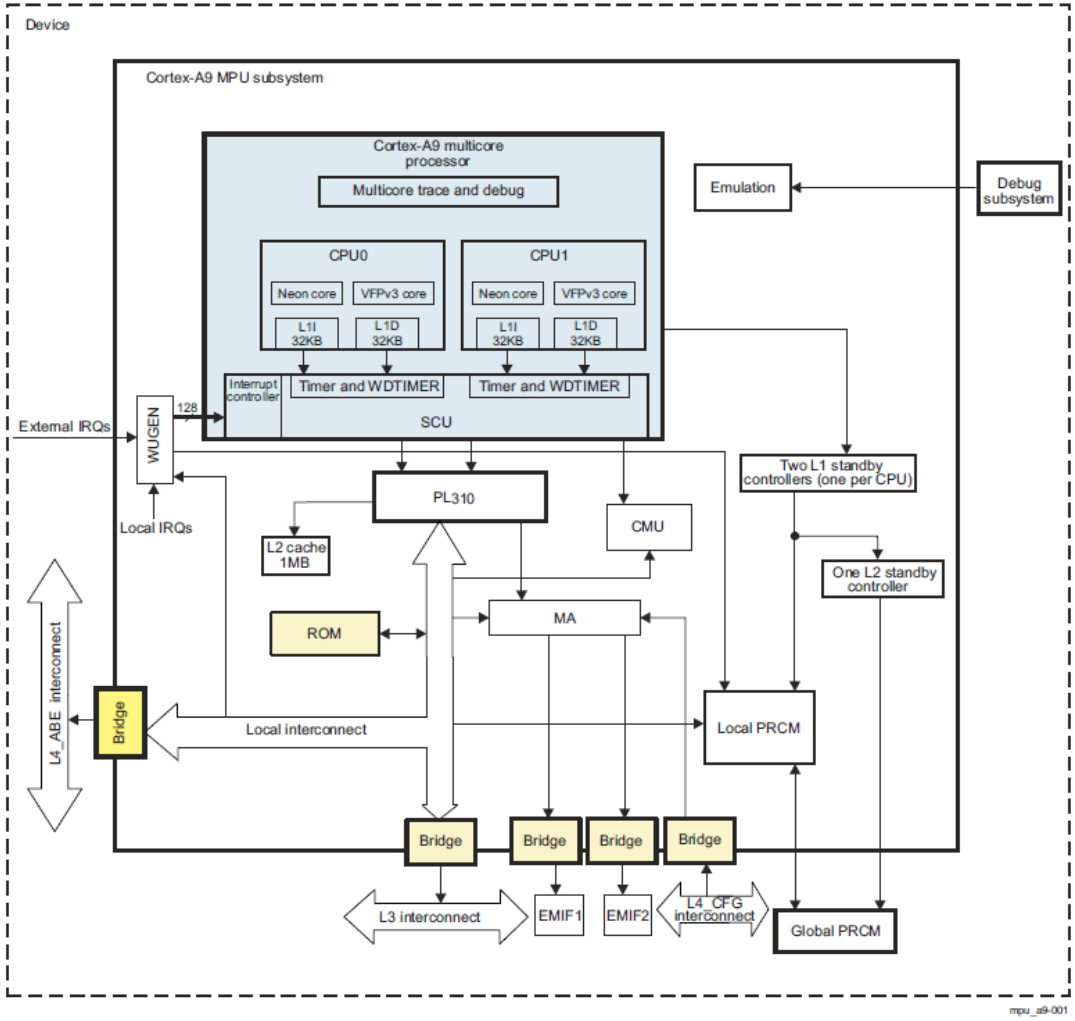
Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4430 Technical Reference Manual, p. 1077.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4470 Technical Reference Manual, p. 1091.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

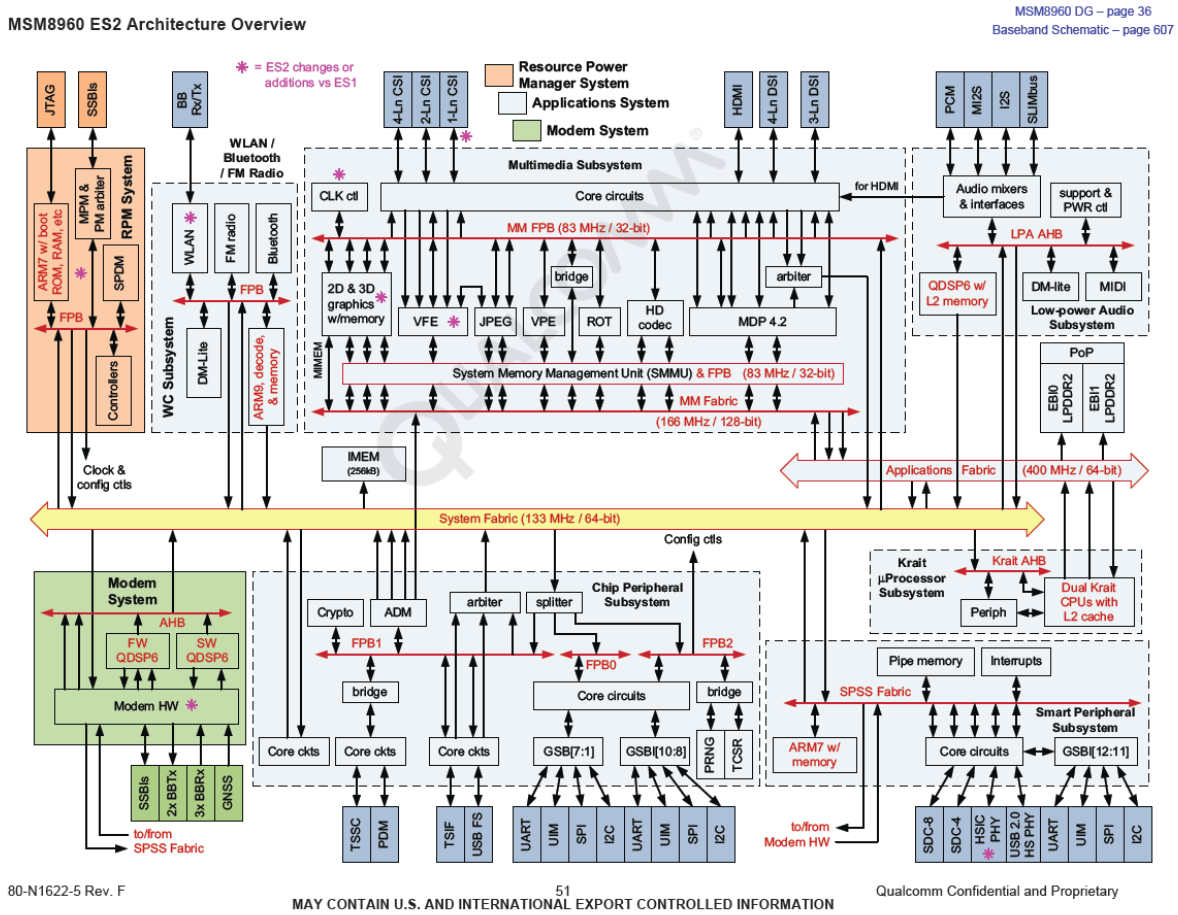
	 MSM8960 Chipset Schematics and Design Guidelines, p. 50. On information and belief, when executing Jazelle instructions within a loop, the processor maintains a loop counter that is bidirectionally connected to the Accused Microprocessor's internal buses.
said loop counter being connected to	Each ARM core contains a general purpose register or other hardware is used as a loop

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

a decrementer,	counter, the loop counter is connected to an ALU or an adder, which serves as a decrementer. For example, in ARMv7 cores there are thirteen general purpose registers R0-R12. **A2.3 ARM core registers** In the application level view, an ARM processor has: • thirteen general-purpose32-bit registers, R0 to R12 • three 32-bit registers, R13 to R15, that sometimes or always have a special use. Registers R13 to R15 are usually referred to by names that indicate their special uses: ARMv7 Architecture Reference Manual, p. A8-246. Any of the general purpose registers is a loop counter when the register is incremented or decremented and is combined with a conditional branch instruction. The second half of the addressing mode mnemonics stands for the stack type you can implement with that address mode: Full Descending, Empty Descending, Full Ascending, and Empty Ascending, With a full stack, *Rn* points to the last stacked value; with an empty stack, *Rn* points to the first unused stack location. ARM stacks are usually full descending. ARM System Developer's Guide, A. N. Sloss, Morgan Kaufmann [TPLBN052655-TPLBN053357 p. 585. The register file provides inputs to and is bidirectionally connected to internal buses (e.g., L3 Interconnect, AXI/AHB Bus, Applications/System Fabric).

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

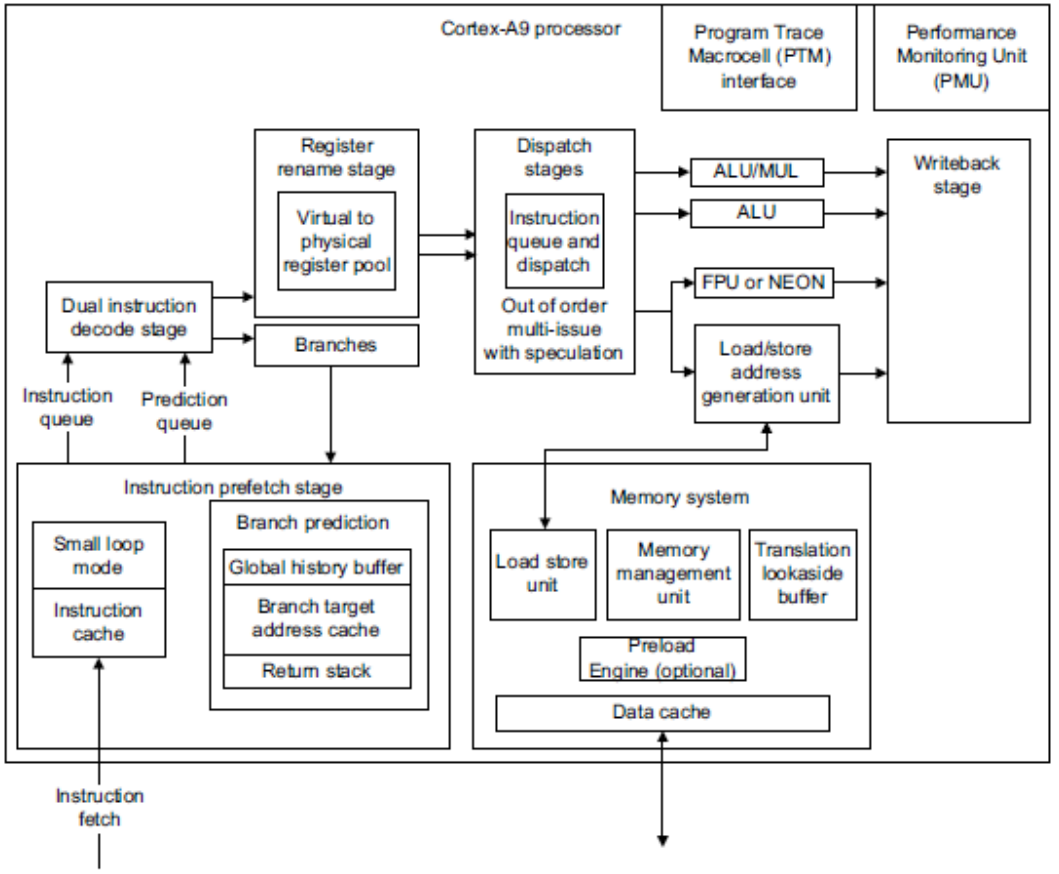
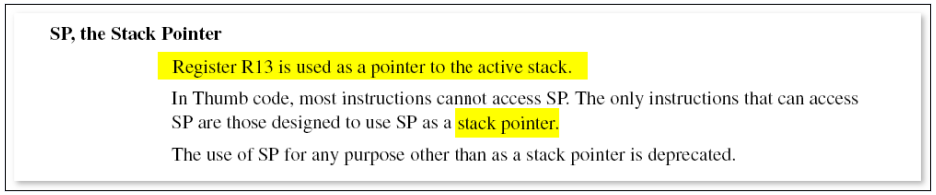
	 **Figure 2-1 Cortex-A9 processor top-level diagram** Cortex-A9 Technical Reference Manual, p.2-2.
said internal data bus being bidirectionally connected to a stack pointer, return stack pointer, mode register and instruction register,	Each ARM core has a stack pointer.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG



SP, the Stack Pointer

Register R13 is used as a pointer to the active stack.

In Thumb code, most instructions cannot access SP. The only instructions that can access SP are those designed to use SP as a stack pointer.

The use of SP for any purpose other than as a stack pointer is deprecated.

ARM Architecture Reference Manual - ARMv7-A and ARMv7-R p. A2-11

On information and belief, the Register File of each Accused Microprocessor is bidirectionally connected to the data bus. The Register File includes Register R13, also known as SP or the stack pointer, which points to the active stack and Register R14, also known as LR, the Link Register, which contains an identical memory address as the top of the return stack.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

SP, the Stack Pointer

Register R13 is used as a pointer to the active stack.

In Thumb code, most instructions cannot access SP. The only instructions that can access SP are those designed to use SP as a stack pointer.

The use of SP for any purpose other than as a stack pointer is deprecated.

Note

Using SP for any purpose other than as a stack pointer is likely to break the requirements of operating systems, debuggers, and other software systems, causing them to malfunction.

LR, the Link Register

Register R14 is used to store the return address from a subroutine. At other times, LR can be used for other purposes.

When a `BL` or `BLX` instruction performs a subroutine call, LR is set to the subroutine return address. To perform a subroutine return, copy LR back to the program counter. This is typically done in one of two ways, after entering the subroutine with a `BL` or `BLX` instruction:

- Return with a `BX LR` instruction.
- On subroutine entry, store LR to the stack with an instruction of the form:
  `PUSH {<registers>,LR}`
  and use a matching instruction to return:
  `POP {<registers>,PC}`

ThumbEE checks and handler calls use LR in a similar way. For details see Chapter A9 *ThumbEE*.

ARMv7 Architecture Reference Manual, p. A2-11.

On information and belief, one or more instruction registers receive instructions from the instruction cache.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

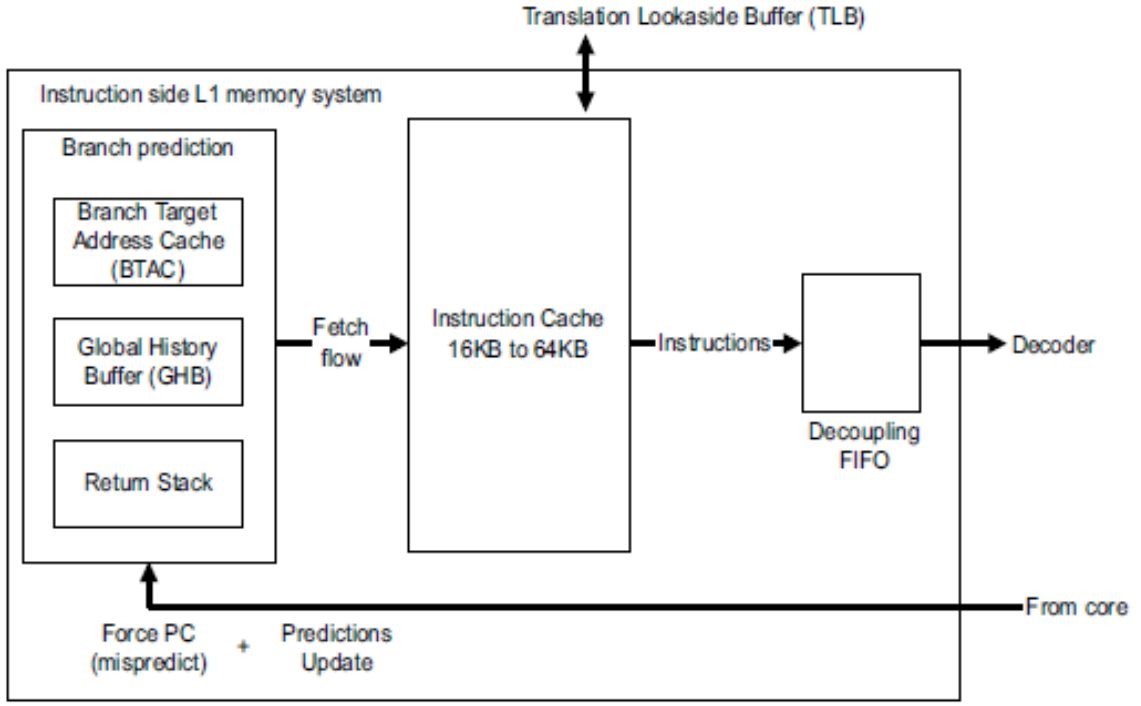


Figure 6-1 Branch prediction and instruction cache

Cortex-A9 Technical Reference Manual p. 6-5.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Figure 2-1 shows a top-level functional diagram of the Cortex-A15 MPCore processor.

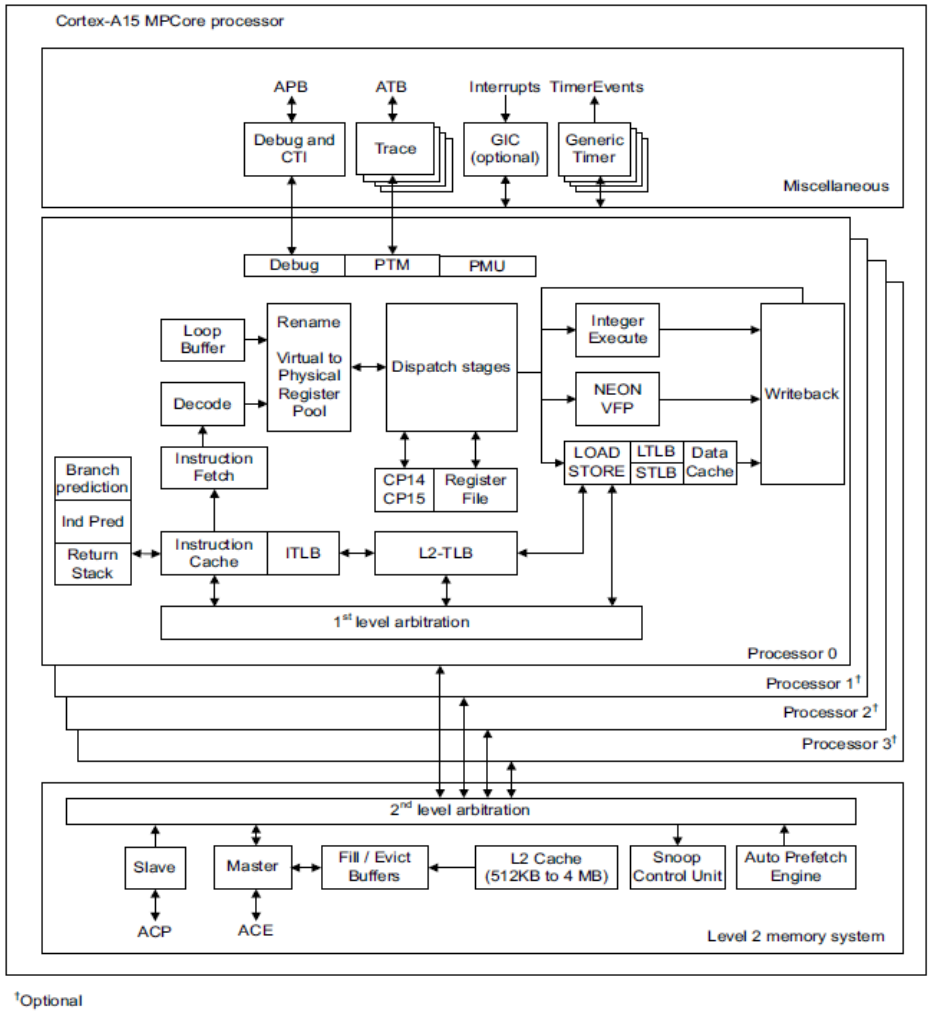


Figure 2-1 Block diagram

Cortex-A15 MPCore Processor Technical Reference Manual p. 2-2.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

	Instruction fetch The instruction fetch unit fetches instructions from the L1 instruction cache and delivers up to three instructions per cycle to the instruction decode unit. It supports dynamic and static branch prediction. The instruction fetch unit includes: • L1 instruction cache that is a 32KB 2-way set-associative cache with 64 bytes cache line and optional parity protection per 16-bits. • 2-level dynamic predictor with BTB for fast target generation. • Return stack. • Static branch predictor. • Indirect predictor. • 32-entry fully-associative L1 instruction TLB. Cortex-A15 MPCore Processor Technical Reference Manual p. 2-3. The ARM core contains a number of mode registers within its system control coprocessors. *See, e.g.*, Cortex-A9 Technical Reference Manual Ch. 4; Cortex-A15 MPCore Processor Technical Reference Manual Ch. 4. On information and belief, the register file provides inputs to and is bidirectionally connected to internal buses (e.g., L3 Interconnect, AXI/AHB Bus, Applications/System Fabric.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

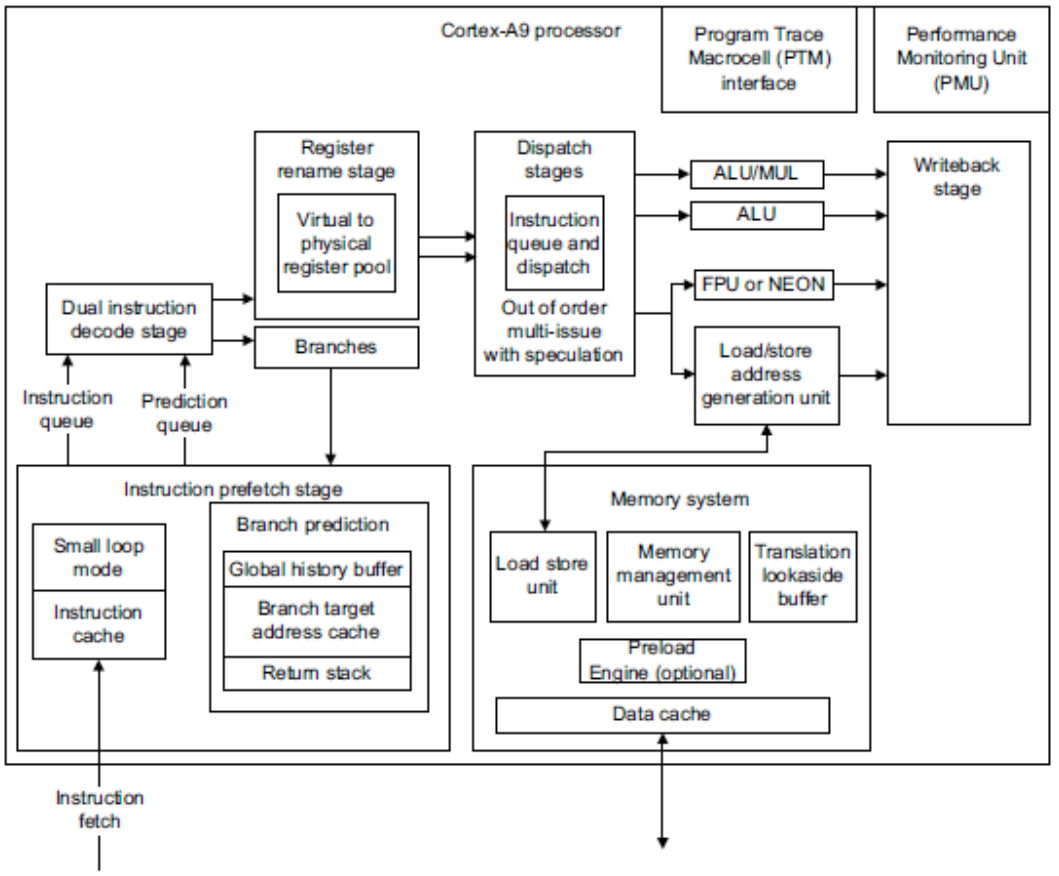
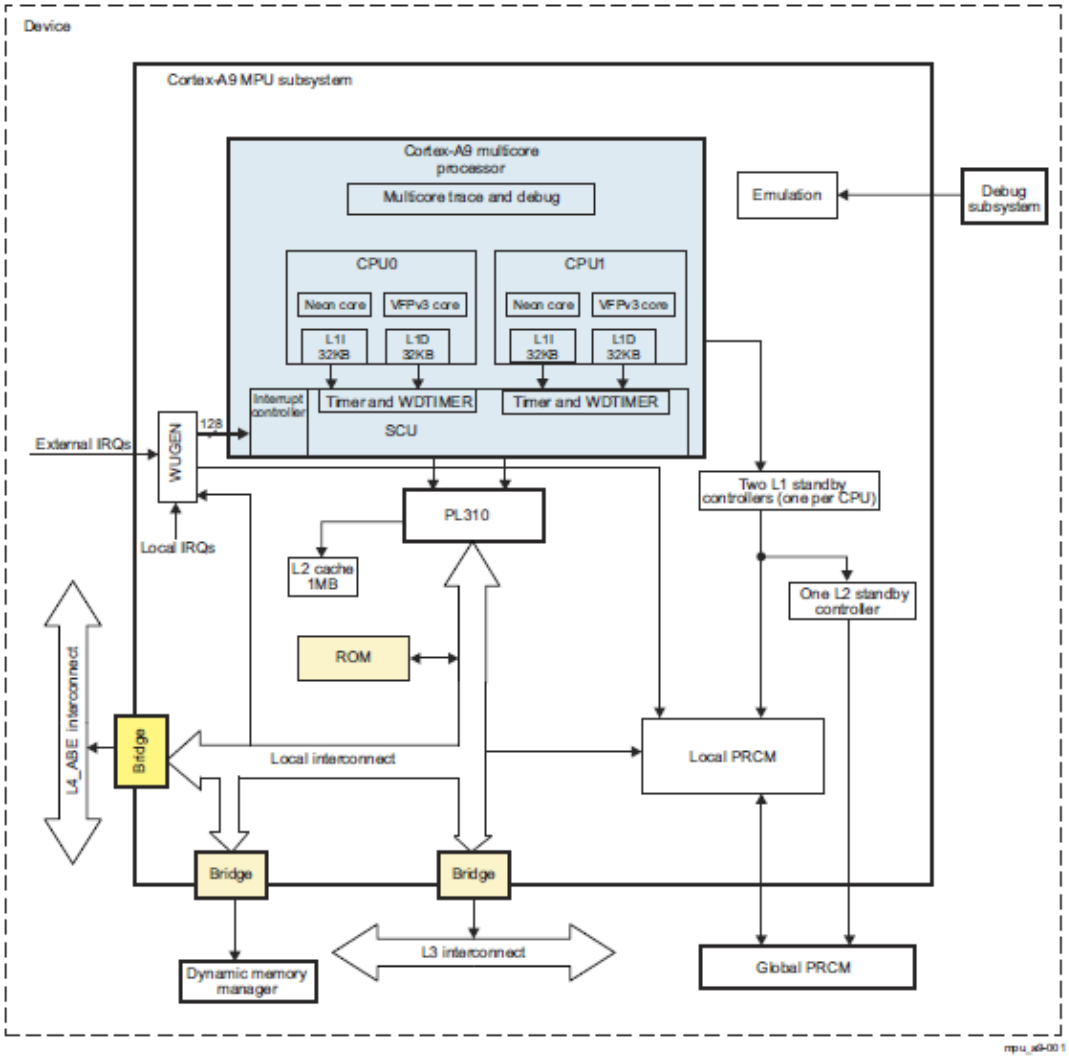


Figure 2-1 Cortex-A9 processor top-level diagram

Cortex-A9 Technical Reference Manual, p.2-2.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

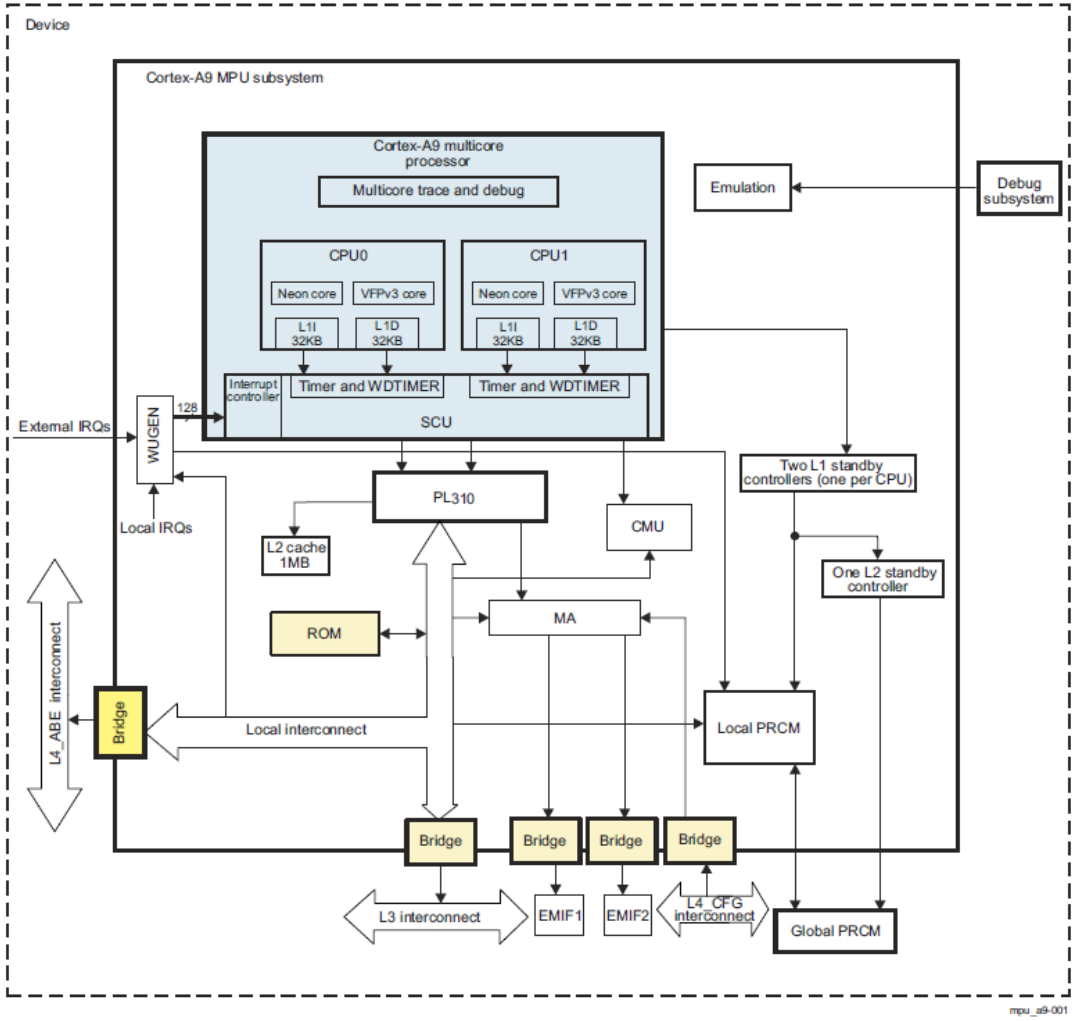
Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4430 Technical Reference Manual, p. 1077.

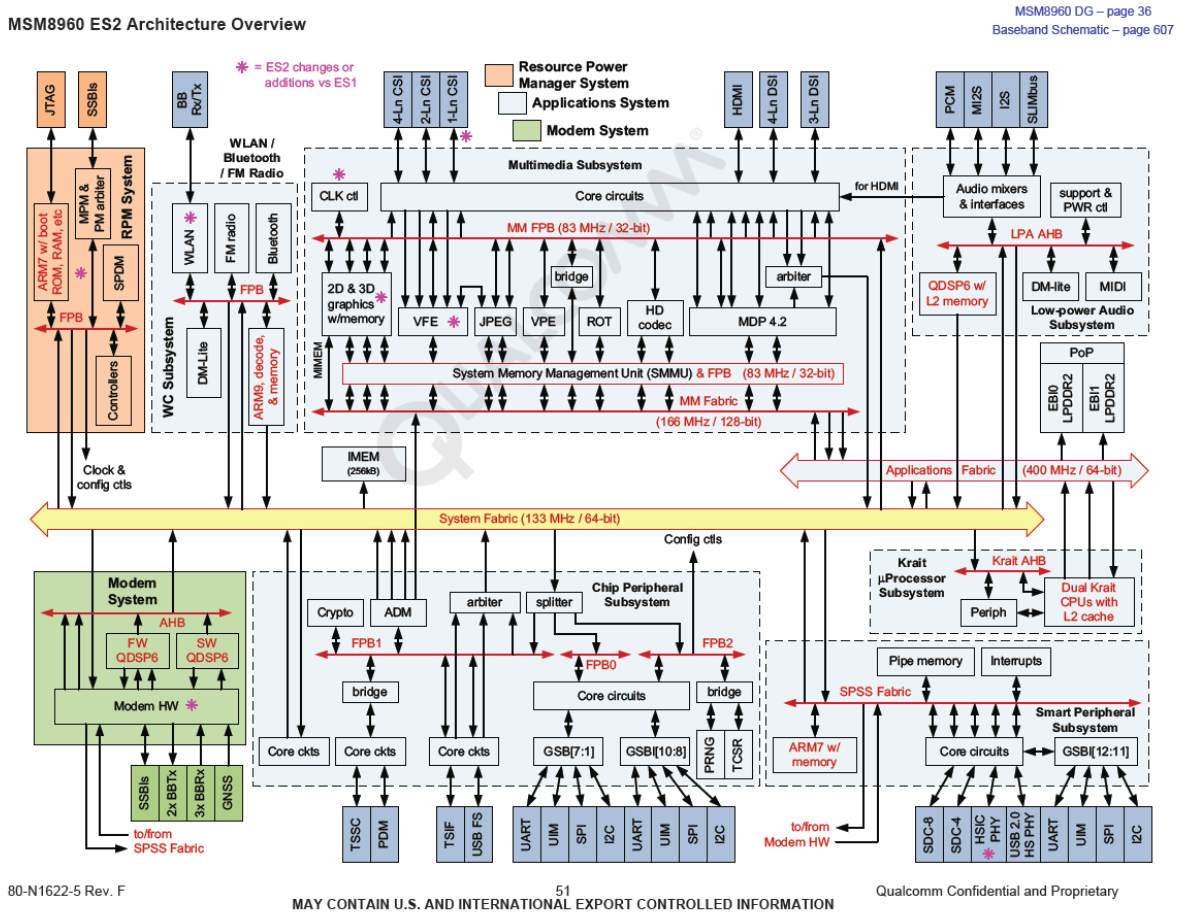
EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4470 Technical Reference Manual, p. 1091.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG



MSM8960 Chipset Schematics and Design Guidelines, p. 50.

When executing Jazelle instructions, the stack pointer is located in register R6.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

	In Java State, the processor assigns several ARM registers to functions specific to the Java machine (e.g., R6= stack pointer, R0-R3= top elements of stack, R4=local variable 0). This hardware reuse contributes to the small size of the additional logic (12k gates) required to implement the Java machine, and keeps all of the state required by the Jazelle extension in ARM registers, In addition, it ensures compatibility with existing operating systems, interrupt handlers and exception code. ARM White Paper, Accelerating to Meet the Challenges of Embedded Java™ at 3.
said stack pointer pointing into said first push down stack,	The stack pointer points to the stack. **SP, the Stack Pointer** Register R13 is used as a pointer to the active stack. In Thumb code, most instructions cannot access SP. The only instructions that can access SP are those designed to use SP as a stack pointer. The use of SP for any purpose other than as a stack pointer is deprecated. **Note** Using SP for any purpose other than as a stack pointer is likely to break the requirements of operating systems, debuggers, and other software systems, causing them to malfunction. ARMv7 Architecture Reference Manual, p. A2-11. When executing Jazelle instructions, the pointer in R6 points to the stack.

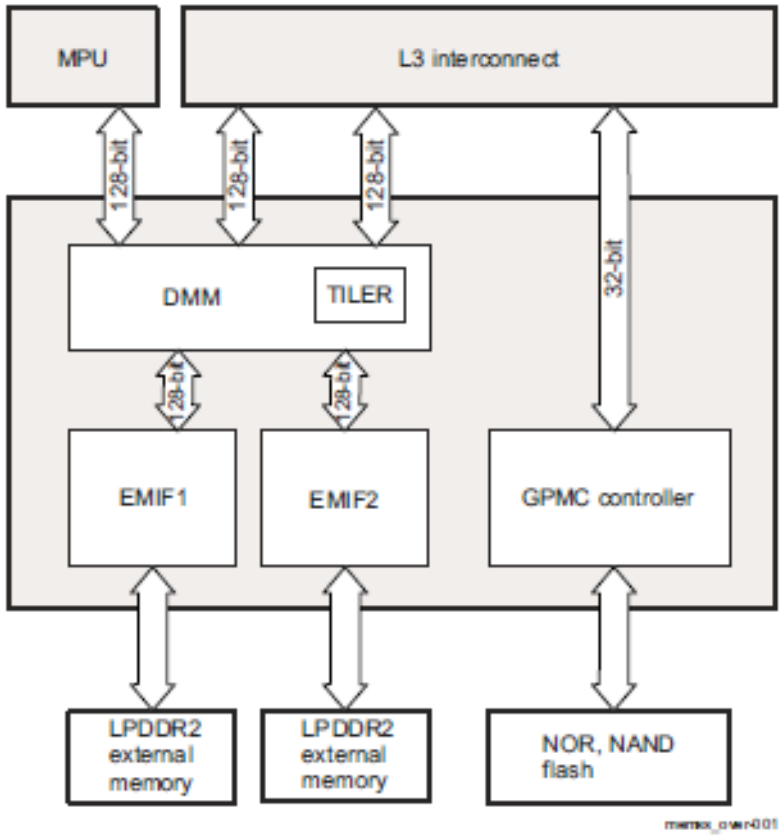
EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

	In Java State, the processor assigns several ARM registers to functions specific to the Java machine (e.g., R6= stack pointer, R0-R3= top elements of stack, R4=local variable 0). This hardware reuse contributes to the small size of the additional logic (12k gates) required to implement the Java machine, and keeps all of the state required by the Jazelle extension in ARM registers, In addition, it ensures compatibility with existing operating systems, interrupt handlers and exception code. ARM White Paper, Accelerating to Meet the Challenges of Embedded Java™ at 3.
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,	On information and belief, the internal buses of the Accused Microprocessors are connected to a memory controller.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Figure 15-1 shows a functional diagram of all memory subsystems in the device.

Figure 15-1. Memory Subsystem Functional Diagram

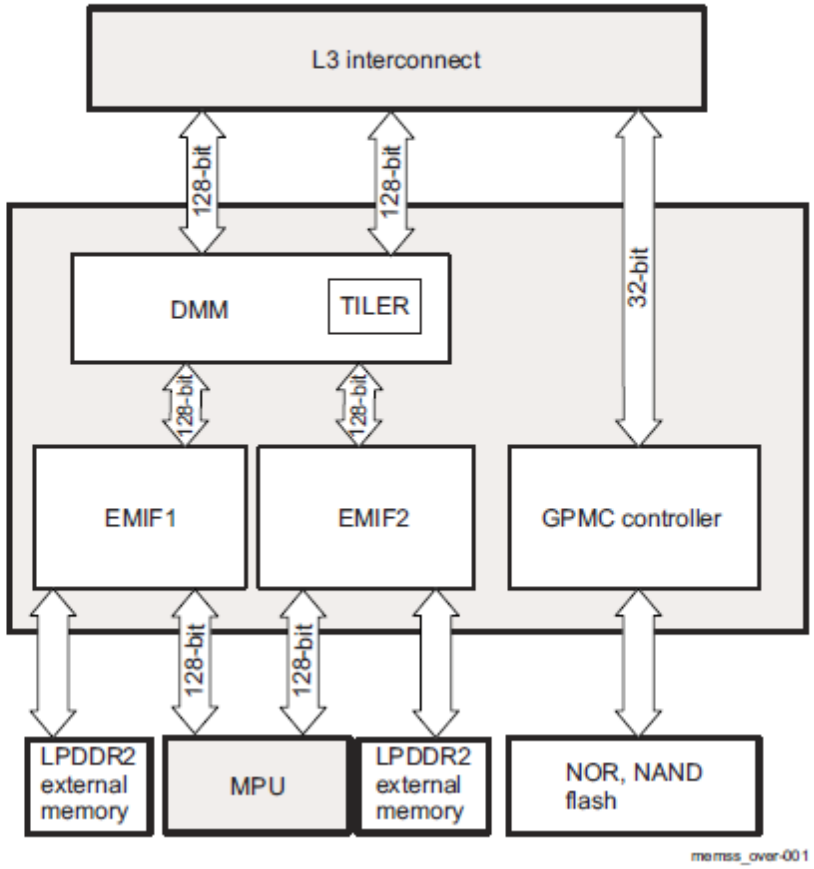


OMAP4430 Technical Reference Manual, p. 3207.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

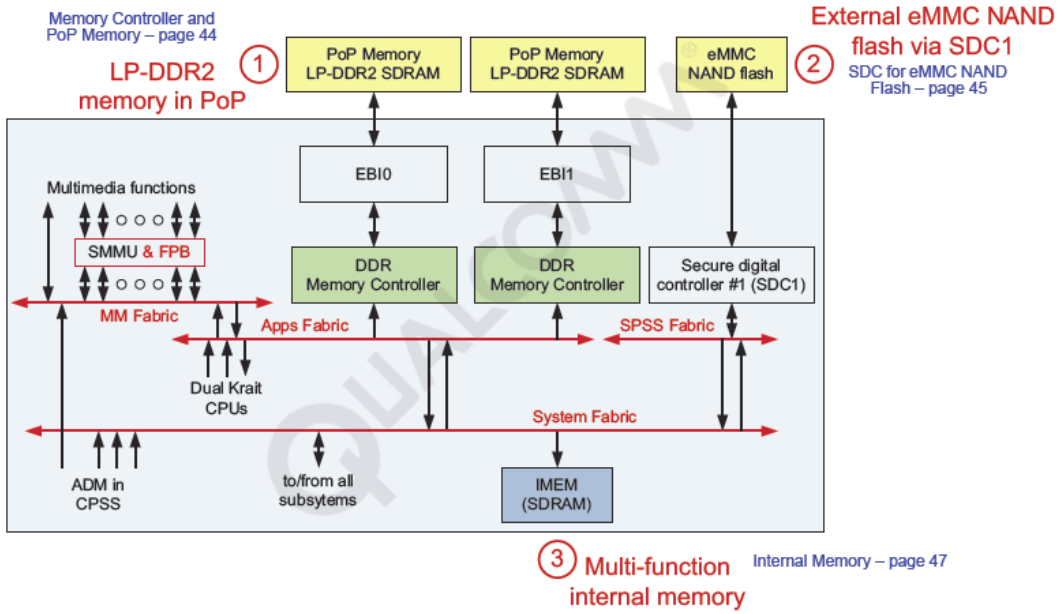
Figure 16-1 shows a functional diagram of all memory subsystems in the device.

Figure 16-1. Memory Subsystem Functional Diagram



OMAP4470 Technical Reference Manual, p. 3302.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG



MSM8960 Chipset Schematics and Design Guidelines, p. 41.

On information and belief, the internal buses are connected to a register of the return stack which serves as a Y register. The internal buses are also connected to the Register File which contains several general purpose registers that may serve as an X register and the program counter.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

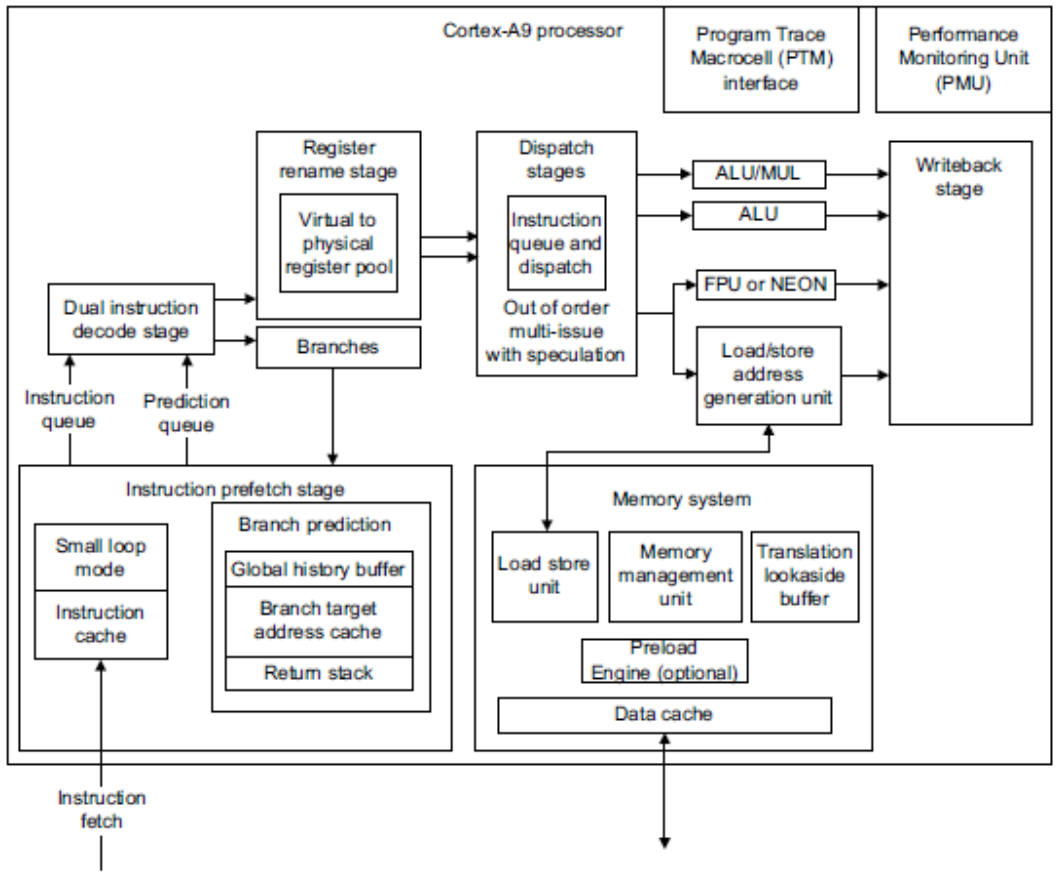


Figure 2-1 Cortex-A9 processor top-level diagram

Cortex-A9 Technical Reference Manual, p.2-2.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

	PC, the Program Counter Register R15 is the program counter: • When executing an ARM instruction, PC reads as the address of the current instruction plus 8. • When executing a Thumb instruction, PC reads as the address of the current instruction plus 4. • Writing an address to PC causes a branch to that address. In Thumb code, most instructions cannot access PC. ARMv7 Architecture Reference Manual, p. A2-11.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Figure 2-1 shows a top-level functional diagram of the Cortex-A15 MPCore processor.

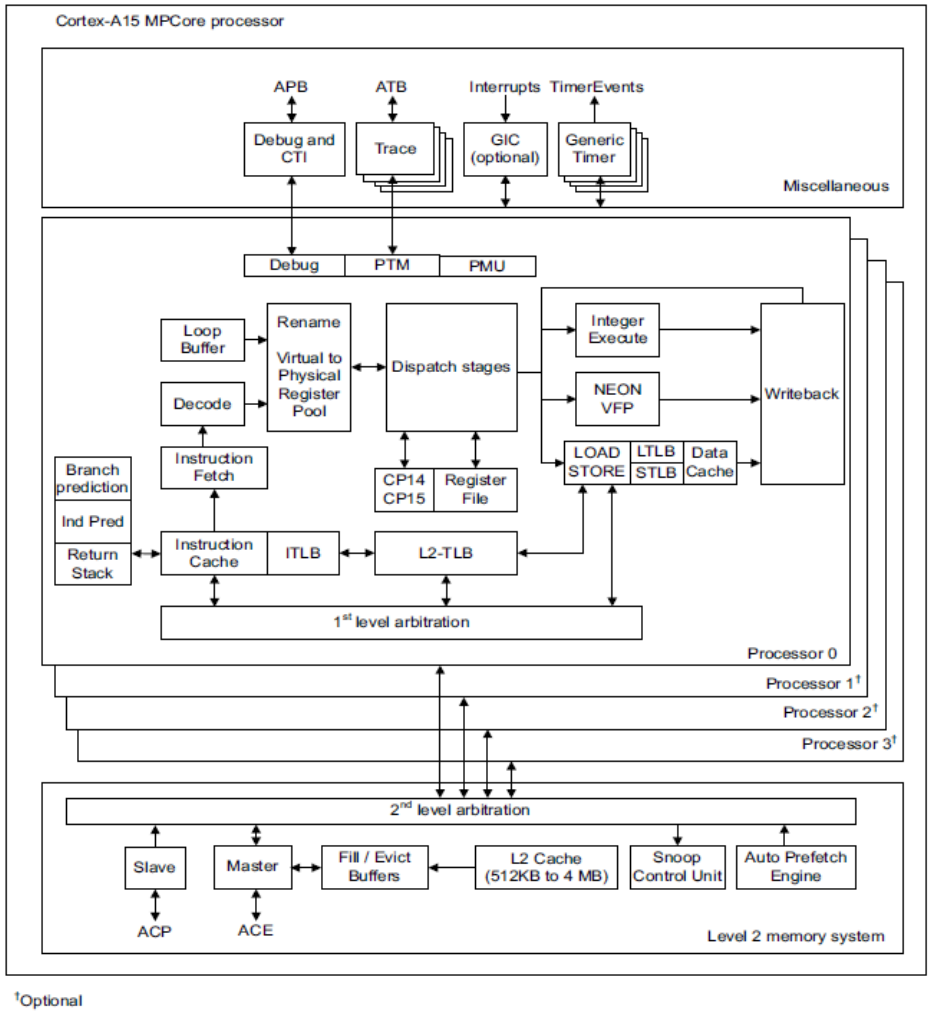
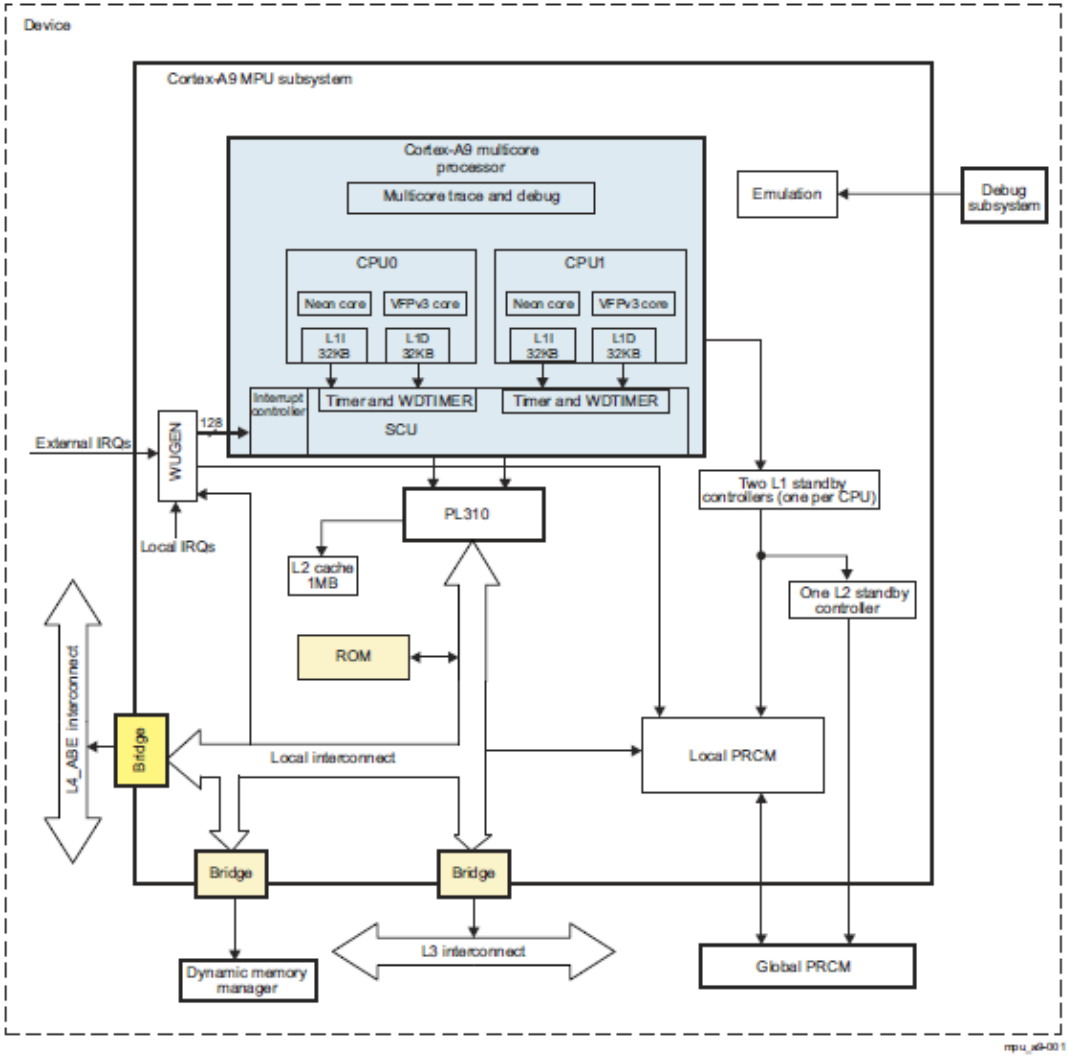


Figure 2-1 Block diagram

Cortex-A15 MPCore Processor Technical Reference Manual p. 2-2.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

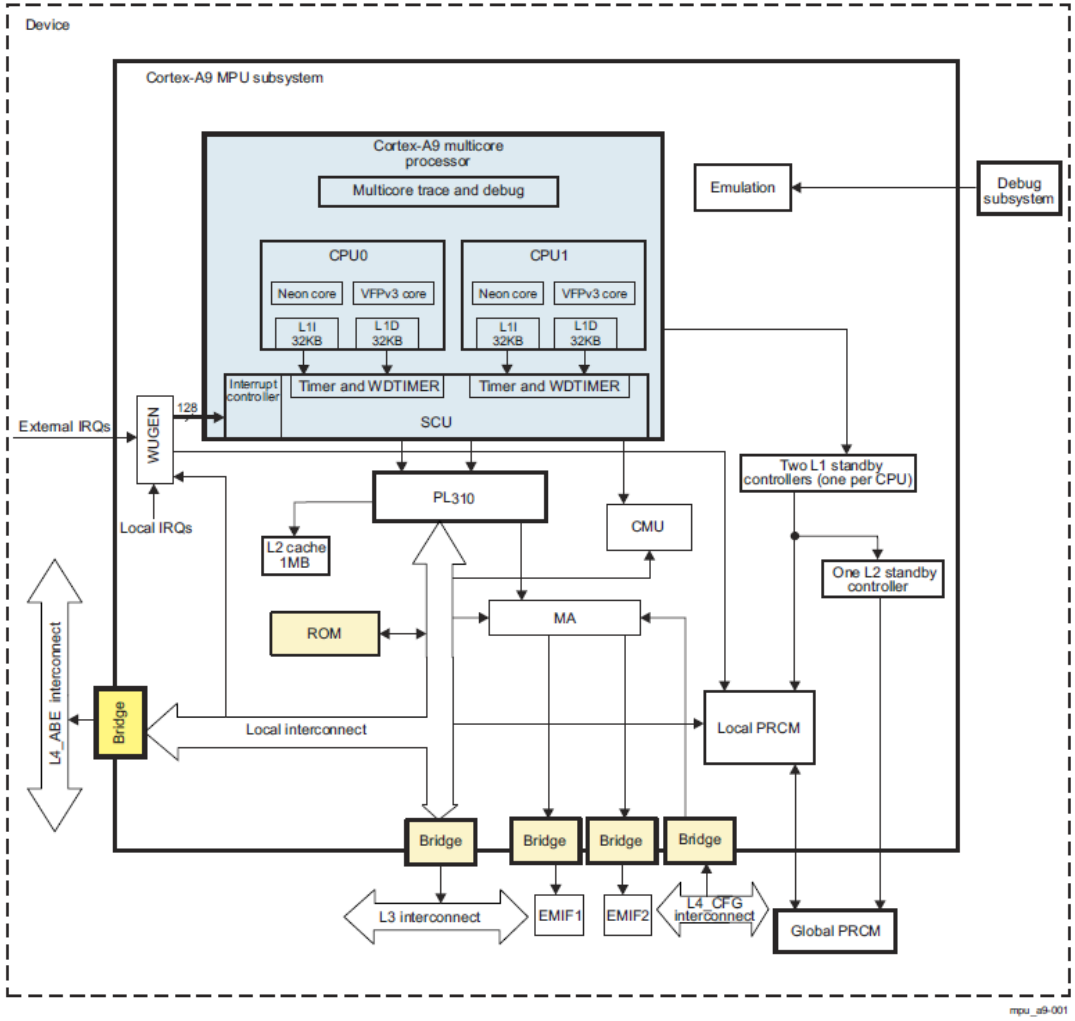
Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4430 Technical Reference Manual, p. 1077.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4470 Technical Reference Manual, p. 1091.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

	When executing Jazelle instructions, register R4 may be used as an X register. In Java State, the processor assigns several ARM registers to functions specific to the Java machine (e.g., R6= stack pointer, R0-R3= top elements of stack, R4=local variable 0). This hardware reuse contributes to the small size of the additional logic (12k gates) required to implement the Java machine, and keeps all of the state required by the Jazelle extension in ARM registers. In addition, it ensures compatibility with existing operating systems, interrupt handlers and exception code. ARM White Paper, Accelerating to Meet the Challenges of Embedded Java™ at 3.
said Y register, X register and program counter providing outputs to an internal address bus,	On information and belief, the program counter, X register, and Y register provide outputs to the internal buses of the Accused Microprocessors. Plaintiff contends that this claim element is literally present. However, in the event that this claim element is not found to be literally present, Plaintiffs contend that the above identified buses, either alone or in combination, are equivalent to "an internal address bus" and any differences are insubstantial. In particular, the buses perform the same function (*i.e.*, provide a means to communicate addresses between the ALU, memory, and registers), in substantially the same way (*i.e.*, by connecting them electrically), and have the same result (*i.e.*, the various connected components transmit addresses to each other).
said internal address bus providing inputs to said memory controller and to an incrementer,	On information and belief, the internal buses of the Accused Microprocessors provide inputs to the memory controller to provide access to RAM. In the case that the Program Counter, X register, or Y register holds an address pointing to external memory, the internal buses provides the address to the memory controller. The internal buses also provide input to an incrementer.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

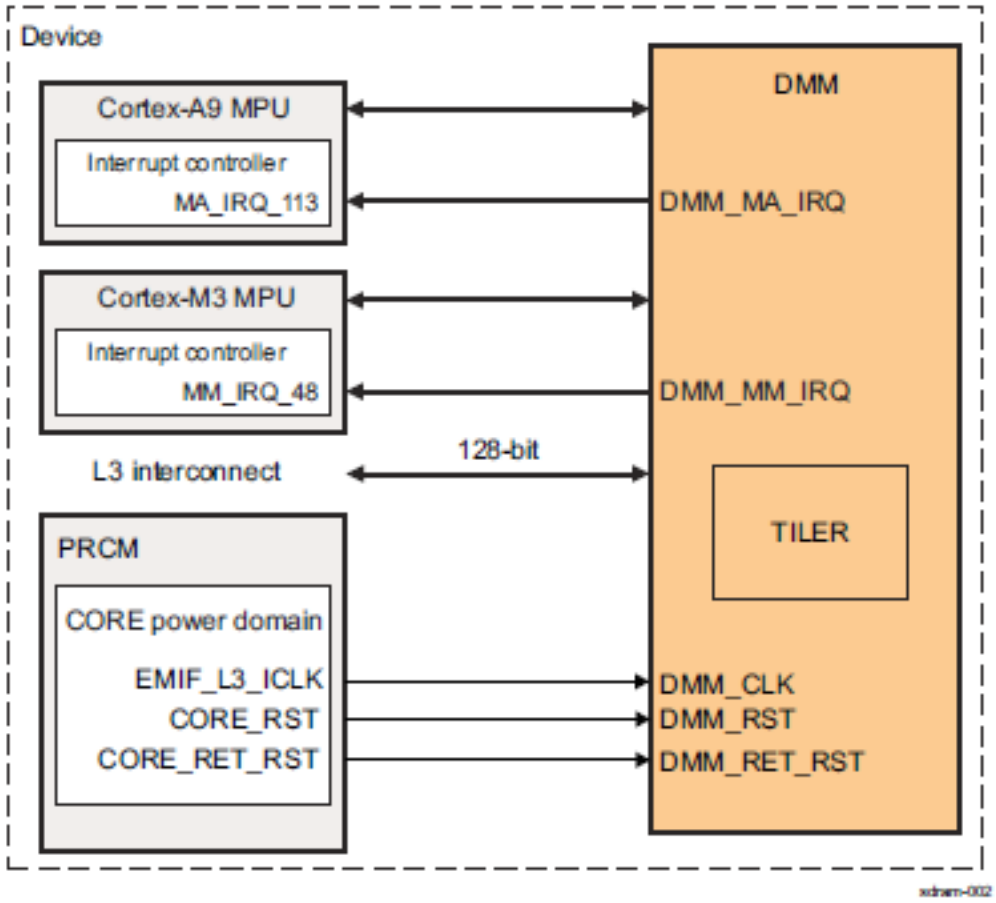
	T == 0 32-bit instructions are fetched (and the PC is incremented by four) and are executed as ARM instructions. **T == 1** 16-bit instructions are fetched (and the PC is incremented by two) and are executed as Thumb instructions. See ARM Architecture Reference Manual at A6-2 . **A3.1.1 Address incrementing and address space overflow** When a processor performs normal sequential execution of instructions, it effectively calculates: `(address_of_current_instruction) + (size_of_executed_instruction)` after each instruction to determine which instruction to execute next. **Note** The size of the executed instruction depends on the current instruction set, and might depend on the instruction executed. If this address calculation overflows the top of the address space, the result is UNPREDICTABLE. In other words, a program must not rely on sequential execution of the instruction at address `0x00000000` after the instruction at address: • `0xFFFFFFFC`, when a 4-byte instruction is executed • `0xFFFFFFFE`, when a 2-byte instruction is executed • `0xFFFFFFFF`, when a single byte instruction is executed. ARMv7 Technical Reference Manual p. A3-2
said incrementer being connected to said internal data bus,	On information and belief, the incrementer is connected to the internal buses in order to increment the instruction address. The internal buses also provide input to an incrementer.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

	T == 0 32-bit instructions are fetched (and the PC is incremented by four) and are executed as ARM instructions. **T == 1** 16-bit instructions are fetched (and the PC is incremented by two) and are executed as Thumb instructions. See ARM Architecture Reference Manual at A6-2 . **A3.1.1 Address incrementing and address space overflow** When a processor performs normal sequential execution of instructions, it effectively calculates: `(address_of_current_instruction) + (size_of_executed_instruction)` after each instruction to determine which instruction to execute next. **Note** The size of the executed instruction depends on the current instruction set, and might depend on the instruction executed. If this address calculation overflows the top of the address space, the result is UNPREDICTABLE. In other words, a program must not rely on sequential execution of the instruction at address `0x00000000` after the instruction at address: • `0xFFFFFFFC`, when a 4-byte instruction is executed • `0xFFFFFFFE`, when a 2-byte instruction is executed • `0xFFFFFFFF`, when a single byte instruction is executed. ARMv7 Technical Reference Manual p. A3-2
said direct memory access central processing unit providing inputs to said memory controller,	On information and belief, the other central processing units of the Accused Microprocessors provide inputs to the memory controller.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

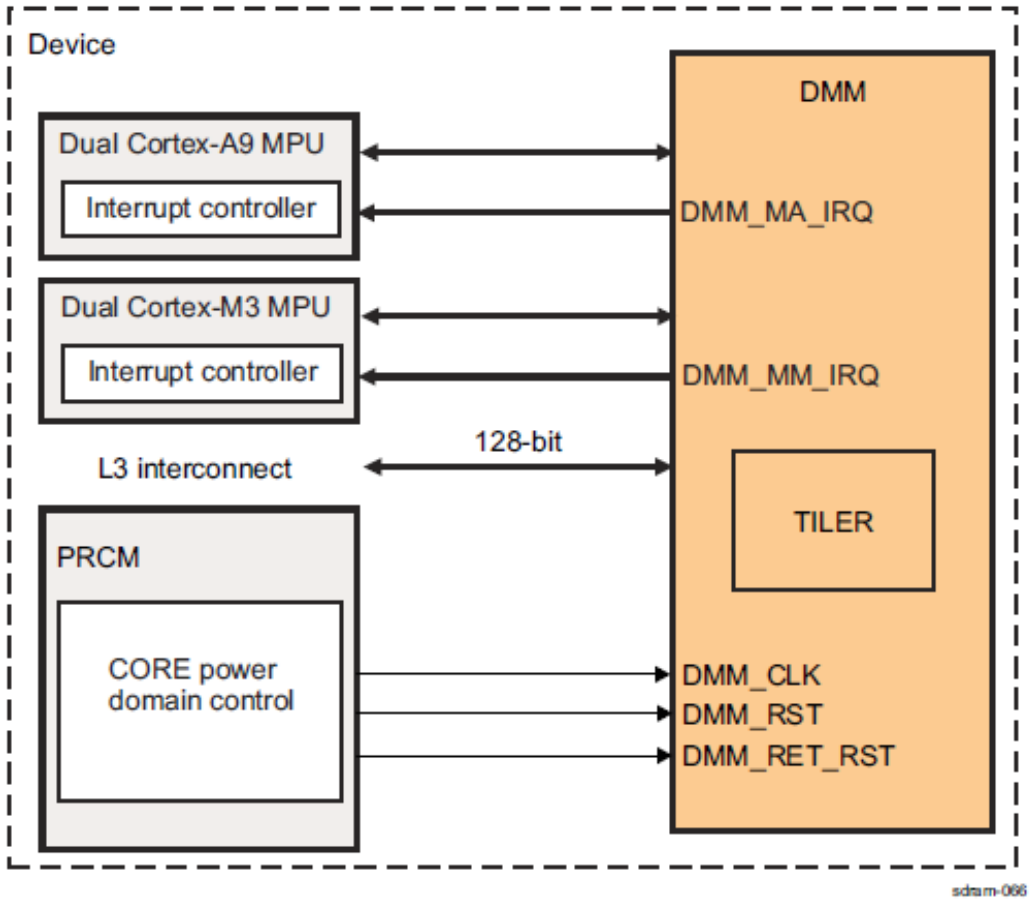
Figure 15-3. DMM Integration



OMAP4430 Technical Reference Manual, p. 3213.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Figure 16-2. DMM and Tiler Overview



OMAP4470 Technical Reference Manual, p. 3308.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

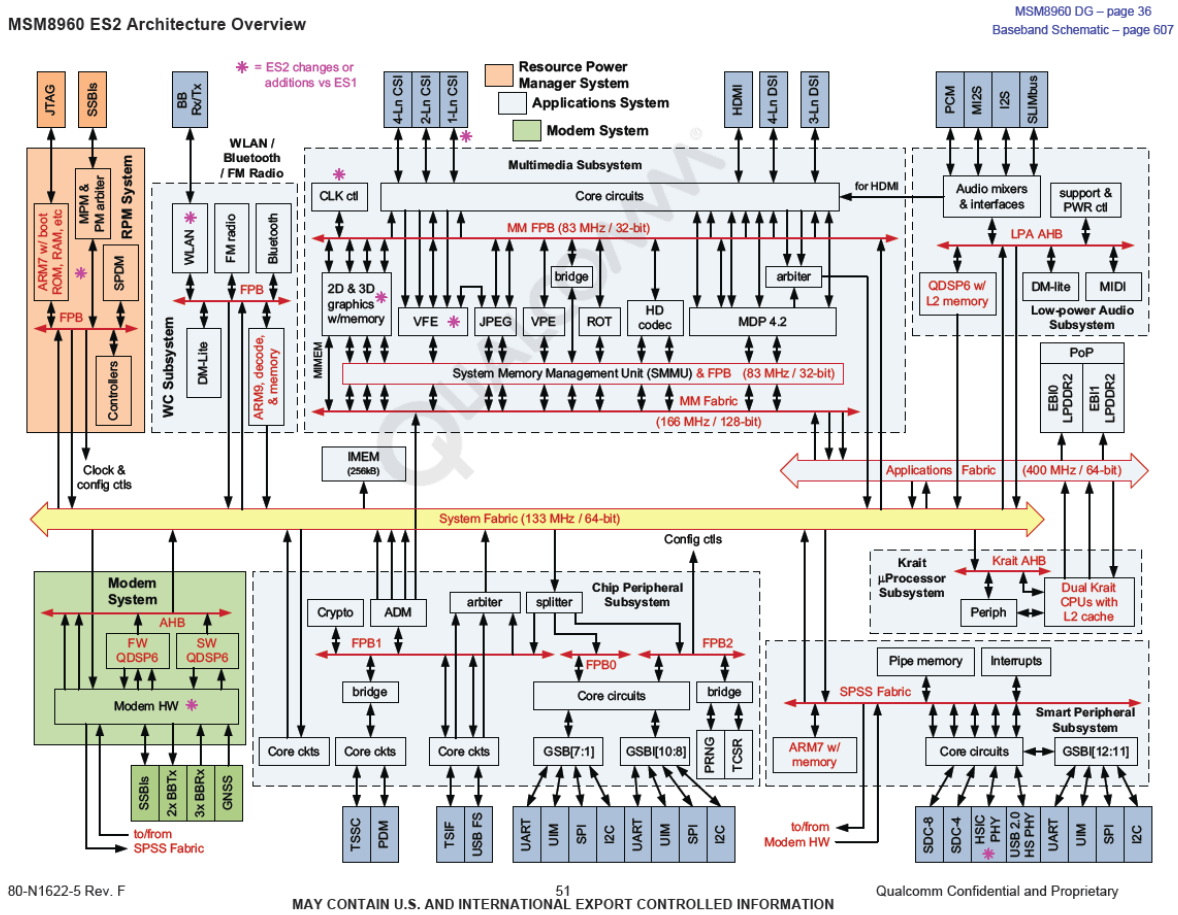
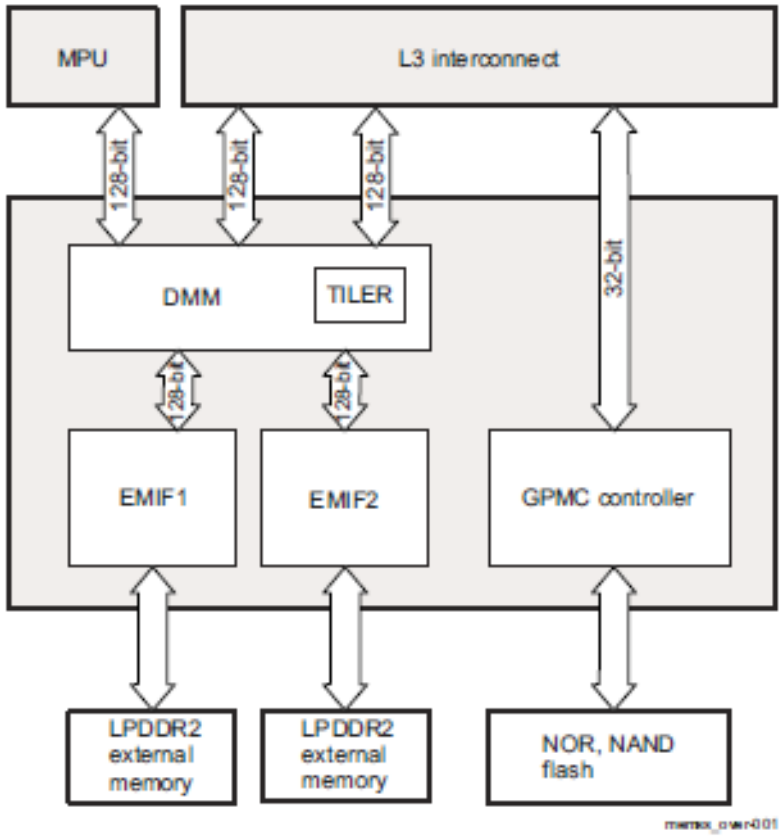
	 MSM8960 Chipset Schematics and Design Guidelines, p. 50.
said memory controller having an address/data bus and a plurality of control lines for connection to a random access memory.	On information and belief the memory controller of each Accused Microprocessor has an address/data bus and a plurality of control lines for connection to LPDDR2 random access memory.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Figure 15-1 shows a functional diagram of all memory subsystems in the device.

Figure 15-1. Memory Subsystem Functional Diagram

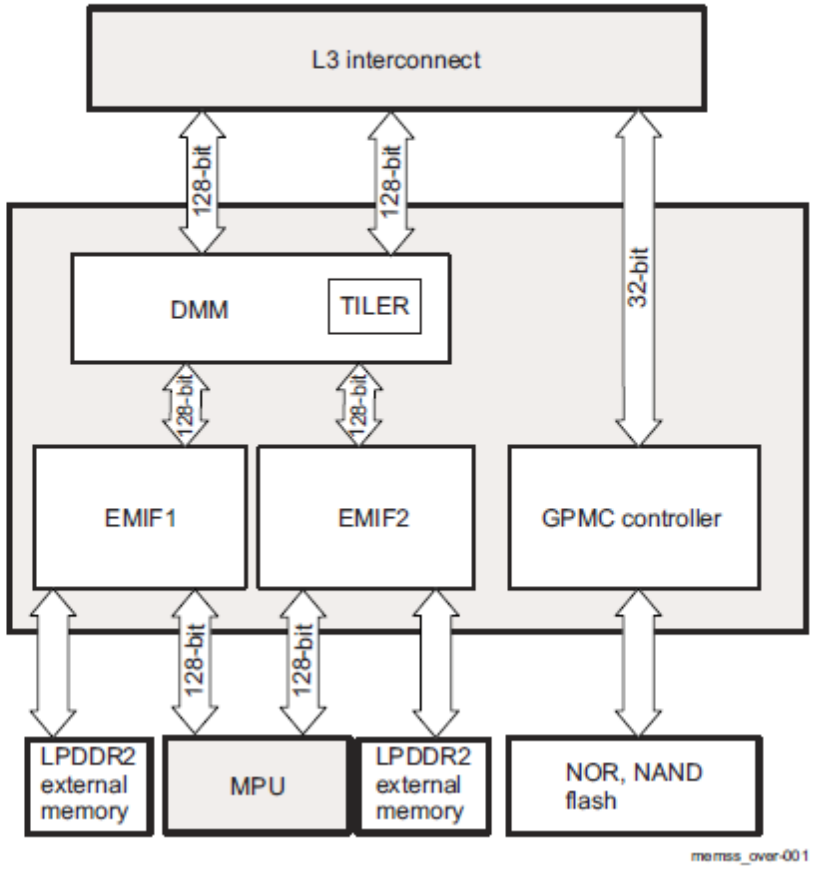


OMAP4430 Technical Reference Manual, p. 3207.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

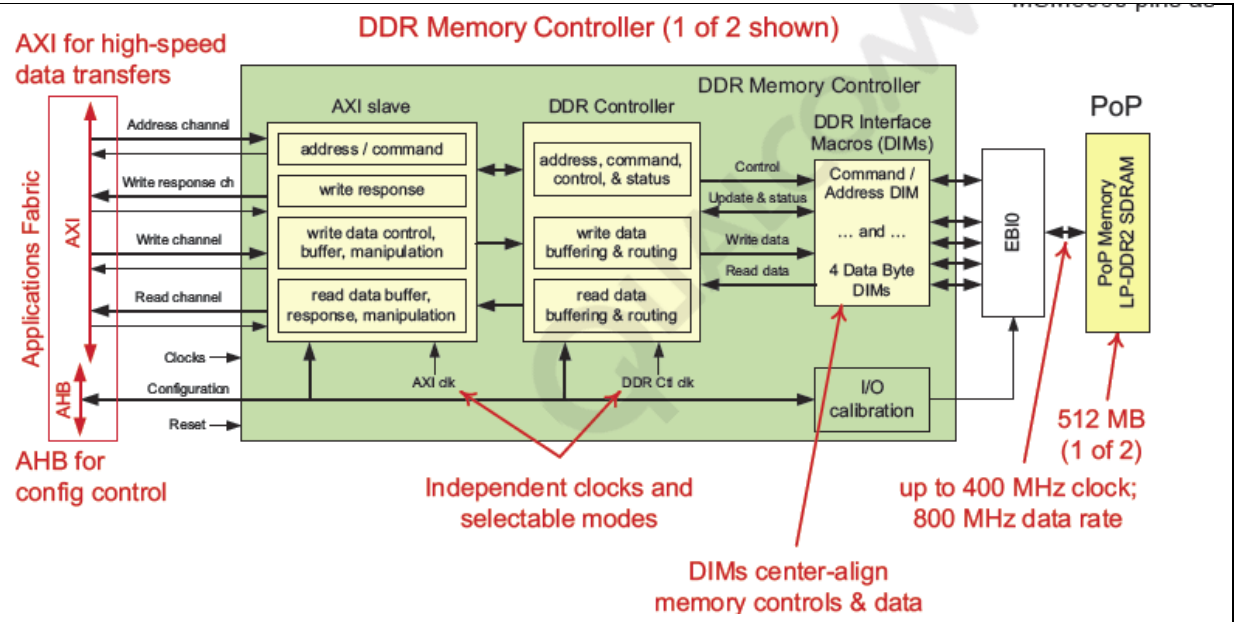
Figure 16-1 shows a functional diagram of all memory subsystems in the device.

Figure 16-1. Memory Subsystem Functional Diagram



OMAP4470 Technical Reference Manual, p. 3302.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

	 MSM8960 Chipset Schematics and Design Guidelines, p. 44.

Claim 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.	It is well known to persons of ordinary skill in the art that, due to pin limitation, DRAM memory addressing is accomplished by multiplexing the row and column addresses on the address pins. On information and belief, this is true for each of the Accused Microprocessors.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

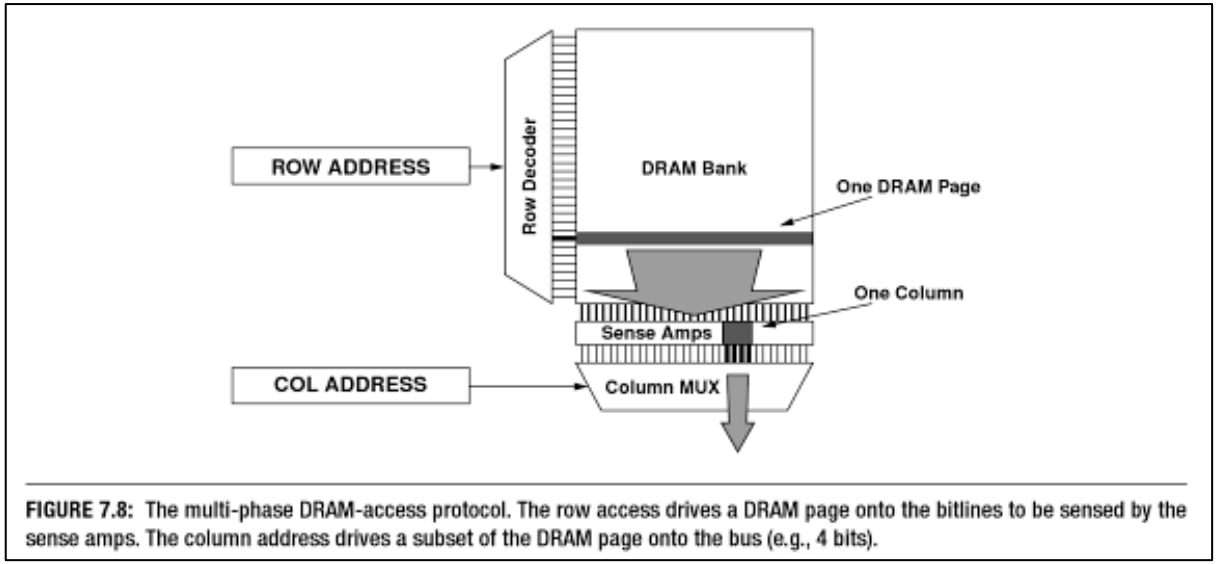


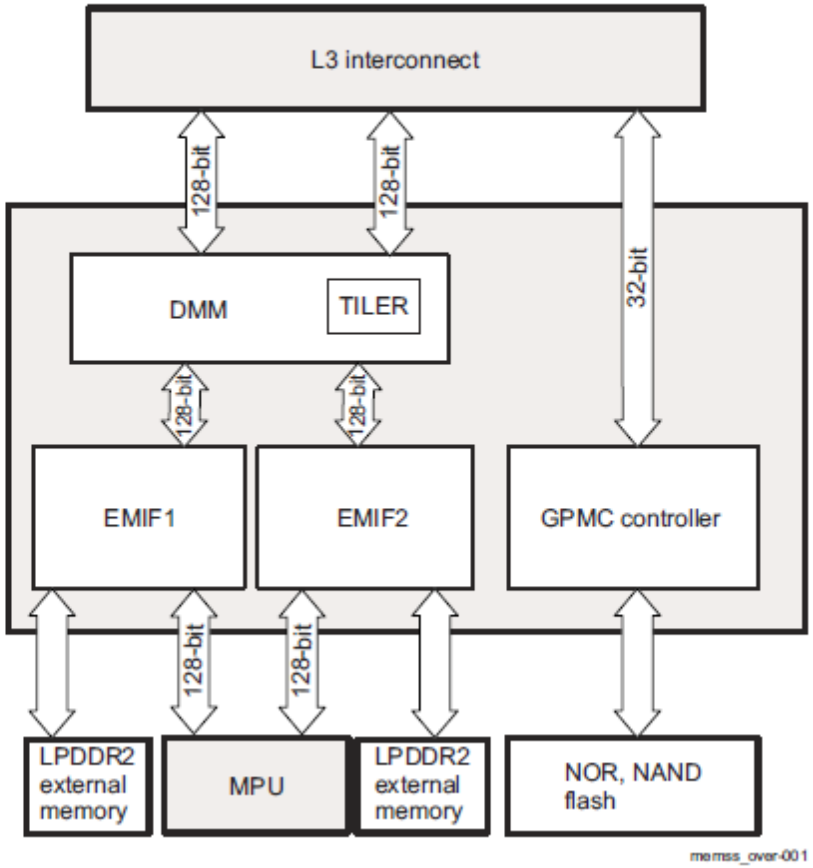
FIGURE 7.8: The multi-phase DRAM-access protocol. The row access drives a DRAM page onto the bitlines to be sensed by the sense amps. The column address drives a subset of the DRAM page onto the bus (e.g., 4 bits).

B. Jacob et al., Memory Systems: Cache, DRAM, DISK [TPLBN053358-TPLBN054374] at 321.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Figure 16-1 shows a functional diagram of all memory subsystems in the device.

Figure 16-1. Memory Subsystem Functional Diagram



OMAP4470 Technical Reference Manual, p. 3302.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Claim 13	
The microprocessor of claim 11 additionally comprising	
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.	The ARM cores include an external instruction fetch that draws multiple sequential instructions and supplies them to the central processing unit in a single memory cycle. On information and belief the Accused Microprocessors have an apparatus for fetching and supplying instructions in parallel. **A3.9.1 Introduction to caches** A cache is a block of high-speed memory that contains a number of entries, each consisting of: • main memory address information, commonly known as a *tag* • the associated data. Caches are used to increase the average speed of a memory access. Cache operation takes account of two principles of locality: **Spatial locality** An access to one location is likely to be followed by accesses to adjacent locations. Examples of this principle are: • sequential instruction execution • accessing a data structure. ARMv7 Reference Manual at A3-51 (165). **A6.1 Thumb instruction set encoding** The Thumb instruction stream is a sequence of halfword-aligned halfwords. Each Thumb instruction is either a single 16-bit halfword in that stream, or a 32-bit instruction consisting of two consecutive halfwords in that stream. ARMv7 Reference Manual at A6-2 (240). **Execution stream** The stream of instructions that would have been executed by sequential execution of the program. ARMv7 Reference Manual at Glossary-5 (2149).

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Claim 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 system clock being provided in a single integrated circuit.	On information and belief, in each Accused Microprocessor, the PLL clocking the CPU consists of a Voltage Controlled Oscillator (VCO) or a Current Controlled Oscillator (ICO), which is a **ring 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. *See* http://en.wikipedia.org/wiki/Ring_oscillator. For example, generally in TI microprocessors, PLLs have voltage-controlled oscillators, which are circuits capable of maintaining an alternating output. TI has published papers regarding its DPLLs confirming the use of ring oscillators.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

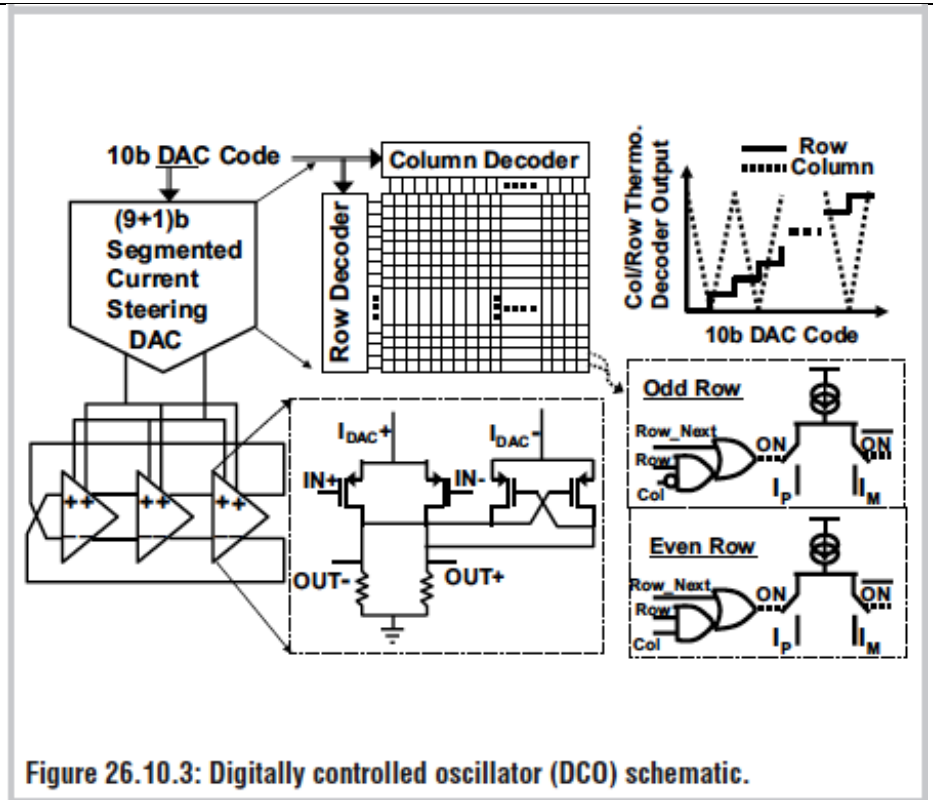
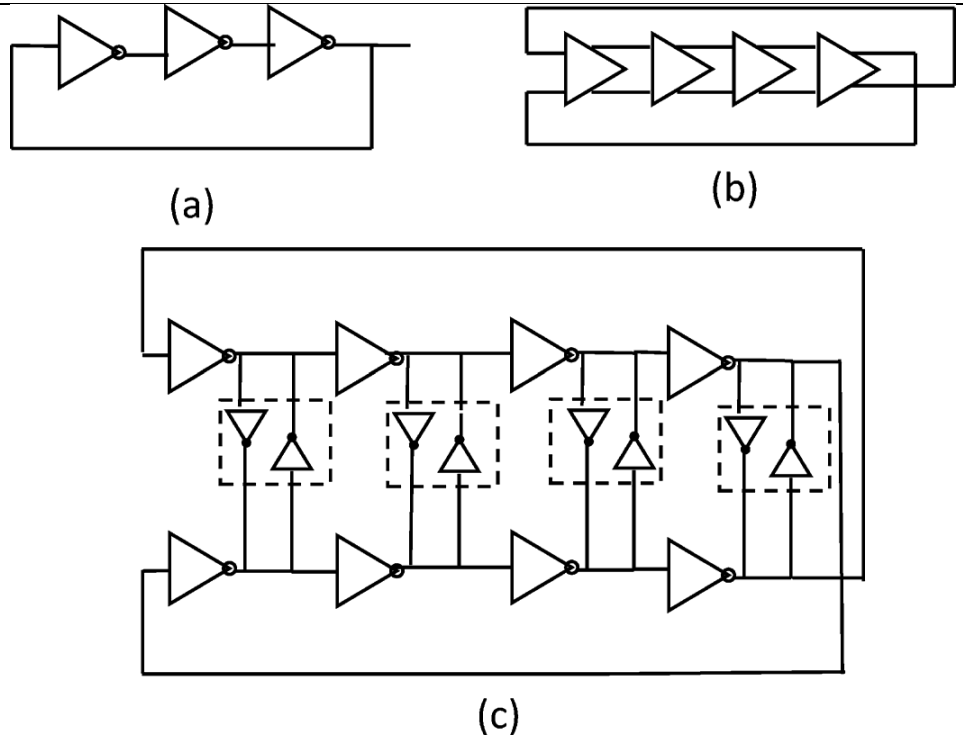


Figure 26.10.3: Digitally controlled oscillator (DCO) schematic.

TPL853_02993475-84, Fig. 26.10.3 (ISSCC 2004 / Session 26 / *Optical and Fast IO 26.10*) (CX-0170).

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG



TPL853_02993485-96, Fig. 3 (CX-0171). "Figs. 17 and 18 show the die micrographs of two Digital PLLs implemented in 65 nm and 45 nm CMOS. . . . Both use a simple three stage ring oscillator shown in Fig. 3(a)." TPL853_02993494.

The TI OMAP4 family of processors is also implemented in 45 nm CMOS technology: "The OMAP4470 high-performance multimedia application device is based on enhanced OMAP architecture and uses 45-nm technology." TPLBN043746 (OMAP4470 Technical Ref Manual). *See also* TPLBN037535 (OMAP4430 Tech Ref Manual).

Furthermore, on information and belief, each of the Accused Microprocessors contains a

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

	DPLL that outputs a clock signal for the MPU. Within each of these DPLLs is either a single-ended or differential ring oscillator. This component of the PLL generates an oscillation (*i.e.*, a clock signal). The ring oscillator is able to produce an oscillation due to the presence of an odd number of inversions arranged in a loop. *See also HTC Corporation v. Technology Properties Limited*, 08-cv-00882-PSG, Trial Tr. 341:13-347:1, 333:20-374:16 (Haroun testimony). On information and belief, the only input to the ring oscillators in the Accused Microprocessors is a current. In addition, while the bias DAC ("digital to analog converter") in the DPLL may receive a digital word, it does not pass this control signal on to the ring oscillator. Indeed, without the bias component of the PLL, the ring oscillators in the Accused Microprocessors would still output an oscillation. Barring the application of dividers to the ring oscillator's output, the only way to change the frequency of the ring oscillator is to change the input current. In other words, the ring oscillator will always generate a clock signal as long as a current is applied to it.

Claim 19	
The microprocessor of claim 11	
in which said first push down stack has a first plurality of stack elements configured as latches,	On information and belief, the first plurality of stack elements is configured as latches (register file or L1 data cache).

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

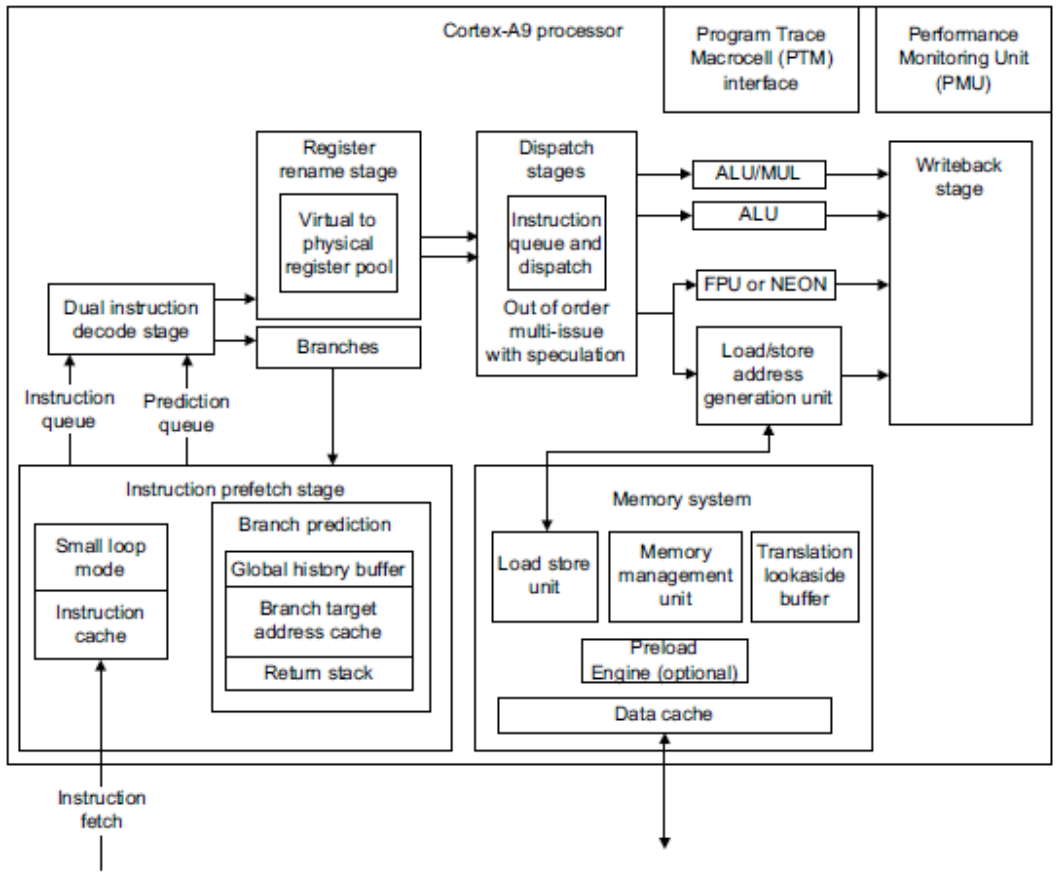


Figure 2-1 Cortex-A9 processor top-level diagram

Cortex-A9 Technical Reference Manual, p.2-2.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

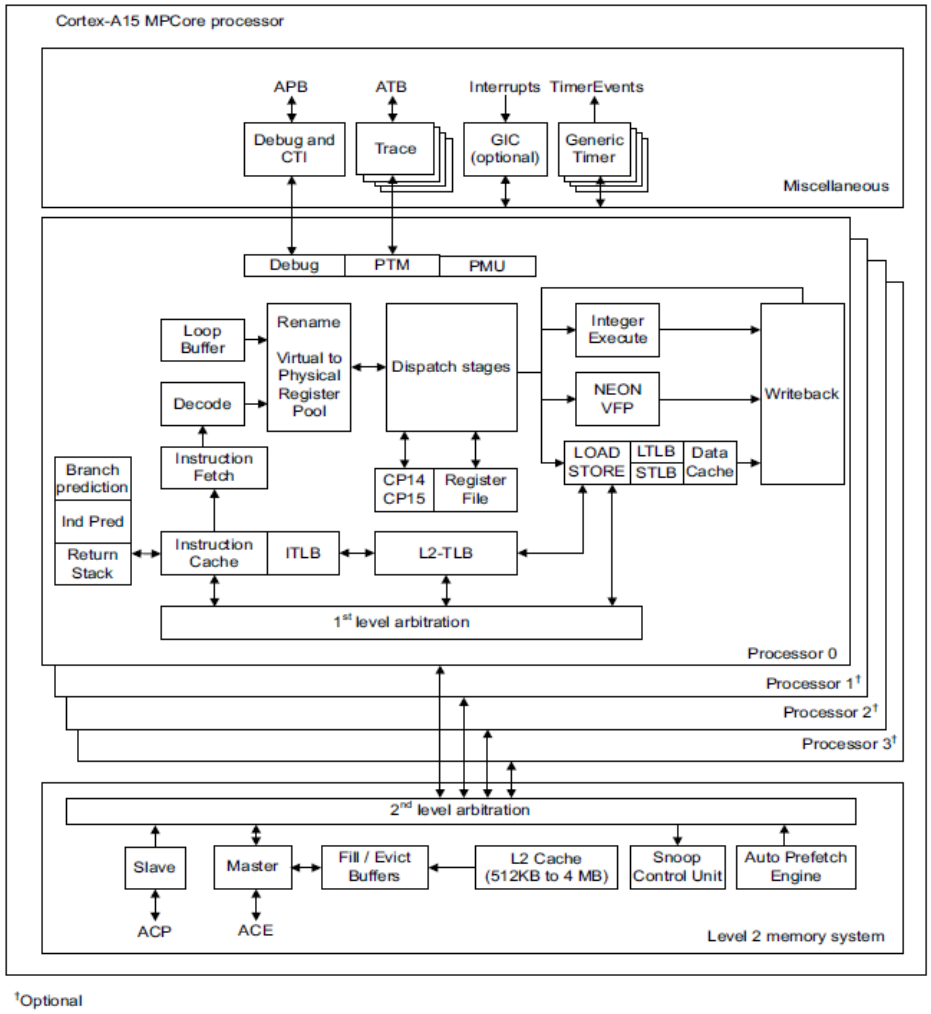
	Figure 2-1 shows a top-level functional diagram of the Cortex-A15 MPCore processor.  **Figure 2-1 Block diagram** Cortex-A15 MPCore Processor Technical Reference Manual p. 2-2.
a second plurality of stack elements	On information and belief, the second plurality of stack elements is configured as random

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

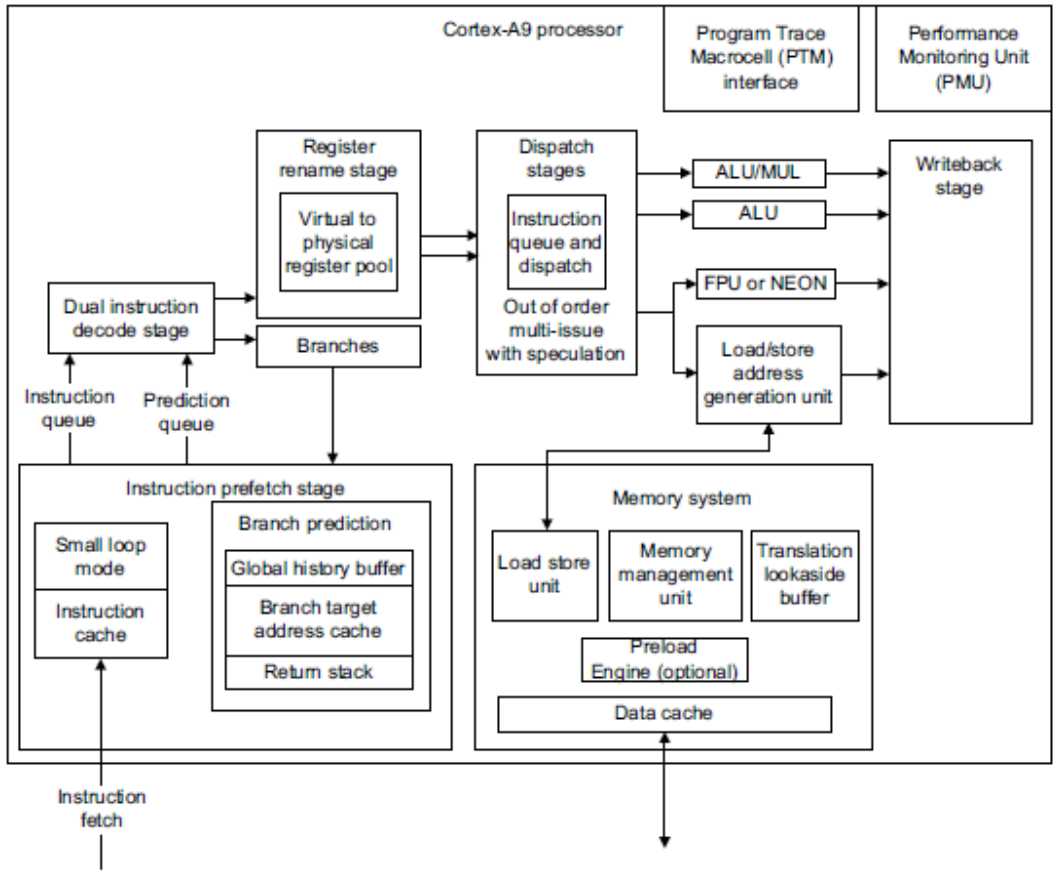
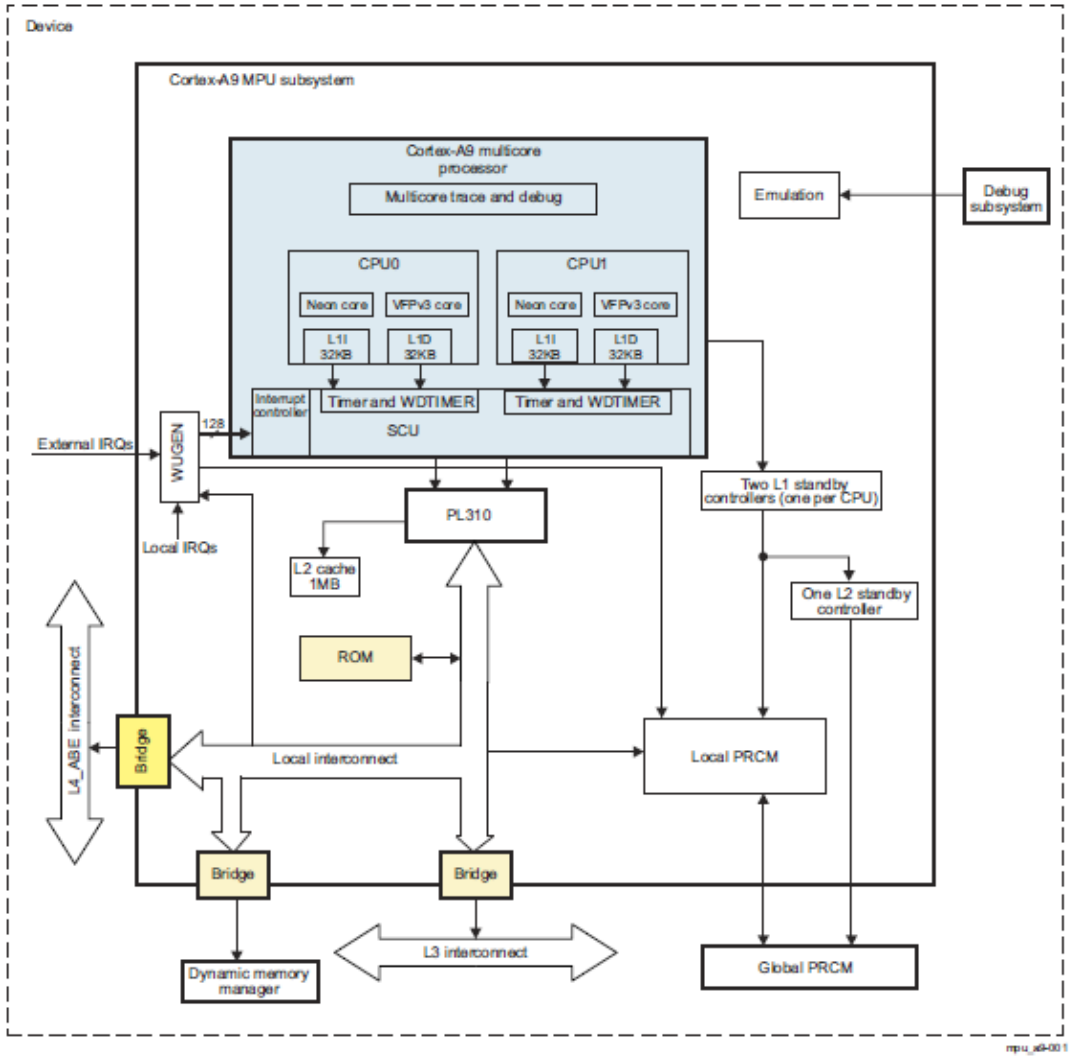
configured as a random access memory,	access memory (L1 cache or L2 cache).  **Figure 2-1 Cortex-A9 processor top-level diagram** Cortex-A9 Technical Reference Manual, p.2-2.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

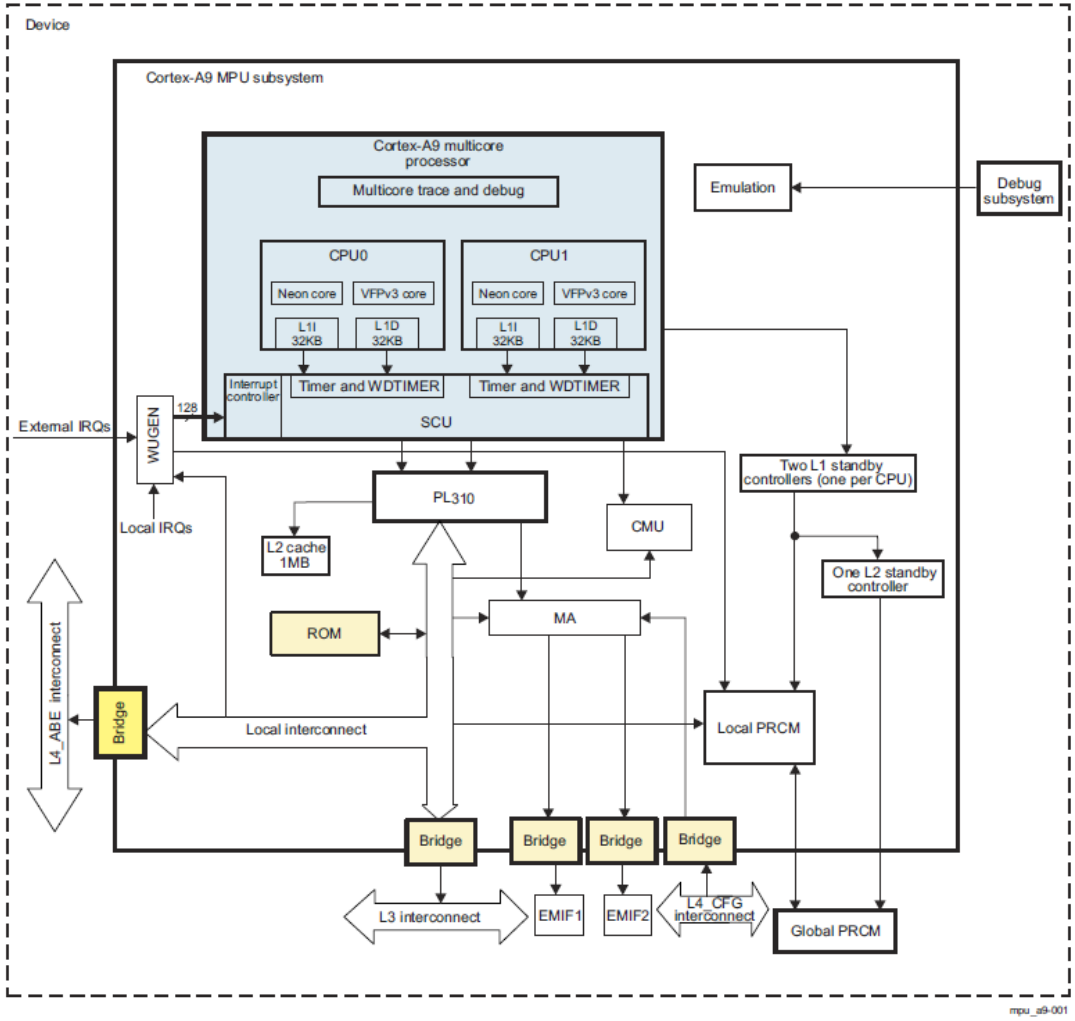
Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4430 Technical Reference Manual, p. 1077.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4470 Technical Reference Manual, p. 1091.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

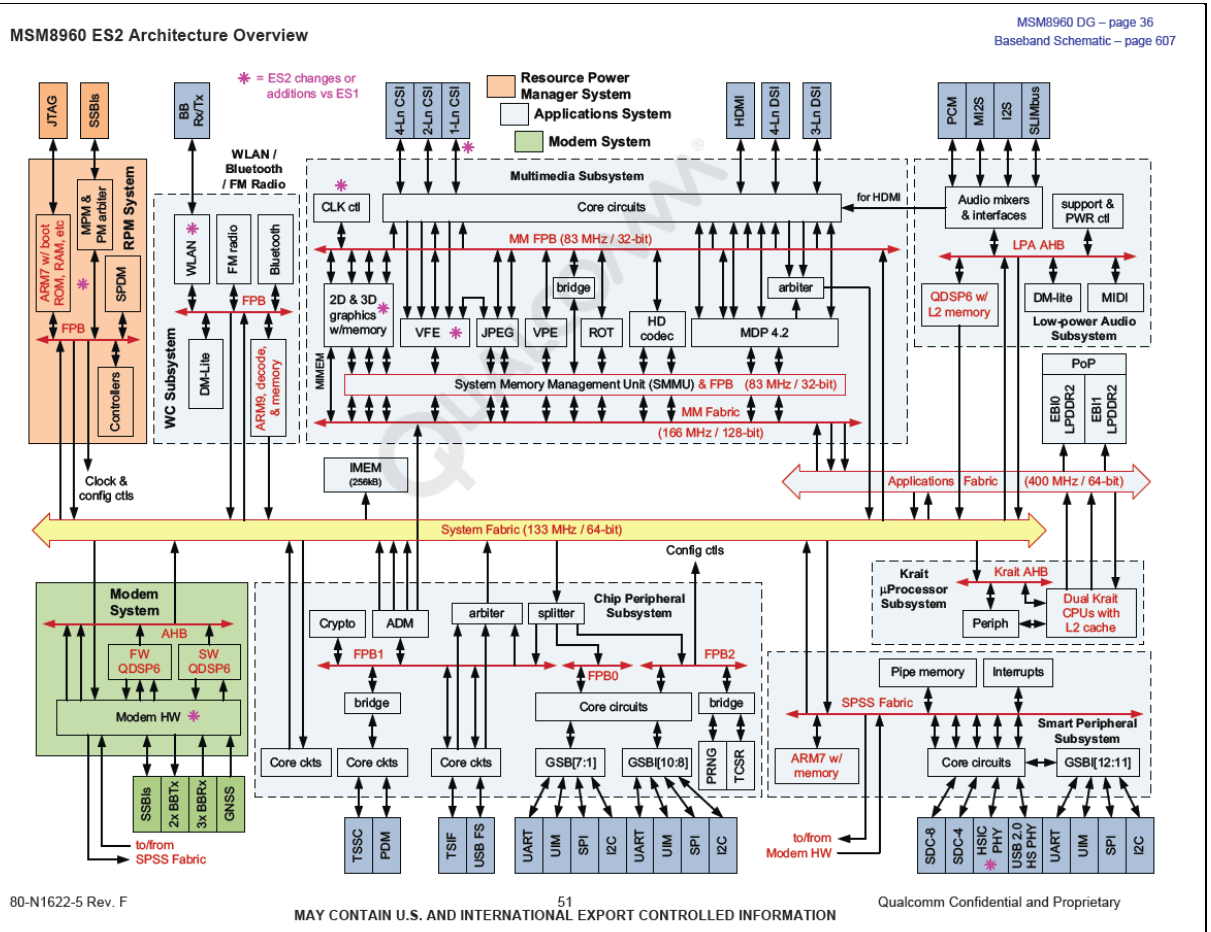
	 MSM8960 Chipset Schematics and Design Guidelines, p. 50.
said first and second plurality of stack elements and said central processing unit being provided in a single integrated circuit,	On information and belief, both the first and the second plurality of stack elements are the central processing unit are provided on the single integrated circuit in the Accused Microprocessors.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

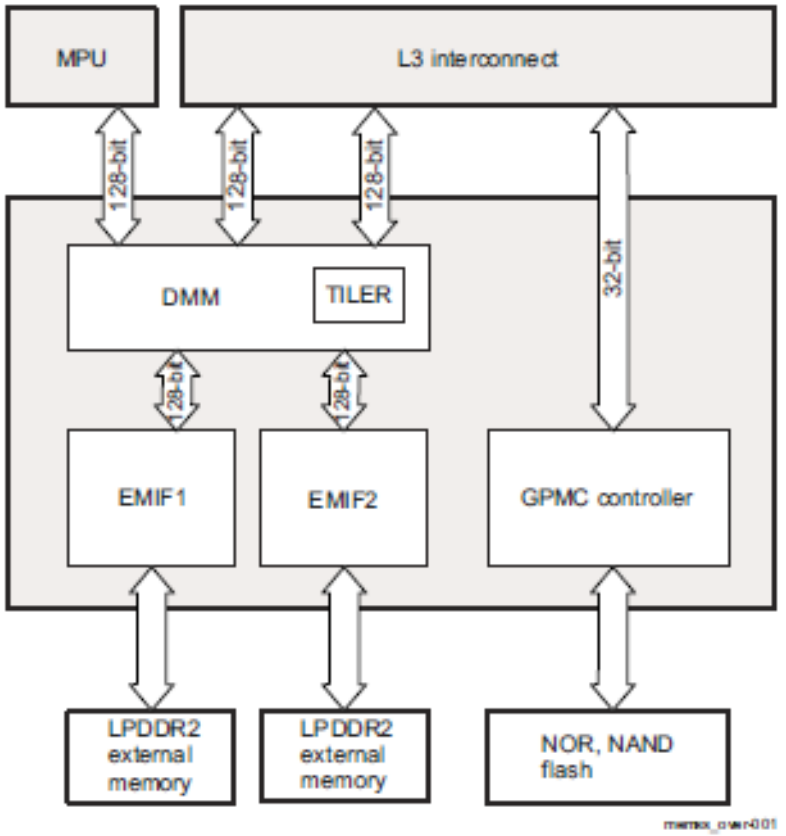
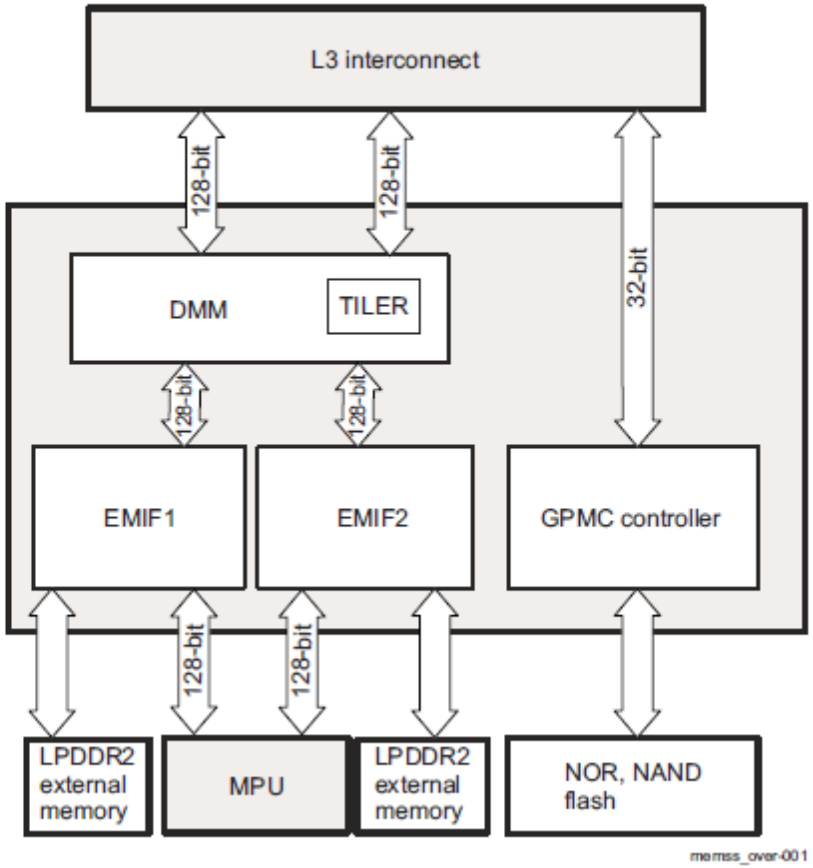
and a third plurality of stack elements configured as a random access memory external to said single integrated circuit.	On information and belief, a third plurality of stack elements is configured as external RAM (*e.g.*, SDRAM). Figure 15-1 shows a functional diagram of all memory subsystems in the device. **Figure 15-1. Memory Subsystem Functional Diagram**  OMAP4430 Technical Reference Manual, p. 3207.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG

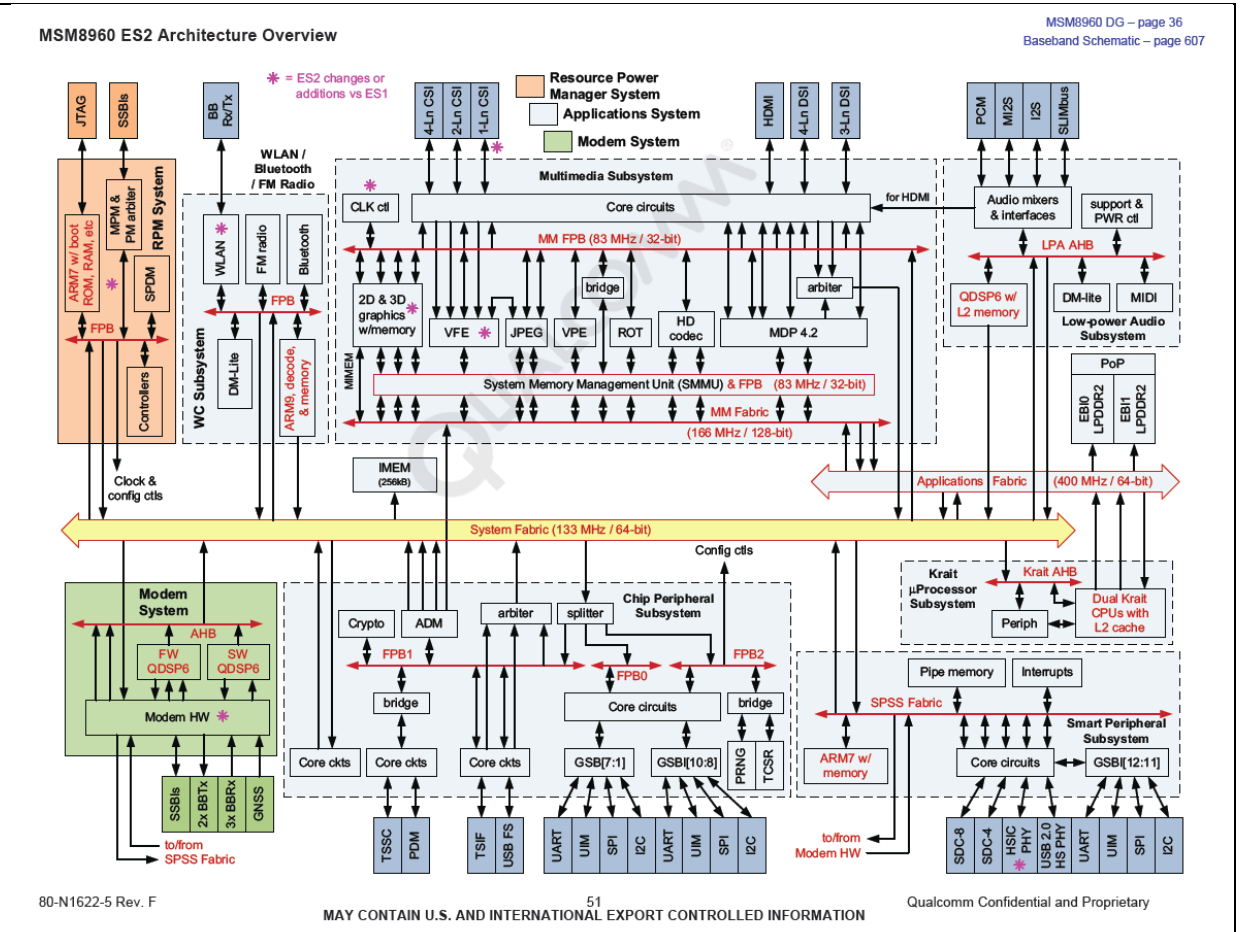
Figure 16-1 shows a functional diagram of all memory subsystems in the device.

Figure 16-1. Memory Subsystem Functional Diagram



OMAP4470 Technical Reference Manual, p. 3302.

EXHIBIT G-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By LG



MSM8960 Chipset Schematics and Design Guidelines, p. 50.

EXHIBIT G-3

DECLARATION OF OLGA I. MAY IN SUPPORT OF DEFENDANTS LG ELECTRONICS, INC. AND LG ELECTRONICS U.S.A., INC.'S MOTION TO STRIKE INFRINGEMENT CONTENTIONS OR, ALTERNATIVELY, COMPEL SUPPLEMENTAL INFRINGEMENT CONTENTIONS ON CHARTED PROCESSORS

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

Claim 1	
[1 preamble[1]] A microprocessor system, comprising	On information and belief, each Accused Product listed in the attached list of Accused Products contains a microprocessor ("Accused Microprocessors"). Each microprocessor is an electronic circuit that interprets and executes programmed instructions. For example, the Optimus L9 (P769) has an OMAP 4430 Processor, and the Escape contains a Qualcomm MSM8960L. Many of the Accused Products contain ARM-based processors or processors that operate in a similar fashion to ARM-based processors. *See* attached list of Accused Products. Thus, infringement is shown below with respect to examples that are representative of the operation of the Accused Products.[2]
[1a] a central processing unit integrated circuit,	On information and belief, each Accused Product contains a **central processing unit** (CPU), which is an electronic circuit on an integrated circuit that controls the interpretation and execution of programmed instructions. Many of the Accused Products contain ARM core CPUs. *See* attached list of Accused Products. For example,

[1] The designations in square brackets before the claim language in each row is added to permit convenient reference to specific claim language. These added designations are not part of the claim language and are not intended to limit the claims in any way. No interpretation is intended to be conveyed by the words grouped together with each designation.

[2] Infringement by the Accused Products is based on the operation of and implementation of the ARM core or a similar core, regardless of the processor manufacturer. *See, e.g.*, http://en.wikipedia.org/wiki/Comparison_of_ARMv7-A_cores and sources cited on that page. This chart provides some examples of such operation that, on information and belief, are representative of the operation of the processors in each of the Accused Products. Discovery is in the early stages, and Plaintiffs anticipate receiving additional documents showing the exact operation of the processor in each of the Accused Products with respect to the accused functionality. But because many processor documents that Plaintiffs would rely on to establish infringement, including those that describe the operation of the processor and its core, are confidential and have not yet been produced in this litigation, Plaintiffs anticipate receiving additional documents to confirm the operational principles shown in this chart from Defendants and/or third parties. Accordingly, Plaintiffs reserve the right to amend, supplement, or augment its claim charts, infringement contentions, or infringement theories based on documents and information later received through discovery.

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

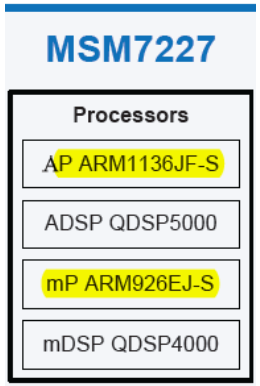
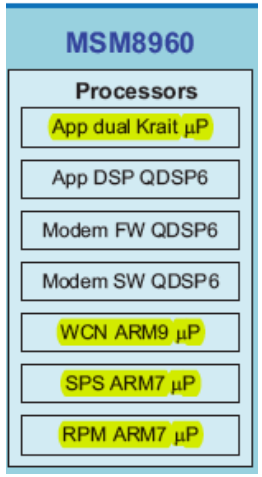
	the Qualcomm MSM7227 microprocessor includes an ARM core CPU. The Qualcomm MSM8960 processor includes ARM cores and a Krait core, which is similar in architecture and design to the ARM cores. *See* http://www.anandtech.com/show/4940/qualcomm-new-snapdragon-s4-msm8960-krait-architecture (Krait similar to ARM core and "capable of decoding any ARMv7-A instructions"). As to Accused Products with Qualcomm processors, *see, e.g.*:   MSM7227 Chipset Training, Introductions and Chipset Overview, 80-VM299-21 Rev. B ("MSM7227 Chipset Training") at 7; MSM8960 Chipset (RTR860x, PM8921, WCD9310, WCN3660) Schematics and Design Guidelines, 80-N1622-5 Rev. F, October 10, 2011 ("MSM8960 Design Guidelines") at 38; *See also* the ARM core documentation cited below. In general, Qualcomm chips have ARM cores, or cores with architecture similar to the ARM architecture that implement the ARM instruction set. *See, e.g.*, MSM8960 Design Guidelines at 88; Qualcomm Technologies, Inc. QMC Chipset Product Roadmaps, December 2012; http://en.wikipedia.org/wiki/Comparison_of_ARMv7-A_cores; http://en.wikipedia.org/wiki/Scorpion_(CPU); http://en.wikipedia.org/wiki/Krait_(CPU); http://www.anandtech.com/show/4940/qualcomm-new-snapdragon-s4-msm8960-krait-architecture; http://www.anandtech.com/show/5559/qualcomm-snapdragon-s4-krait-performance-preview-msm8960-adreno-225-benchmarks/2.

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

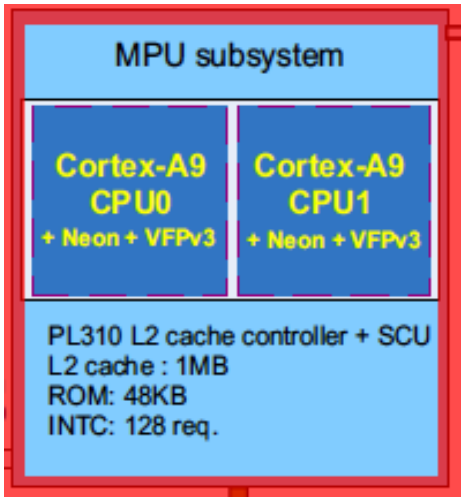
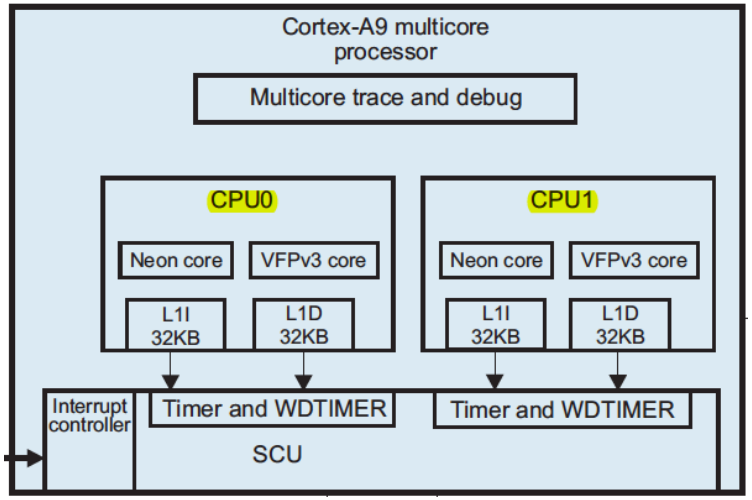
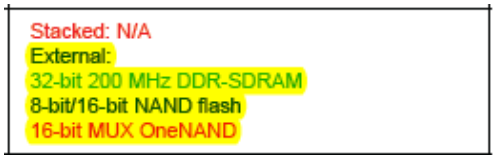
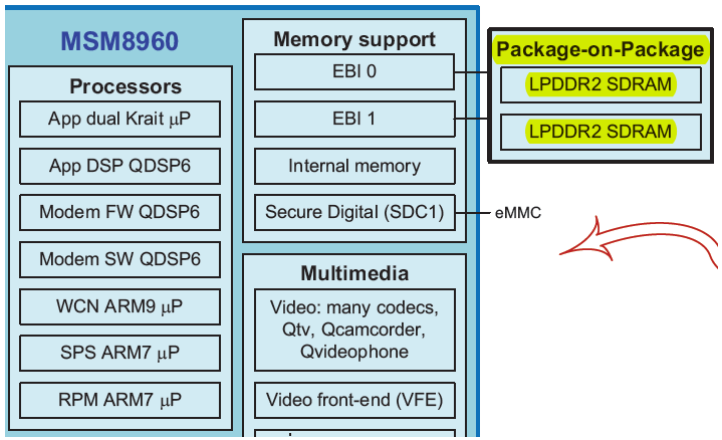
	As to Accused Products with OMAP processors and other ARM processors, *see, e.g.*:   OMAP4430 ES2.x Technical Reference Manual [Public] (Rev. AP) [TPLBN008500-TPLBN014259] ("OMAP4430 TRM") at 275 [TPLBN008774], 1077 [TPLBN009576].
[1b] a memory external of said central processing unit integrated circuit,	On information and belief, each Accused Product has memory external of the central processing unit integrated circuit. As to Accused Products with Qualcomm processors, *see e.g.*:

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG





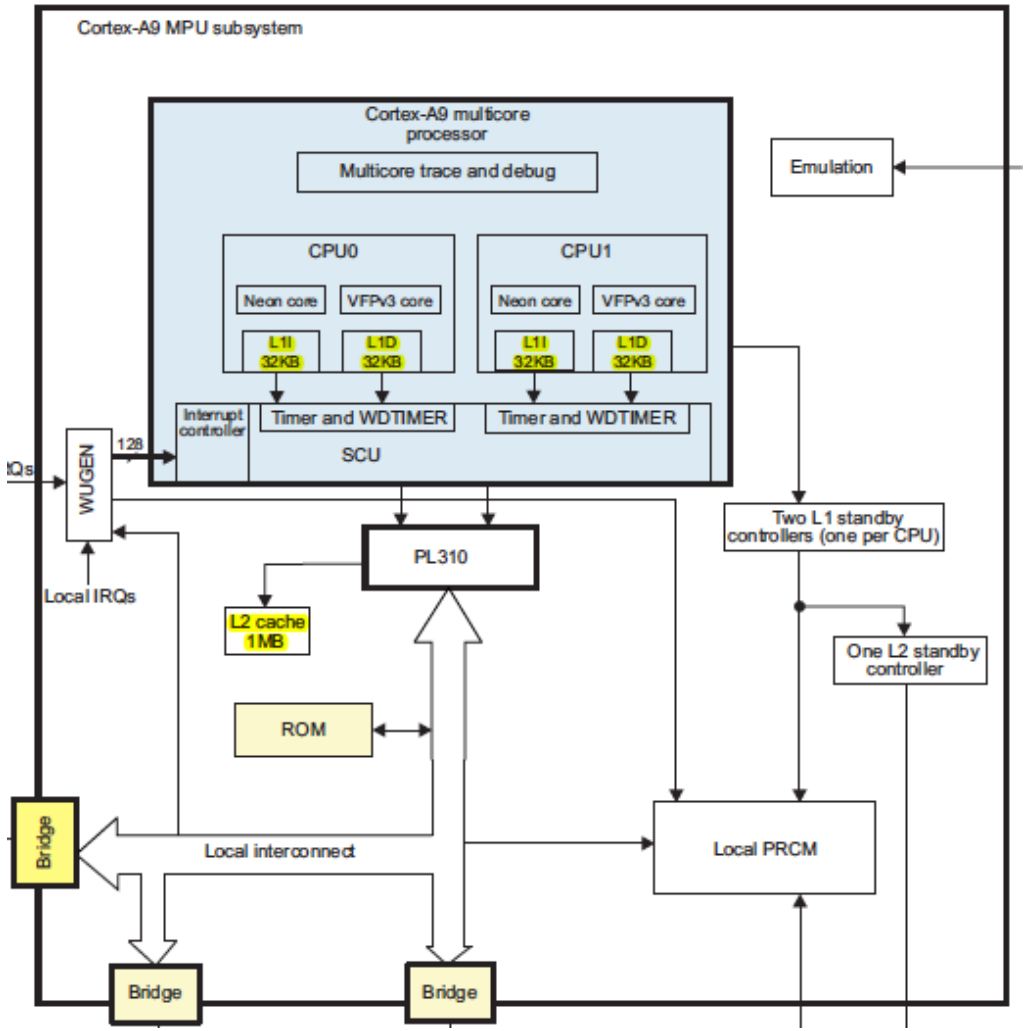
MSM7227 Chipset Training at 9; MSM8960 Design Guidelines at 38. *See also* Cortex-A9 Technical Reference Manual, Revision r2p2 [TPLBN034858-TPLBN035087] ("Cortex-A9 TRM") at 1-2 (19) [TPLBN034876], 1-4 (21) [TPLBN034878], 2-2 (33) [TPLBN034890]; Cortex-A8 Technical Reference Manual, Revision r3p2 [TPLBN034278-TPLBN034857] ("Cortex-A8 TRM"), at 1-2 (27) [TPLBN034304], 1-4 (29) [TPLBN034306]; ARM1176JZF-S Technical Reference Manual, Revision r0p7 [TPLBN035088-TPLBN035846] ("ARM11 TRM"), at 1-8 (34) [TPLBN035121]; and the ARM core documentation cited below.

As to Accused Products with OMAP processors and other ARM processors, *see, e.g.*:

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

	4.1.1 Introduction The dual Cortex™-A9 microprocessor unit (MPU) subsystem of the device is based on the symmetric multiprocessor (SMP) architecture, thus the dual Cortex-A9 MPU subsystem delivers higher performance and optimal power management, debug and emulation capabilities. The dual Cortex-A9 MPU subsystem incorporates two Cortex-A9 central processing units (CPUs), level 2 (L2) cache shared between the two CPUs, and uses PL310 as L2 cache controller. Each CPU has 32KB of level 1 (L1) instruction cache, 32KB of L1 data cache, separate dedicated power domain, and includes one Neon™ and Vector Floating Point Unit (VFPv3) coprocessors. The dual Cortex-A9 MPU subsystem also includes standard CoreSight™ components to support SMP debug and emulation, snoop control unit (SCU), interrupt controller (GIC), and clock and reset manager. The MPU subsystem handles transactions among the ARM® core, the L3 interconnect, the DMM (dynamic memory manager), the L4-ABE, and the interrupt controller (INTC). From this point below, every reference to the dual Cortex-A9 MPU Subsystem, Cortex-A9 MPU subsystem, and MPU subsystem are equivalent. Figure 4-1 shows a high-level block diagram of the MPU subsystem. OMAP4430 TRM at 1076 [TPLBN009575]; *see also* Cortex-A8 Technical Reference Manual, Revision r3p2 [TPLBN034278-TPLBN034857] ("Cortex-A8 TRM"), at 1-2 (27) [TPLBN034304], 1-4 (29) [TPLBN034306].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG



OMAP4430 TRM at 1077 [TPLBN009576] (Fig. 4-1); *see also* Cortex-A9 Technical Reference Manual, Revision r2p2 [TPLBN034858-TPLBN035087] ("Cortex-A9 TRM") at 1-2 (19) [TPLBN034876], 1-4 (21) [TPLBN034878], 2-2 (33) [TPLBN034890]; Cortex-A8 Technical Reference Manual, Revision r3p2 [TPLBN034278-TPLBN034857] ("Cortex-A8 TRM"), at 1-2 (27) [TPLBN034304], 1-4 (29) [TPLBN034306].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

	1.3.1 Cortex-A9 MPU Subsystem Description The Cortex-A9 MPU subsystem integrates the following submodules: • ARM Cortex-A9 MPCore – Two ARM Cortex-A9 central processing units (CPUs) – ARM Version 7 ISA™: Standard ARM instruction set plus Thumb®-2, Jazelle® RCT and Jazelle DBX Java™ accelerators – Neon™ SIMD coprocessor and VFPv3 per CPU – Interrupt controller (Cortex-A9 MPU INTC) with up to 128 interrupt requests – One general-purpose timer and one watchdog timer per CPU – Debug and trace features – 32-KB instruction and 32-KB data level 1 (L1) caches per CPU • Shared 1-MB level 2 (L2) cache • 48 KB bootable ROM • Local power, reset, and clock managment (PRCM) module • Emulation features • Digital phase-locked loop (DPLL) OMAP4430 TRM at 276 [TPLBN008775].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

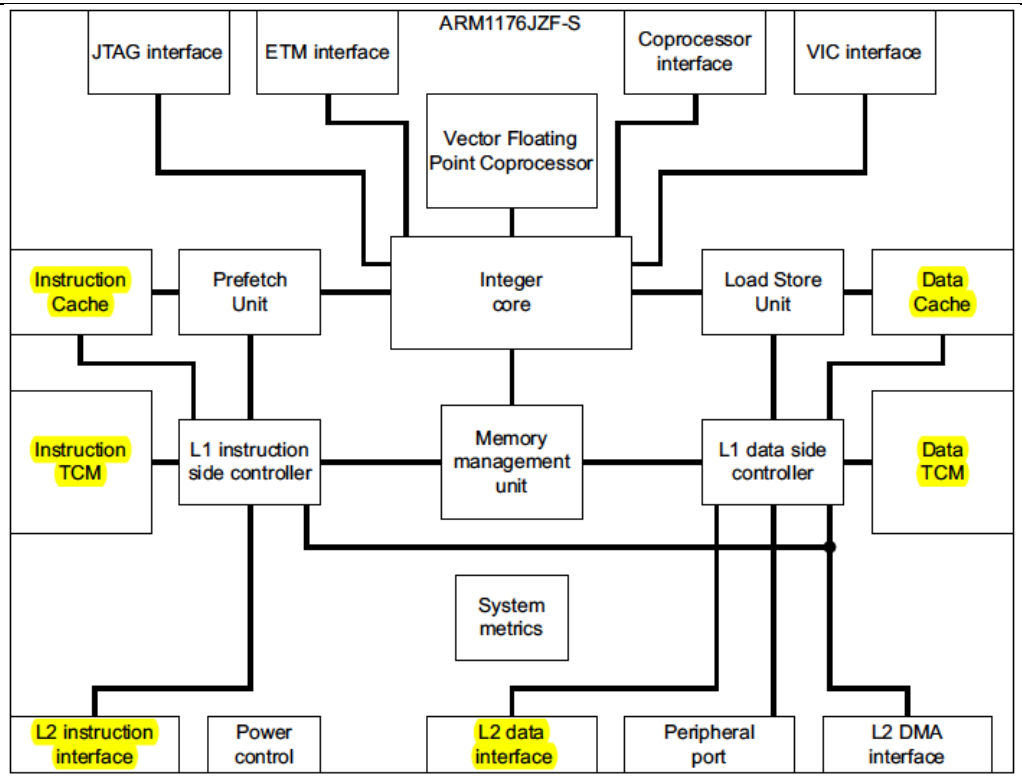
	 **Figure 1-1 ARM1176JZF-S processor block diagram** ARM1176JZF-S Technical Reference Manual, Revision r0p7 [TPLBN035088-TPLBN035846] ("ARM11 TRM"), at 1-8 (34) [TPLBN035121].
[1c] a bus connecting said central processing unit integrated circuit to said memory,	On information and belief, each Accused Product has a bus connecting the central processing unit integrated circuit to said memory.

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

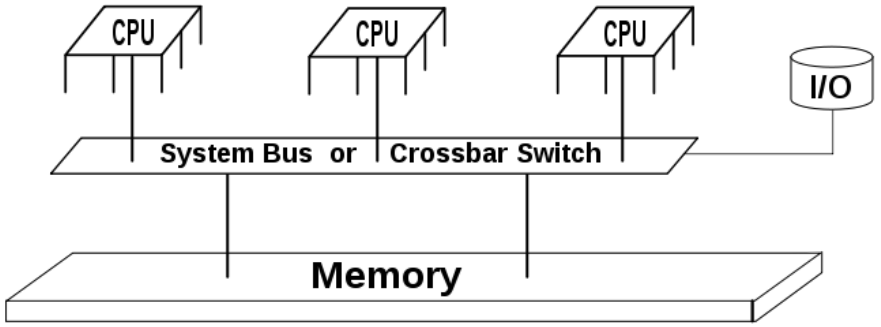
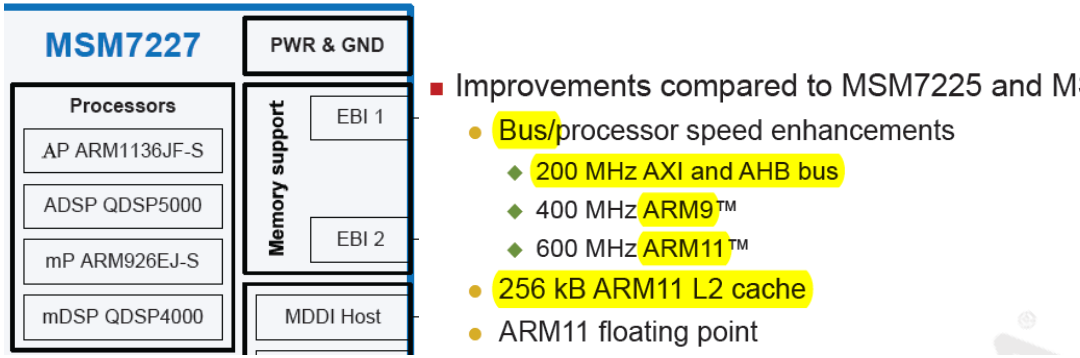
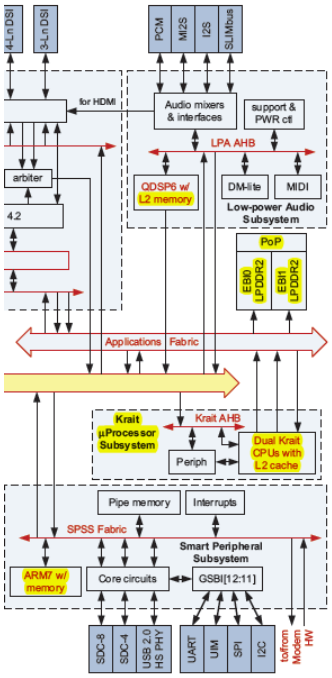
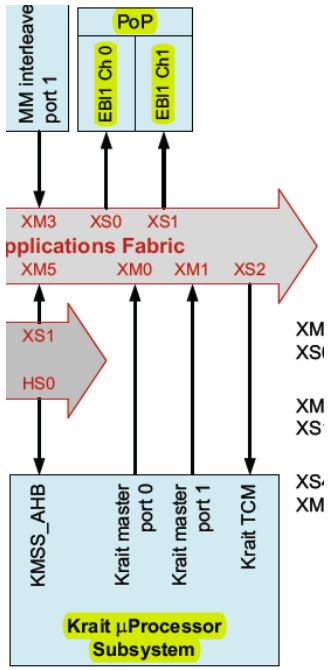
and	 As to Accused Products with Qualcomm processors, *see e.g.*:  MSM7227 Chipset Training at 5, 7. *See also* the ARM core documentation cited below.

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG





Major MSM8960 functional blocks (left) are shared by its internal systems, subsystem, and busses – the 'architecture' (right)

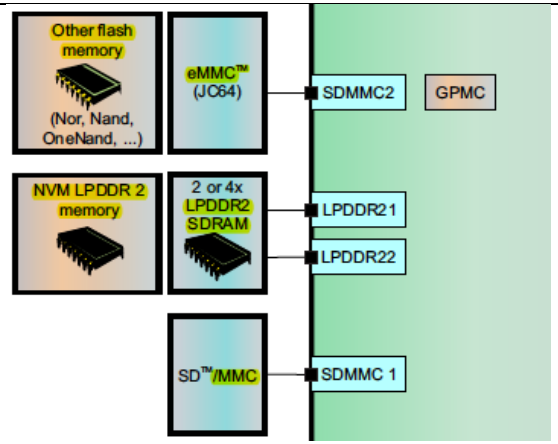
Key DDR memory controller features

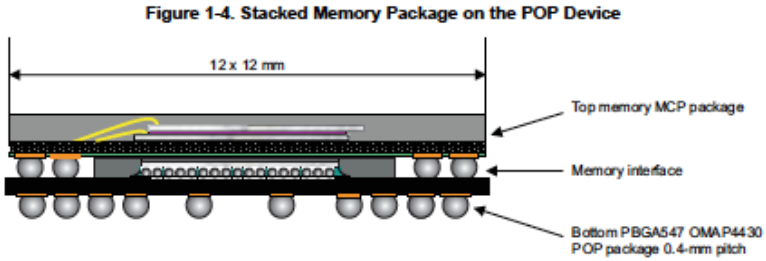
- Supports DDR on address bus
- Two major clock domains (AXI slave and DDR controller)
- Multiple AXI-to-DDR clock modes
  – Synchronous (1:1), iso-synchronous (1:2), asynchronous
- Highly flexible with many configurable parameters, including interface timing
- Flexible memory page management with various page open/close policies
- Out-of-order command execution and read data return
- Multiple high-priority tiers for sophisticated handling of priority requests

MSM8960 Design Guidelines at 38, 44, 94. *See also* the ARM core documentation cited below.

As to Accused Products with OMAP processors and other ARM processors, *see, e.g.*:

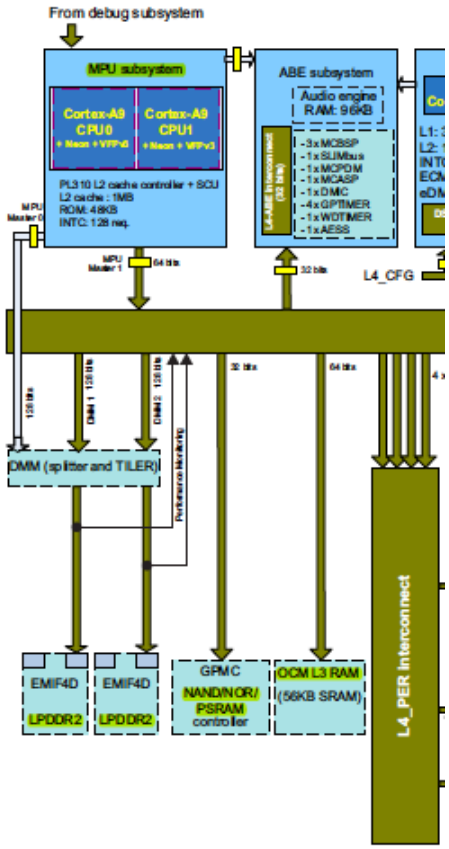
EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG





OMAP4430 TRM at 273 [TPLBN008772], 285 [TPLBN008784].

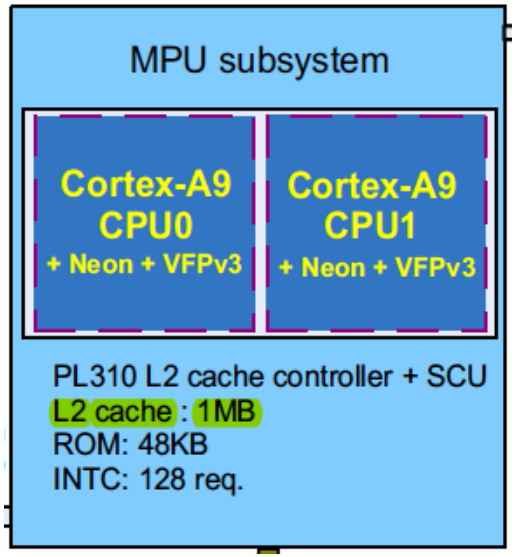
EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

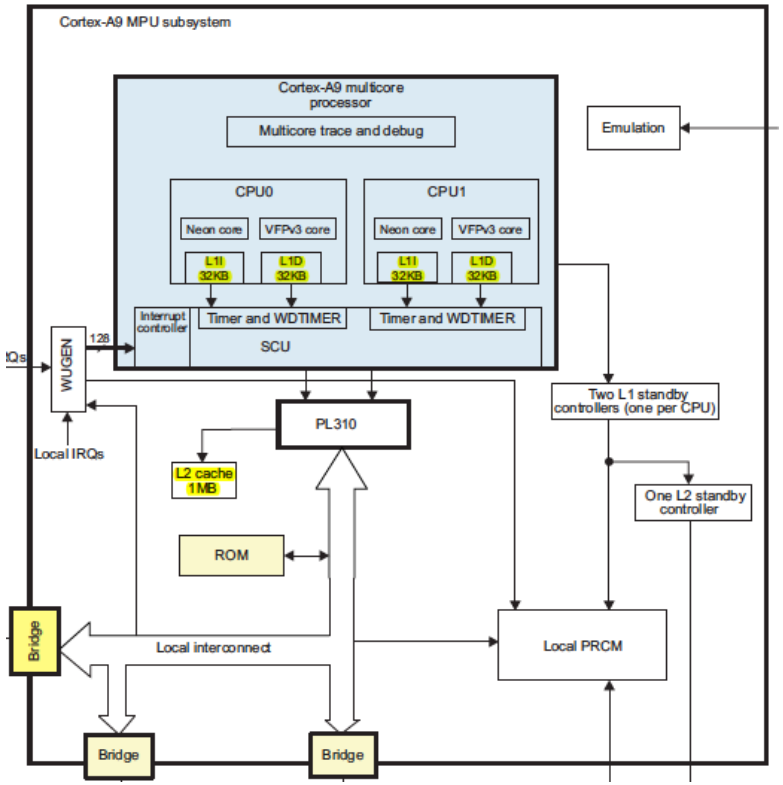


- The device also integrates:
  - On-chip memory
  - External memory interfaces
  - Memory management
  - Level 3 (L3) and level 4 (L4) interconnects
  - System and connecting peripherals

OMAP4430 TRM at 275 [TPLBN008774], 272 [TPLBN008771] ; *see also* Cortex-A9 TRM, at 1-2 (19) [TPLBN034876], 1-4 (21) [TPLBN034878], 2-2 (33) [TPLBN034890]; Cortex-A8 TRM, at 1-2 (27) [TPLBN034304], 1-4 (29) [TPLBN034306].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG





OMAP4430 TRM at 275 [TPLBN008774] (Fig. 1-2), 1088 [TPLBN022288] (Fig. 4-1); *see also* Cortex-A9 TRM, at 1-2 (19) [TPLBN034876], 1-4 (21) [TPLBN034878], 2-2 (33) [TPLBN034890].

2.2 Interfaces

The processor has the following external interfaces:

- AMBA AXI interfaces
- APB CoreSight interface
- DFT interface.

See the *AMBA AXI Protocol Specification*, the *CoreSight Architecture Specification*, the *CoreSight PFT Architecture Specification*, and the *Cortex-A9 MBIST Controller Technical Reference Manual* for more information on these interfaces.

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

1.4 Features

The Cortex-A9 processor includes the following features :

- superscalar, variable length, out-of-order pipeline with dynamic branch prediction
- full implementation of the ARM architecture v7-A instruction set
- Security Extensions
- Harvard level 1 memory system with *Memory Management Unit* (MMU).
- two 64-bit AXI master interfaces with Master 0 for the data side bus and Master 1 for the instruction side bus
- ARMv7 Debug architecture
- support for trace with the *Program Trace Macrocell* (PTM) interface
- support for advanced power management with up to 3 power domains
- optional Preload Engine
- optional Jazelle hardware acceleration
- optional Data Engine with MPE and VFPv3.

Cortex-A9 TRM, at 1-2 (19) [TPLBN034876], 1-4 (21) [TPLBN034878], 1-6 (23) [TPLBN034880], 2-2 (33) [TPLBN034890], 2-4 (35) [TPLBN034892], Glossary-2 (216) [TPLBN035073].

1.4.1 AMBA AXI interface

The AXI bus interface is the main interface to the system bus. It performs L2 cache fills and noncacheable accesses for both instructions and data. The AXI interface supports 64-bit or 128-bit wide input and output data buses. It also supports multiple outstanding requests on the AXI bus. The AXI signals are synchronous to the **CLK** input. A wide range of bus clock to core clock ratios is possible through the use of the AXI clock enable signal **ACLKEN**. See the *AMBA AXI Protocol Specification* for more information.

Cortex-A8 TRM, at 1-7 (32) [TPLBN034309], *see also* Glossary-2 (564) [TPLBN034841], 1-2 (27) [TPLBN034304], 1-4 (29) [TPLBN034306].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

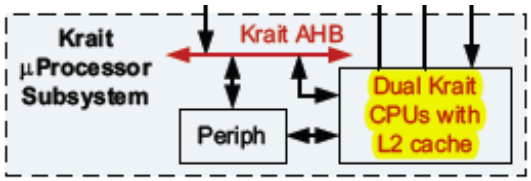
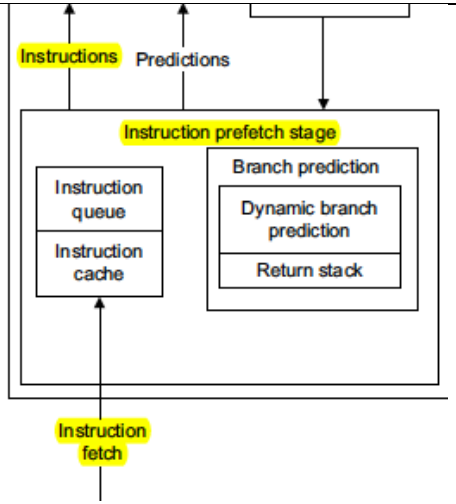
	1.5.5 AMBA AXI interface The bus interface provides high bandwidth connections between the processor, second level caches, on-chip RAM, peripherals, and interfaces to external memory. There are separate bus interfaces for: • instruction fetch, 64-bit data • data read/write, 64-bit data • peripheral access, 32-bit data • DMA, 64-bit data. ARM11 TRM, at 1-16 (42) [TPLBN035129], *see also* 1-8 (34) [TPLBN035121] (Fig. 1-1).
[1d] means connected to said bus for fetching instructions for said central processing unit integrated circuit on said bus from said memory,	On information and belief, each Accused Product has means connected to said bus for fetching instructions for said central processing unit integrated circuit on said bus from said memory. As to Accused Products with Qualcomm processors, *see e.g.*: ▪ Improvements compared to MSM7225 and MSM7200A devices • Bus/processor speed enhancements ◆ 200 MHz AXI and AHB bus ◆ 400 MHz ARM9™ ◆ 600 MHz ARM11™ • 256 kB ARM11 L2 cache • ARM11 floating point  MSM7227 Chipset Training at 5; MSM8960 Design Guidelines at 51,86, 88, 454; http://www.anandtech.com/show/4940/qualcomm-new-snapdragon-s4-msm8960-krait-architecture (showing fetch and decode stages for the Krait core; "The architecture can fetch and decode three instructions per clock. The decoders are equally capable of decoding any ARMv7-A instructions."). *See also* the ARM core documentation cited below. As to Accused Products with OMAP processors and other ARM processors, *see, e.g.*:

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

	1.3.1 Cortex-A9 MPU Subsystem Description The Cortex-A9 MPU subsystem integrates the following submodules: • ARM Cortex-A9 MPCore – Two ARM Cortex-A9 central processing units (CPUs) – ARM Version 7 ISA™: Standard ARM instruction set plus Thumb®-2, Jazelle® RCT and Jazelle DBX Java™ accelerators – Neon™ SIMD coprocessor and VFPv3 per CPU – Interrupt controller (Cortex-A9 MPU INTC) with up to 128 interrupt requests – One general-purpose timer and one watchdog timer per CPU – Debug and trace features – 32-KB instruction and 32-KB data level 1 (L1) caches per CPU • Shared 1-MB level 2 (L2) cache • 48 KB bootable ROM • Local power, reset, and clock managment (PRCM) module • Emulation features • Digital phase-locked loop (DPLL) OMAP4430 TRM at 276 [TPLBN008775]; *see also* ARMv7-A Architecture Reference Manual [TPLBN049289-TPLBN051446] ("ARMv7 Reference Manual") at A3-3 (117) [TPLBN049405] re: "prefetch[ing] instructions," A3-28 (142) [TPLBN049430] re: "instruction fetches."; Cortex-A8 TRM at 1-4 (29) [TPLBN034306]; Cortex-A8 TRM at 1-5 (30) [TPLBN034307].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG



Cortex-A9 TRM at 2-2 (33) [TPLBN034890].

Memory coherency	A memory is coherent if the value read by a data read or instruction fetch is the value that was most recently written to that location. Memory coherency is made difficult when there are multiple possible physical locations that are involved, such as a system that has main memory, a write buffer and a cache.

Cortex-A9 TRM at Glossary-11 (225) [TPLBN035082].

Prefetching	In pipelined processors, the process of fetching instructions from memory to fill up the pipeline before the preceding instructions have finished executing. Prefetching an instruction does not mean that the instruction must be executed.

Cortex-A9 TRM at Glossary-12 (226) [TPLBN035083].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

A3.9.2 Memory hierarchy

Memory close to a processor has very low latency, but is limited in size and expensive to implement. Further from the processor it is easier to implement larger blocks of memory but these have increased latency. To optimize overall performance, an ARMv7 memory system can include multiple levels of cache in a hierarchical memory system. Figure A3-5 shows such a system, in an ARMv7-A implementation of a VMSA, supporting virtual addressing.

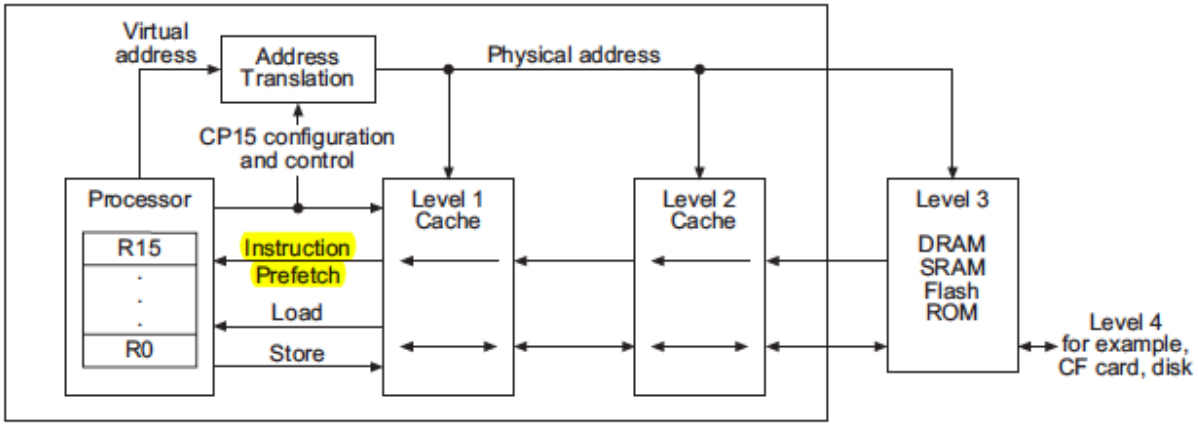


Figure A3-5 Multiple levels of cache in a memory hierarchy

ARMv7 Reference Manual at A3-52 (166) [TPLBN049454] *see also* A3-53 – A3-54 (167-68) [TPLBN049455-56].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

For implicit accesses:

- Cache linefills and evictions have no effect on the single-copy atomicity of explicit transactions or instruction fetches.
- Instruction fetches are single-copy atomic for each instruction fetched.

  Note

  32-bit Thumb instructions are fetched as two 16-bit items.

- Translation table walks are performed as 32-bit accesses aligned to 32 bits, each of which is single-copy atomic.

ARMv7 Reference Manual at A3-27 – 3-28 (141-142) [TPLBN049429-30].

A3.5.4 Normal memory

Normal memory is idempotent, meaning that it exhibits the following properties:

- read accesses can be repeated with no side effects
- repeated read accesses return the last value written to the resource being read
- read accesses can prefetch additional memory locations with no side effects
- write accesses can be repeated with no side effects, provided that the contents of the location are unchanged between the repeated writes
- unaligned accesses can be supported
- accesses can be merged before accessing the target memory system.

ARMv7 Reference Manual at A3-28 – A3-29 (142-143) [TPLBN049430-31].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

A2.2 ARM core data types and arithmetic

All ARMv7-A and ARMv7-R processors support the following data types in memory:

Byte 8 bits
Halfword 16 bits
Word 32 bits
Doubleword 64 bits.

Processor registers are 32 bits in size. The instruction set contains instructions supporting the following data types held in registers:

- 32-bit pointers
- unsigned or signed 32-bit integers
- unsigned 16-bit or 8-bit integers, held in zero-extended form
- signed 16-bit or 8-bit integers, held in sign-extended form
- two 16-bit integers packed into a register
- four 8-bit integers packed into a register
- unsigned or signed 64-bit integers held in two registers.

ARMv7 Reference Manual at A2-3 (35) [TPLBN049323].

The instructions that operate on packed halfwords or bytes include some multiply instructions that use just one of two halfwords, and *Single Instruction Multiple Data* (SIMD) instructions that operate on all of the halfwords or bytes in parallel.

Direct instruction support for 64-bit integers is limited, and most 64-bit operations require sequences of two or more instructions to synthesize them.

ARMv7 Reference Manual at A2-3 (35) [TPLBN049323].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

A3.1.1 Address incrementing and address space overflow

When a processor performs normal sequential execution of instructions, it effectively calculates:

```
(address_of_current_instruction) + (size_of_executed_instruction)
```

after each instruction to determine which instruction to execute next.

Note

The size of the executed instruction depends on the current instruction set, and might depend on the instruction executed.

ARMv7 Reference Manual at A3-2 (116) [TPLBN049404].

Table 4-3. ARM Core Key Features (continued)

Feature	Comment
L2 cache (PL310)	Main characteristics are: • Physically addressed and physically tagged • 16-way associative • 32-byte line length • Critical word first transactions • Prefetching capability

OMAP4430 TRM at 1086 [TPLBN009585].

2.2 L3 Memory Space Mapping

The memory space system is hierarchical: level 1 (L1), level 2 (L2), L3, and L4. L1 and L2 are memories in the Cortex-A9 MPU, Cortex™-M3 MPU, and the digital signal processor (DSP) subsystems. L3 handles many types of data transfers, including data exchange with system on-chip/external memories. The chip-level interconnect, which consists of one L3 and four L4s, enables communication among all modules and subsystems.

OMAP4430 TRM at 292 [TPLBN008791].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

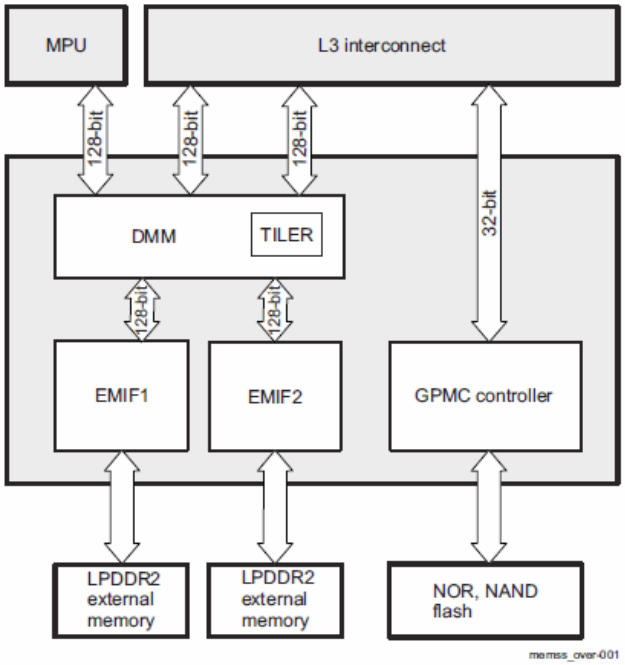
15.1.4 GPMC Overview

The general-purpose memory controller (GPMC) is an unified memory controller dedicated to interfacing external memory devices:

- Asynchronous SRAM-like memories and application-specific integrated circuit (ASIC) devices
- Asynchronous, synchronous, and page mode (available only in nonmuxed mode) burst NOR flash devices
- NAND flash
- Pseudo-SRAM devices

OMAP4430 TRM at 3209 [TPLBN011708].

Figure 15-1. Memory Subsystem Functional Diagram



OMAP4430 TRM at 3207 [TPLBN011706].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

For ARMv6-M, instruction fetches are always halfword-aligned and data accesses are always naturally aligned.

Address calculations are normally performed using ordinary integer instructions. This means that they wrap around if they overflow or underflow the address space. Another way of describing this is that any address calculation is reduced modulo 2^{32}.

Normal sequential execution of instructions effectively calculates:

```
(address_of_current_instruction) + (size_of_executed_instruction)
```

after each instruction to determine the instruction to execute next. If this calculation overflows the top of the address space, the result is UNPREDICTABLE. In ARMv6-M this condition cannot occur because the top of memory is defined to always have the *eXecute Never* (XN) memory attribute associated with it. See *The system address map* on page B3-258 for more information. An access violation is reported if this scenario occurs.

The information in this section only applies to instructions that are executed, including those that fail their condition code check. Most ARM implementations prefetch instructions ahead of the currently-executing instruction.

ARMv6-M Architecture Reference Manual [TPLBN035847-TPLBN036282] ("ARMv6 Reference Manual") at A3-42 (42) [TPLBN035888].

A3.5.2 Normal memory

Normal memory is idempotent, meaning that it exhibits the following properties:

- read transactions can be repeated with no side effects
- repeated read transactions return the last value written to the resource being read
- read transactions can prefetch additional memory locations with no side effects
- write transactions can be repeated with no side effects, provided that the location is unchanged between the repeated writes
- unaligned accesses are supported
- transactions can be merged prior to accessing the target memory system.

Normal memory can be read and write, or read-only. The Normal memory attribute is further defined as being Shareable or Non-Shareable, and describes most memory used in a system.

ARMv6 Reference Manual at A3-50 – 51 (50-51) [TPLBN035896-97].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

[1e] 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,	On information and belief, in each Accused Product, the means for fetching instructions is 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. As to Accused Products with Qualcomm processors, *see e.g.*: ■ Improvements compared to MSM7225 and MSM7200A devices ● Bus/processor speed enhancements ◆ 200 MHz AXI and AHB bus ◆ 400 MHz ARM9™ ◆ 600 MHz ARM11™ ● 256 kB ARM11 L2 cache ● ARM11 floating point Dual Krait µP CPUs, each with: • Up to 1.5 GHz • 1 MB L2 cache • 32 kB L1 instruction and data caches • ARM v7 compliant • TrustZone support • VeNum 128-bit SIMD MM coprocessor MSM7227 Chipset Training at 5; MSM8960 Design Guidelines at 86, 88, 454; http://www.anandtech.com/show/4940/qualcomm-new-snapdragon-s4-msm8960-krait-architecture (showing fetch and decode stages for the Krait core; "The architecture can fetch and decode three instructions per clock. The decoders are equally capable of decoding any ARMv7-A instructions."). *See also* the ARM core documentation cited below. As to Accused Products with OMAP processors and other ARM processors, *see, e.g.*: ***1.3.12 On-Chip Memory Description*** The on-chip memory is divided into L3 OCM RAM, SAR ROM, SAR RAM, and memories in the subsystems (Cortex-A9, DSP, Cortex-M3, ABE, and IVA-HD). • The L3 OCM RAM consists of 56KB of on-chip SRAM. • The save-and-restore (SAR) ROM consists of 4KB and contains a linked list of descriptors used by the system DMA (sDMA). • The SAR RAM consists of 8KB divided into four blocks. It is used as context-saving memory when the device goes into off mode. OMAP4430 TRM at 283 [TPLBN008782].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

	1.3.1 Cortex-A9 MPU Subsystem Description The Cortex-A9 MPU subsystem integrates the following submodules: • ARM Cortex-A9 MPCore – Two ARM Cortex-A9 central processing units (CPUs) – ARM Version 7 ISA™: Standard ARM instruction set plus Thumb®-2, Jazelle® RCT and Jazelle DBX Java™ accelerators – Neon™ SIMD coprocessor and VFPv3 per CPU – Interrupt controller (Cortex-A9 MPU INTC) with up to 128 interrupt requests – One general-purpose timer and one watchdog timer per CPU – Debug and trace features – 32-KB instruction and 32-KB data level 1 (L1) caches per CPU • Shared 1-MB level 2 (L2) cache • 48 KB bootable ROM • Local power, reset, and clock managment (PRCM) module • Emulation features • Digital phase-locked loop (DPLL) OMAP4430 TRM at 276 [TPLBN008775], 1078 [TPLBN009577], 1084 [TPLBN009583]. • PL310 L2 cache controller (revision r2p0) with 1MB cache and two 64-bit slave and two 64-bit master ports (For details about L2 cache controller, see the ARM PL310 Cache Controller TRM, available at infocenter.arm.com/help/index.jsp). • Local interconnect: Connects the ARM Cortex-A9 multicore processor to the level 3 (L3) interconnect, dynamic memory manager, ABE interconnect, local PRCM, PL310 L2 cache controller, on-chip ROM memory, and the WUGEN. OMAP4430 TRM at 1084 [TPLBN009583]. • PL310 L2 cache controller (revision r2p0) with 1MB cache size – 16-way set associative – 32-byte line size – Two slave ports and two master ports – Includes four 256-bit line-fill-buffers (LFBs) shared by the master ports – Each slave port includes two 256-bit line-read-buffers (LRBs) – Includes four 256-bit store buffers with merge capability

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

OMAP4430 TRM at 1084 [TPLBN009583].

- Caches memories:
  - 32-KB L1 instruction and 32-KB data caches – 4-way associative on each core
  - 1-MB L2 unified cache – 16-way associative

OMAP4430 TRM at 1084 [TPLBN009583].

4.3.1 Cortex-A9 MPU Subsystem Block Diagram

The Cortex-A9 MPU subsystem integrates the following group of submodules:

- Two ARM Cortex-A9 CPUs. Each CPU contains:
  - ARM version 7 ISA™: Standard ARM instruction set plus Thumb®-2 , Jazelle® RCT and Jazelle DBX Java™ accelerator
  - Neon SIMD coprocessor and VFPv3
- INTC: Handles module interrupts (For details, see Chapter 17, *Interrupt Controller.*)
- PL310 L2 cache controller (revision r2p0) with 1MB cache and two 64-bit slave and two 64-bit master ports (For details about L2 cache controller, see the ARM PL310 Cache Controller TRM, available at infocenter.arm.com/help/index.jsp).

OMAP4430 TRM at 1084 [TPLBN009583].

4.3.3 Local Interconnect

The local interconnect is used in the device design to connect the two 64-bit buses of the PL310 L2 cache controller to the L3 interconnect (64-bit width), EMIF interconnect port (128-bit width), ABE interconnect port (32-bit width), and local power manager. The local interconnect must do some minimal address decoding to decide where to forward the requests.

Main features:

- Connects to the EMIF through a 128-bit interconnect port
- Connects to the L3 interconnect through a 64-bit port
- Connects to the ABE through a 32-bit port
- Supports Single-Request-Multiple-Data (data handshaking) burst mode to pipeline requests
- Supports multiple outstanding requests
- Supports posted and nonposted write transactions based on the attributes of the transactions coming from the ARM Cortex-A9 processor; this is hardcoded and is not software-configurable.

OMAP4430 TRM at 1086 [TPLBN009585].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

Neon	Includes advanced SIMD instructions and the ARM VFPv3 instructions

OMAP4430 TRM at 1086 [TPLBN009584].

Table 4-3. ARM Core Key Features

Feature	Comment
ARM version 7 ISA	Standard ARM instruction set + Thumb-2, Jazelle RCT/DBX Java accelerator, and media extensions. Backward-compatible with previous ARM ISA versions.
L1 Icache and Dcache	32KB each
L2 cache (PL310)	Main characteristics are: • Physically addressed and physically tagged • 16-way associative • 32-byte line length • Critical word first transactions • Prefetching capability • Pseudo-random victim selection policy • Two 256-bit LFBs in each master port • Two 256-bit LRBs in each slave port • Three 256-bit write buffers (merging capable) • Three 256-bit eviction buffers • Two 64-bit slave ports from SCU • Two 64-bit master ports, one to L3 and one to EMIF
Neon	Includes advanced SIMD instructions and the ARM VFPv3 instructions

OMAP4430 TRM at 1085-86 [TPLBN009584-85].

1.1 About the Cortex-A9 processor

The Cortex-A9 processor is a high-performance, low-power, ARM macrocell with an L1 cache subsystem that provides full virtual memory capabilities. The Cortex-A9 processor implements the ARMv7-A architecture and runs 32-bit ARM instructions, 16-bit and 32-bit Thumb instructions, and 8-bit Java™ bytecodes in Jazelle state.

Figure 1-1 shows a Cortex-A9 uniprocessor in a design with a PL390 Interrupt Controller and an L2C-310 L2 Cache Controller.

Cortex-A9 TRM at 1-2 (19) [TPLBN034876], *see also* 1-6 – 1-7 (23-24) [TPLBN034880-81], 3-2 (48) [TPLBN034905]; *see also* Cortex-A8 TRM at 2-3 (43) [TPLBN034320].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

7.3 About the L1 instruction side memory system

The L1 instruction side memory system is responsible for providing an instruction stream to the Cortex-A9 processor. To increase overall performance and to reduce power consumption, it contains the following functionality:

- dynamic branch prediction
- instruction caching.

Cortex-A9 TRM at 7-5 (119) [TPLBN034976].

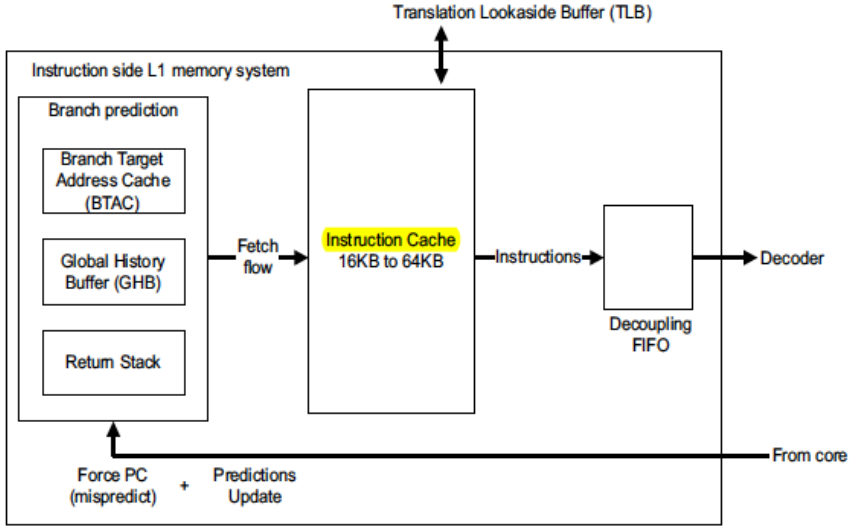


Figure 7-1 Branch prediction and instruction cache

Cortex-A9 TRM at 7-5 (119) [TPLBN034976].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

The Prefetch Unit (PFU)

The Prefetch Unit implements a two-level prediction mechanism, comprising:

- a two-way BTAC of 512 entries organized as two-way x 256 entries implemented in RAMs.
- a *Global History Buffer* (GHB) containing 4096 2-bit predictors implemented in RAMs
- a return stack with eight 32-bit entries.

The prediction scheme is available in ARM state, Thumb state, ThumbEE state, and Jazelle state. It is also capable of predicting state changes from ARM to Thumb, and from Thumb to ARM. It does not predict any other state changes. Nor does it predict any instruction that changes the mode of the core. See *Program flow prediction* on page 7-6.

Cortex-A9 TRM at 7-5 (119) [TPLBN034976].

Instruction Cache Controller

The instruction cache controller fetches the instructions from memory depending on the program flow predicted by the prefetch unit.

The instruction cache is 4-way set associative. It comprises the following features:

- configurable sizes of 16KB, 32KB, or 64KB
- *Virtually Indexed Physically Tagged* (VIPT)
- 64-bit native accesses so as to provide up to four instructions per cycle to the prefetch unit
- security extensions support
- no lockdown support.

Cortex-A9 TRM at 7-5 – 7-6 (119-20) [TPLBN034976-77].

WPTT32LINK	**0**	Indicates the size of the last executed address when in Thumb state: 0 = 16-bit instruction 1 = 32-bit instruction.

Cortex-A9 TRM at A-25 (196) [TPLBN035053].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

A3.1.1 Address incrementing and address space overflow

When a processor performs normal sequential execution of instructions, it effectively calculates:

```
(address_of_current_instruction) + (size_of_executed_instruction)
```

after each instruction to determine which instruction to execute next.

Note

The size of the executed instruction depends on the current instruction set, and might depend on the instruction executed.

ARMv7 Reference Manual at A3-2 (116) [TPLBN049404].

A3.9.1 Introduction to caches

A cache is a block of high-speed memory that contains a number of entries, each consisting of:

- main memory address information, commonly known as a *tag*
- the associated data.

Caches are used to increase the average speed of a memory access. Cache operation takes account of two principles of locality:

Spatial locality

An access to one location is likely to be followed by accesses to adjacent locations. Examples of this principle are:

- sequential instruction execution
- accessing a data structure.

ARMv7 Reference Manual at A3-51 (165) [TPLBN049453].

A6.1 Thumb instruction set encoding

The Thumb instruction stream is a sequence of halfword-aligned halfwords. Each Thumb instruction is either a single 16-bit halfword in that stream, or a 32-bit instruction consisting of two consecutive halfwords in that stream.

ARMv7 Reference Manual at A6-2 (240) [TPLBN049528].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

Execution stream

The stream of instructions that would have been executed by sequential execution of the program.

ARMv7 Reference Manual at Glossary-5 (2149) [TPLBN051437].

15.1.6 OCM Overview

The on-chip memory (OCM) subsystem consists of the following OCM controllers: one connected to an on-chip ROM (SAR ROM), one connected to an on-chip RAM (SAR RAM), and one connected to an on-chip SRAM (L3 SRAM). Each memory controller has its own dedicated interface to the L3 interconnect.

OMAP4430 TRM at 3211 [TPLBN011710].

15.1.6.1 SAR ROM

This on-chip ROM contains 4KB of memory and a linked list of descriptors used by sDMA during the restore context operation (when the device transitions from off to on mode).

The device-embedded SAR ROM has the following characteristics:

- 4-KB ROM
- 32-bit access per cycle
- Support for single- and burst-access transactions

OMAP4430 TRM at 3211 [TPLBN011710].

15.1.6.2 SAR RAM

The on-chip SAR RAM contains 8K bytes and is mapped as 4 blocks with irregular region sizes. This memory content is preserved when the device goes into off mode (as long as the wake-up voltage domain remains supplied). It is used as context-saving memory to be written by software so that sDMA restores its saved content when the device transitions from off to on mode.

The device-embedded SAR RAM has the following characteristics:

- Support for single-access transactions
  - Operates at full L4-PER interconnect clock frequency
  - 32-bit access per cycle

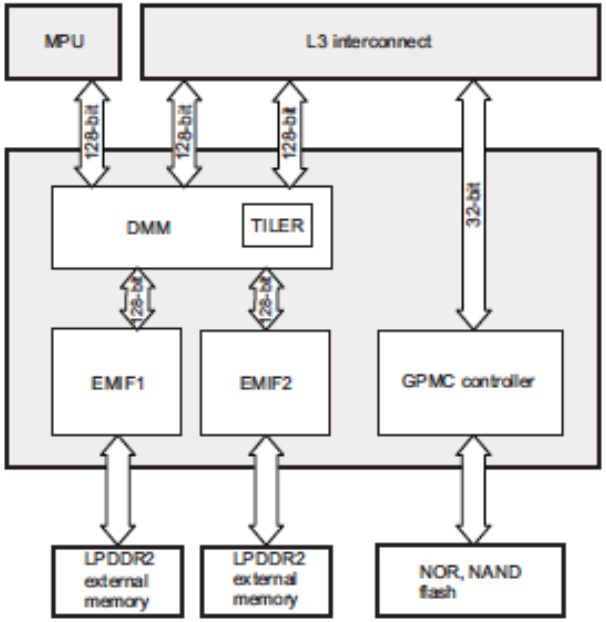
OMAP4430 TRM at 3211 [TPLBN011710].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

	15.1.6.3 L3 OCM_RAM The on-chip L3 OCM_RAM contains 56KB of RAM, and partitioning is defined by the L3 firewall logic. The device-embedded L3 OCM_RAM has the following characteristics: • Support for single and burst access transactions: – Operates at full L3 interconnect clock frequency – Fully pipelined, one 32-bit access per cycle • Restricted access support OMAP4430 TRM at 3211 [TPLBN011710]. ***1.3.14 External Memory Interface Description*** There are two main interfaces for connection to external memories: general-purpose memory controller (GPMC) and dual-channel SDRAM controller (SDRC). The GPMC supports: • Asynchronous SRAM memories • Asynchronous/synchronous NOR flash memories • NAND flash memories • Pseudo-SRAM devices The SDRC/EMIF allows: • Connection between the device and LPDDR2-type memory. It supports double-data rate (DDR) and single-data rate (SDR) protocols. The EMIF is the interface between LPDDR2 SDRAM/NVM and the Cortex-A9 MPU subsystem, ISS, IVA-HD subsystem, SGX, and DMA controllers. • PHY is the DDR physical interface, which implements data-rate conversion in compliance with LPDDR2 JEDEC requirements. OMAP4430 TRM at 283 [TPLBN008782].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

Figure 15-1. Memory Subsystem Functional Diagram



OMAP4430 TRM at 3207 [TPLBN011706] (Fig. 15-1).

15.1.3 EMIF Overview

The extended memory interface (EMIF) module is typically located near the DMM module, as shown in Figure 15-1.

The EMIF provides connectivity between the device and LPDDR2-type memory. It manages data bus read/write accesses between external memories, the microprocessor unit (MPU), and the direct memory access (DMA) controller. It acts as the primary interface between SDRAM and all functional blocks, such as the ARM®, IVA-HD, imaging subsystem (ISS), GFX, and DMA controllers.

OMAP4430 TRM at 3208 [TPLBN011707].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

	15.1.3.1 Main Features The EMIF has the following capabilities: • Supports JEDEC standard-compliant LPDDR2-SDRAM (S2 and S4) and LPDDR2-NVM devices • 2-GB SDRAM address range over two chip selects (CSs) (configurable with the DMM; see Section 15.2, *Dynamic Memory Manager*, for more information) • Supports two independent CSs, with their corresponding register sets, and independent page tracking • Both CSs must have the same memory type and size if they are both SDRAM or both nonvolatile memory (NVM). LPDDR2-SDRAM can be used in parallel with LPDDR2-NVM and can have a different size. • Flexible address muxing scheme which permits choosing different bank-mapping allocation by configuring the bank, column, and row-address decoding ordering • 16- or 32-bit data path to external SDRAM • Supports LPDDR2 devices with 1, 2, 4, or 8 internal banks • Supports the following data bus widths: – 128-bit level 3 (L3) interconnect data bus width – 16- and 32-bit SDRAM data bus width OMAP4430 TRM at 3209 [TPLBN011708]. ***15.1.4 GPMC Overview*** The general-purpose memory controller (GPMC) is an unified memory controller dedicated to interfacing external memory devices: • Asynchronous SRAM-like memories and application-specific integrated circuit (ASIC) devices • Asynchronous, synchronous, and page mode (available only in nonmuxed mode) burst NOR flash devices • NAND flash • Pseudo-SRAM devices OMAP4430 TRM at 3209 [TPLBN011708].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

	15.1.4.1 GPMC Features The GPMC is the external memory controller of the device. The GPMC data access engine provides a flexible programming model for communication with all standard memories. The GPMC supports various accesses: • Asynchronous read/write access • Asynchronous read page access (4, 8, and 16 Word16) • Synchronous read/write access • Synchronous read/write burst access without wrap capability (4, 8, and 16 Word16) • Synchronous read/write burst access with wrap capability (4, 8, and 16 Word16) • Address/data-multiplexed access • Little- and big-endian access The GPMC can communicate with a wide range of external devices: • External asynchronous or synchronous 8-bit wide memory or device (nonburst device) • External asynchronous or synchronous 16-bit wide memory or device • External 16-bit nonmultiplexed device with limited address range (2KB) • External 16-bit address/data-multiplexed NOR flash device • External 8-bit and 16-bit NAND flash device • External 16-bit pseudo-SRAM (pSRAM) device The main features of the GPMC are: • 8- or 16-bit-wide data path to external memory device OMAP4430 TRM at 3210 [TPLBN011709].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

Figure 15-50. GPMC to 16-Bit Address/Data-Multiplexed Memory

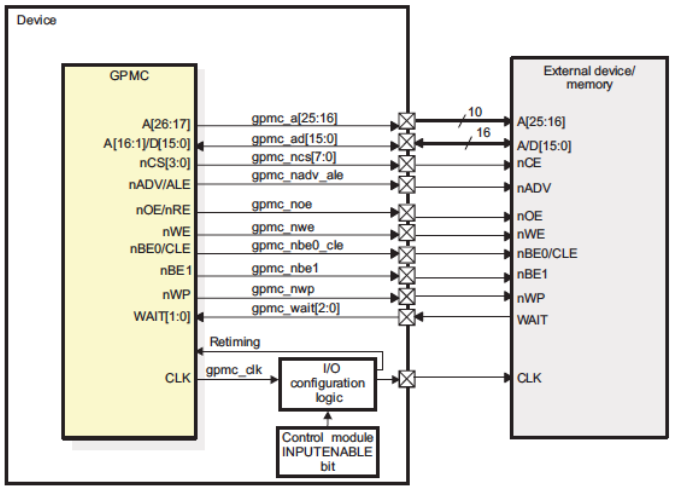
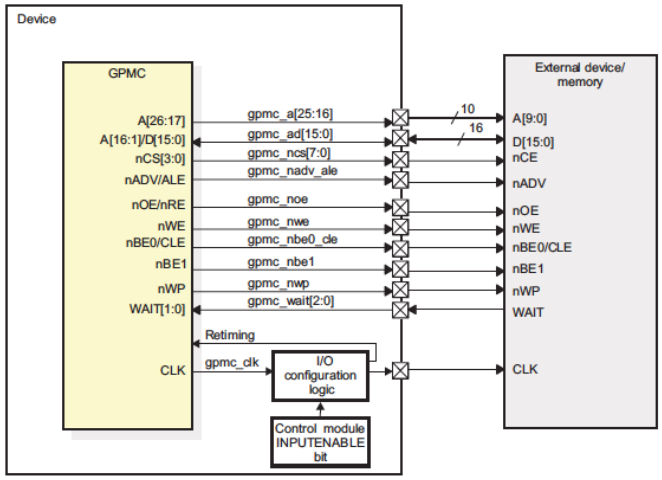
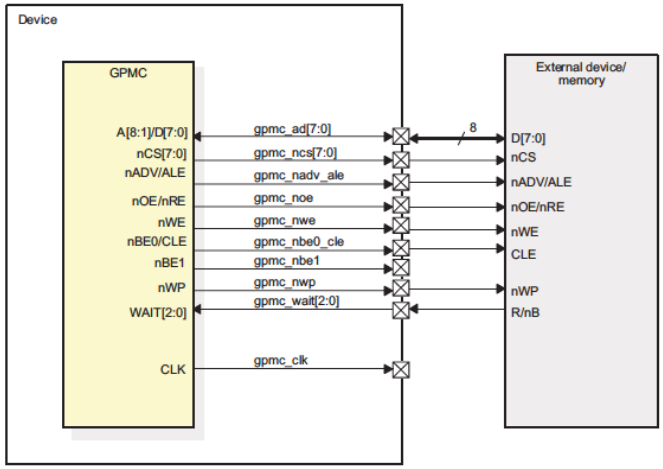


Figure 15-51. GPMC to 16-Bit Nonmultiplexed Memory



OMAP4430 TRM at 3366 [TPLBN011865].

Figure 15-52. GPMC to 8-Bit NAND Device



OMAP4430 TRM at 3367 [TPLBN011866].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

Figure 15-53. GPMC Integration

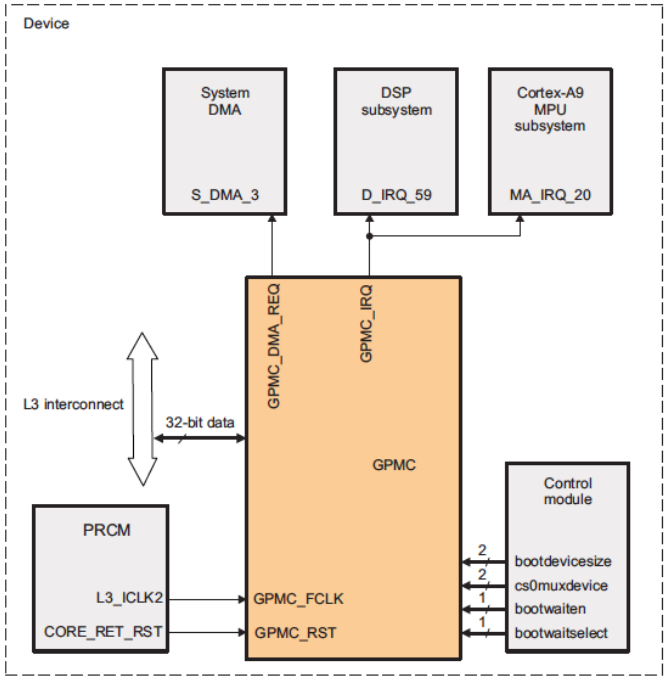
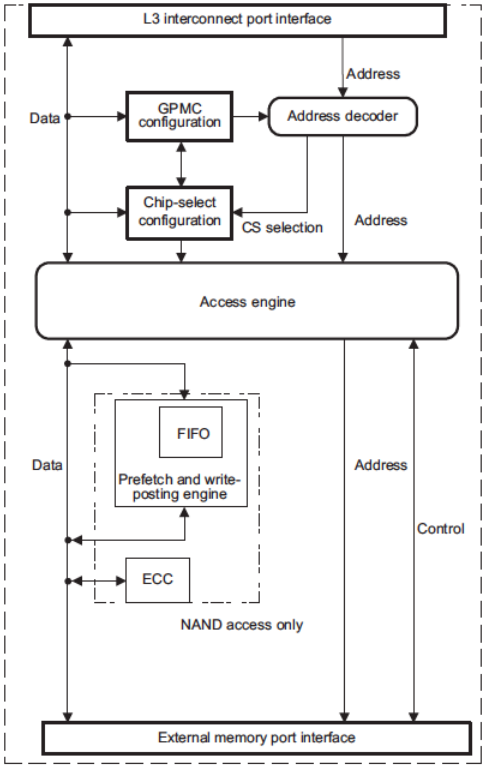


Figure 15-54. GPMC Block Diagram



OMAP4430 TRM at 3370 [TPLBN011869], 3373 [TPLBN011872].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

15.4.4.7 L3 Interconnect Interface

The GPMC L3 interconnect interface is a pipelined interface including an 16 x 32-bit word write buffer.

Any system host can issue external access requests through the GPMC.

The device system can issue the following requests through this interface:

- One 8-bit / 16-bit / 32-bit interconnect access (read/write)
- Two incrementing 32-bit interconnect accesses (read/write)
- Two wrapped 32-bit interconnect accesses (read/write)
- Four incrementing 32-bit interconnect accesses (read/write)
- Four wrapped 32-bit interconnect accesses (read/write)
- Eight incrementing 32-bit interconnect accesses (read/write)
- Eight wrapped 32-bit interconnect accesses (read/write)

OMAP4430 TRM at 3375 [TPLBN011874].

1.1 About the processor

The ARM1176JZF-S processor incorporates an integer core that implements the ARM11 ARM architecture v6. It supports the ARM and Thumb™ instruction sets, Jazelle technology to enable direct execution of Java bytecodes, and a range of SIMD DSP instructions that operate on 16-bit or 8-bit data values in 32-bit registers.

ARM11 TRM at 1-2 (28) [TPLBN035115], *see also* 2-2 (75) [TPLBN035162].

The ARM1176JZF-S processor features:

- Instruction and Data *Memory Management Units* (MMUs), managed using MicroTLB structures backed by a unified Main TLB
- Instruction and data caches, including a non-blocking data cache with *Hit-Under-Miss* (HUM)
- virtually indexed and physically addressed caches
- 64-bit interface to both caches
- level one *Tightly-Coupled Memory* (TCM) that you can use as a local RAM with DMA

ARM11 TRM at 1-2 (28) [TPLBN035115].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

1.4 ARM1176JZF-S architecture with Jazelle technology

The ARM1176JZF-S processor has three instruction sets:

- the 32-bit ARM instruction set used in ARM state, with media instructions
- the 16-bit Thumb instruction set used in Thumb state
- the 8-bit Java bytecodes used in Jazelle state.

For details of both the ARM and Thumb instruction sets, see the *ARM Architecture Reference Manual*. For full details of the ARM1176JZF-S Java instruction set, see the *Jazelle V1 Architecture Reference Manual*.

ARM11 TRM at 1-6 (32) [TPLBN035119].

1.4.2 The Thumb instruction set

The Thumb instruction set is a subset of the most commonly used 32-bit ARM instructions. Thumb instructions are 16 bits long, and have a corresponding 32-bit ARM instruction that has the same effect on the processor model. Thumb instructions operate with the standard ARM register configuration, enabling excellent interoperability between ARM and Thumb states.

Thumb has all the advantages of a 32-bit core:

- 32-bit address space
- 32-bit registers
- 32-bit shifter and *Arithmetic Logic Unit* (ALU)
- 32-bit memory transfer.

Thumb therefore offers a long branch range, powerful arithmetic operations, and a large address space.

The availability of both 16-bit Thumb and 32-bit ARM instruction sets, gives you the flexibility to emphasize performance or code size on a subroutine level, according to the requirements of their applications. For example, you can code critical loops for applications such as fast interrupts and DSP algorithms using the full ARM instruction set, and linked with Thumb code.

ARM11 TRM at 1-6 (32) [TPLBN035119].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

	1.5.1 Integer core The ARM1176JZF-S processor is built around the ARM11 integer core. It is an implementation of the ARMv6 architecture, that runs the ARM, Thumb, and Java instruction sets. The processor contains EmbeddedICE-RT™ logic and a JTAG debug interface to enable hardware debuggers to communicate with the processor. The following sections describe the core in more detail: ARM11 TRM at 1-8 (34) [TPLBN035121]. **Thumb instruction set** The Thumb instruction set contains a subset of the most commonly-used 32-bit ARM instructions encoded into 16-bit wide opcodes. This reduces the amount of memory required for instruction storage. ARM11 TRM at 1-10 (36) [TPLBN035123]. **2.3 Processor operating states** The processor has these operating states: **ARM state** 32-bit, word-aligned ARM instructions are executed in this state. **Thumb state** 16-bit, halfword-aligned Thumb instructions. **Jazelle state** Variable length, byte-aligned Java instructions. In Thumb state, the *Program Counter* (PC) uses bit 1 to select between alternate halfwords. In Jazelle state, all instruction fetches are in words. ARM11 TRM at 2-12 (85) [TPLBN035172].

1.5.4 Memory system

The level-one memory system provides the core with:

- separate instruction and data caches
- separate instruction and data Tightly-Coupled Memories
- 64-bit datapaths throughout the memory system
- virtually indexed, physically tagged caches
- memory access controls and virtual memory management
- support for four sizes of memory page
- two-channel DMA into TCMs
- I-fetch, D-read/write interface, compatible with multi-layer AMBA AXI
- 32-bit dedicated peripheral interface
- export of memory attributes for second-level memory system.

ARM11 TRM at 1-12 (38) [TPLBN035125].

7.1 About the level one memory system

The processor level one memory system consists of:

- separate Instruction and Data Caches in a Harvard arrangement
- separate Instruction and Data *Tightly-Coupled Memory* (TCM) areas
- a DMA system for accessing the TCMs

ARM11 TRM at 7-2 (373) [TPLBN035460].

7.2 Cache organization

Each cache is implemented as a four-way set associative cache of configurable size. The caches are virtually indexed and physically tagged. You can configure the cache sizes in the range of 4 to 64KB. Both the Instruction Cache and the Data Cache can provide two words per cycle for all requesting sources.

ARM11 TRM at 7-3 (374) [TPLBN035461].

Each Data TCM is implemented in parallel with the Data Cache and each Instruction TCM is implemented in parallel with the Instruction Cache. Each TCM has a single movable base address, specified in CP15 register c9, see *c9, Data TCM Region Register* on page 3-89 and *c9, Instruction TCM Region Register* on page 3-91.

ARM11 TRM at 7-7 (379) [TPLBN035466].

8.1.2 Level two instruction-side controller

The level two instruction-side controller contains the level two Instruction Fetch Interface. See *Instruction Fetch Interface*.

The level two instruction-side controller handles all instruction-side cache misses including those for Noncacheable locations. It is responsible for the sequencing of cache operations for Instruction Cache linefills, making requests for the individual stores through the *Prefetch Unit* (PU) to the Instruction Cache. The decoupling involved means that the level two instruction-side controller contains some buffering.

Instruction Fetch Interface

The Instruction Fetch Interface is a read-only interface that services the Instruction Cache on cache misses, including the fetching of instructions for the PU that are held in memory marked as Noncacheable. The interface is optimized for cache linefills rather than individual requests.

ARM11 TRM at 8-3 (390) [TPLBN035477].

The processor level two interconnect system uses the following 64-bit wide AXI interfaces:

- Instruction Fetch Interface
- Data Read/Write Interface
- DMA Interface.

Another interface is also provided, the Peripheral Interface. This is a 32-bit AXI interface.

ARM11 TRM at 8-2 (389) [TPLBN035476].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

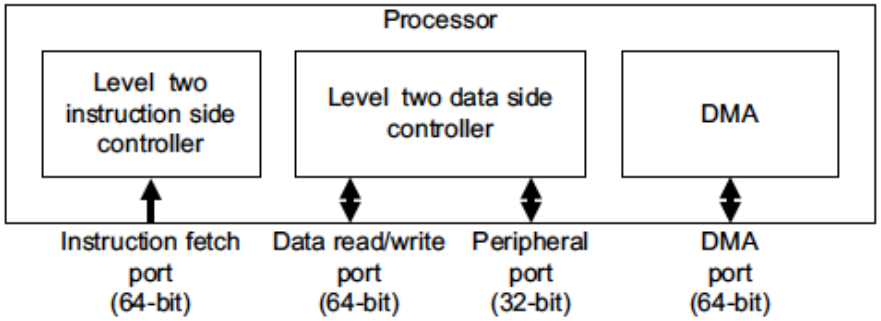


Figure 8-1 Level two interconnect interfaces

ARM11 TRM at 8-2 (389) [TPLBN035476].

1.5.3 Prefetch unit

The prefetch unit fetches instructions from the instruction cache, Instruction TCM, or from external memory and predicts the outcome of branches in the instruction stream.

ARM11 TRM at 1-11 (38) [TPLBN035125].

Instruction fetch

Servicing instruction cache misses and noncacheable instruction fetches.

ARM11 TRM at 1-16 (42) [TPLBN035129].

ARMv6-M is a subset of ARMv7-M, that provides:

- a lightweight version of the ARMv7-M programming model
- the Debug Extension that includes architecture extensions for debug support, see Chapter C1 *ARMv6-M Debug*.
- ARMv6 Thumb 16-bit instruction set compatibility at the application level

ARMv6 Reference Manual at A1-26 (26) [TPLBN035872].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

A5.1 Thumb instruction set encoding

The Thumb instruction stream is a sequence of halfword-aligned halfwords. Each Thumb instruction is either a single 16-bit halfword in that stream, or a 32-bit instruction consisting of two consecutive halfwords in that stream.

ARMv6 Reference Manual at A5-82 (82) [TPLBN035928].

A3.8 Caches and memory hierarchy

Support for caches in ARMv6-M is limited to memory attributes. These can be exported on a supporting bus protocol such as AMBA AHB or AMBA AXI to support system caches.

In situations where a breakdown in coherency can occur, software must manage the caches using cache maintenance operations that are memory mapped and IMPLEMENTATION DEFINED.

A3.8.1 Introduction to caches

A cache is a block of high-speed memory locations containing both address information and the associated data. The purpose is to increase the average speed of a memory access. Caches operate on two principles of locality:

Spatial locality an access to one location is likely to be followed by accesses from adjacent locations, for example sequential instruction execution or usage of a data structure

Temporal locality an access to an area of memory is likely to be repeated within a short time period, for example execution of a code loop.

ARMv6 Reference Manual at A3-63 (63) [TPLBN035909], *see also* A6-137 (137) [TPLBN035983].

Execution stream

The stream of instructions that would have been executed by sequential execution of the program.

ARMv6 Reference Manual at Glossary-428 (428) [(428) [TPLBN036274].

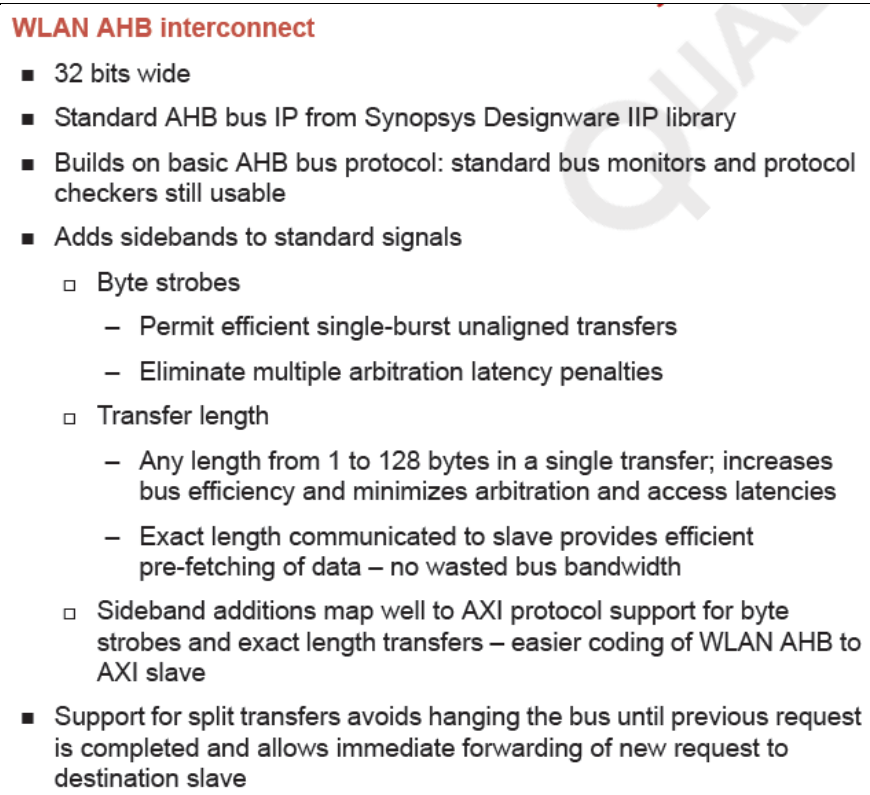
Note

ARMv7-M has limited support for 64-bit integers. Most 64-bit operations require sequences of two or more instructions to synthesize them.

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

	ARMv6 Reference Manual at A2-31 (31) [TPLBN035877]. The Fetch stages can hold up to four instructions, where branch prediction is performed on instructions ahead of execution of earlier instructions. The Issue and Decode stages can contain any instruction in parallel with a predicted branch. The Execute, Memory, and Write stages can contain a predicted branch, an ALU or multiply instruction, a load/store multiple instruction, and a coprocessor instruction in parallel execution. ARM11 TRM at 1-23 (53) [TPLBN035140].
[1f] said bus having a width at least equal to a number of bits in each of the instructions times a number of the instructions fetched in parallel,	On information and belief, in each Accused Product, said bus has a width at least equal to a number of bits in each of the instructions times a number of the instructions fetched in parallel. As to Accused Products with Qualcomm processors, *see, e.g.*: ▪ Improvements compared to MSM7225 and MSM7200A devices • Bus/processor speed enhancements ◆ 200 MHz AXI and AHB bus ◆ 400 MHz ARM9™ ◆ 600 MHz ARM11™ • 256 kB ARM11 L2 cache • ARM11 floating point Dual Krait µP CPUs, each with: • Up to 1.5 GHz • 1 MB L2 cache • 32 kB L1 instruction and data caches • ARM v7 compliant • TrustZone support • VeNum 128-bit SIMD MM coprocessor

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG



MSM7227 Chipset Training at 5; *see also* MSM8960 Design Guidelines at 454, 86, 88; http://www.anandtech.com/show/4940/qualcomm-new-snapdragon-s4-msm8960-krait-architecture (showing fetch and decode stages for the Krait core; "The architecture can fetch and decode three instructions per clock. The decoders are equally capable of decoding any ARMv7-A instructions."); and the ARM core documentation cited below.

As to Accused Products with OMAP processors and other ARM processors, *see, e.g.*:

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

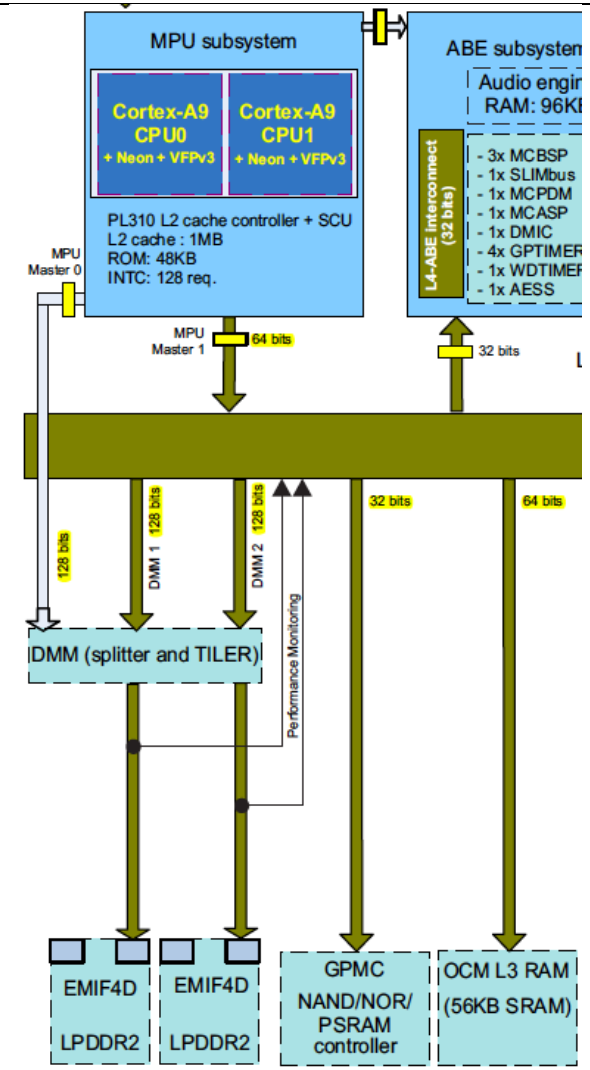
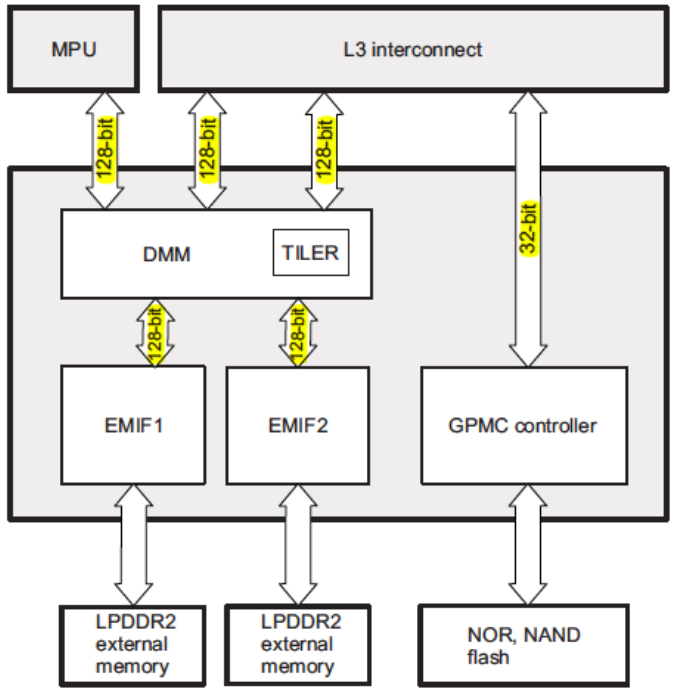


Figure 15-1. Memory Subsystem Functional Diagram



OMAP4430 TRM at 275 [TPLBN008774], 3207 [TPLBN011706].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

1.1 About the Cortex-A9 processor

The Cortex-A9 processor is a high-performance, low-power, ARM macrocell with an L1 cache subsystem that provides full virtual memory capabilities. The Cortex-A9 processor implements the ARMv7-A architecture and runs 32-bit ARM instructions, 16-bit and 32-bit Thumb instructions, and 8-bit Java™ bytecodes in Jazelle state.

Figure 1-1 shows a Cortex-A9 uniprocessor in a design with a PL390 Interrupt Controller and an L2C-310 L2 Cache Controller,

Cortex-A9 TRM at 1-2 (19) [TPLBN034876], *see also* 1-6 – 1-7 (23-24) [TPLBN034880-81], 3-2 (48) [TPLBN034905], A5-2 (108) [TPLBN034965]; *see also* Cortex-A8 TRM at 2-3 (43) [TPLBN034320].

8.1.1 About the Cortex-A9 L2 interface

The Cortex-A9 L2 interface consists of two 64-bit wide AXI bus masters:

- M0 is the data side bus
- M1 is the instruction side bus and has no write channels.

Cortex-A9 TRM at 8-2 (128) [TPLBN034985].

The EMIF has the following capabilities:

- Supports JEDEC standard-compliant LPDDR2-SDRAM (S2 and S4) and LPDDR2-NVM devices
- 2-GB SDRAM address range over two chip selects (CSs) (configurable with the DMM; see Section 15.2, *Dynamic Memory Manager*, for more information)
- Supports two independent CSs, with their corresponding register sets, and independent page tracking
- Both CSs must have the same memory type and size if they are both SDRAM or both nonvolatile memory (NVM). LPDDR2-SDRAM can be used in parallel with LPDDR2-NVM and can have a different size.
- Flexible address muxing scheme which permits choosing different bank-mapping allocation by configuring the bank, column, and row-address decoding ordering
- 16- or 32-bit data path to external SDRAM
- Supports LPDDR2 devices with 1, 2, 4, or 8 internal banks
- Supports the following data bus widths:
  - 128-bit level 3 (L3) interconnect data bus width
  - 16- and 32-bit SDRAM data bus width

OMAP4430 TRM at 3209 [TPLBN011708].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

	15.1.4.1 GPMC Features The GPMC is the external memory controller of the device. The GPMC data access engine provides a flexible programming model for communication with all standard memories. The GPMC supports various accesses: • Asynchronous read/write access • Asynchronous read page access (4, 8, and 16 Word16) • Synchronous read/write access • Synchronous read/write burst access without wrap capability (4, 8, and 16 Word16) • Synchronous read/write burst access with wrap capability (4, 8, and 16 Word16) OMAP4430 TRM at 3210-11 [TPLBN011709-10]. ***15.1.6 OCM Overview*** The on-chip memory (OCM) subsystem consists of the following OCM controllers: one connected to an on-chip ROM (SAR ROM), one connected to an on-chip RAM (SAR RAM), and one connected to an on-chip SRAM (L3 SRAM). Each memory controller has its own dedicated interface to the L3 interconnect. OMAP4430 TRM at 3211 [TPLBN011710].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

	15.1.6.1 SAR ROM This on-chip ROM contains 4KB of memory and a linked list of descriptors used by sDMA during the restore context operation (when the device transitions from off to on mode). The device-embedded SAR ROM has the following characteristics: • 4-KB ROM • 32-bit access per cycle • Support for single- and burst-access transactions **15.1.6.2 SAR RAM** The on-chip SAR RAM contains 8K bytes and is mapped as 4 blocks with irregular region sizes. This memory content is preserved when the device goes into off mode (as long as the wake-up voltage domain remains supplied). It is used as context-saving memory to be written by software so that sDMA restores its saved content when the device transitions from off to on mode. The device-embedded SAR RAM has the following characteristics: • Support for single-access transactions – Operates at full L4-PER interconnect clock frequency – 32-bit access per cycle **15.1.6.3 L3 OCM_RAM** The on-chip L3 OCM_RAM contains 56KB of RAM, and partitioning is defined by the L3 firewall logic. The device-embedded L3 OCM_RAM has the following characteristics: • Support for single and burst access transactions: – Operates at full L3 interconnect clock frequency – Fully pipelined, one 32-bit access per cycle • Restricted access support OMAP4430 TRM at 3211 [TPLBN011710].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

15.4.4.7 L3 Interconnect Interface

The GPMC L3 interconnect interface is a pipelined interface including an 16 x 32-bit word write buffer.

Any system host can issue external access requests through the GPMC.

The device system can issue the following requests through this interface:

- One 8-bit / 16-bit / 32-bit interconnect access (read/write)
- Two incrementing 32-bit interconnect accesses (read/write)
- Two wrapped 32-bit interconnect accesses (read/write)
- Four incrementing 32-bit interconnect accesses (read/write)
- Four wrapped 32-bit interconnect accesses (read/write)
- Eight incrementing 32-bit interconnect accesses (read/write)
- Eight wrapped 32-bit interconnect accesses (read/write)

OMAP4430 TRM at 3375 [TPLBN011874].

1.4.1 AMBA AXI interface

The AXI bus interface is the main interface to the system bus. It performs L2 cache fills and noncacheable accesses for both instructions and data. The AXI interface supports 64-bit or 128-bit wide input and output data buses. It also supports multiple outstanding requests on the AXI bus. The AXI signals are synchronous to the **CLK** input. A wide range of bus clock to core clock ratios is possible through the use of the AXI clock enable signal **ACLKEN**. See the *AMBA AXI Protocol Specification* for more information.

Cortex-A8 TRM, at 1-7 (32) [TPLBN034309], *see also* Glossary-2 (564) [TPLBN034841].

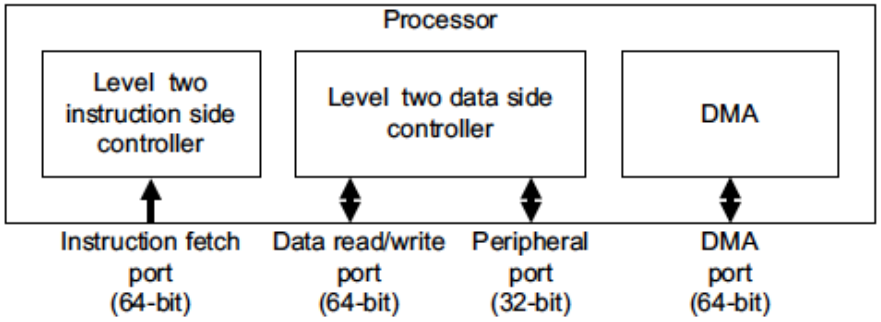


Figure 8-1 Level two interconnect interfaces

ARM11 TRM at 8-2 (389) [TPLBN035476].

1.1 About the processor

The ARM1176JZF-S processor incorporates an integer core that implements the ARM11 ARM architecture v6. It supports the ARM and Thumb™ instruction sets, Jazelle technology to enable direct execution of Java bytecodes, and a range of SIMD DSP instructions that operate on 16-bit or 8-bit data values in 32-bit registers.

ARM11 TRM at 1-2 (28) [TPLBN035115], *see also* 2-2 (75) [TPLBN035162].

1.4 ARM1176JZF-S architecture with Jazelle technology

The ARM1176JZF-S processor has three instruction sets:

- the 32-bit ARM instruction set used in ARM state, with media instructions
- the 16-bit Thumb instruction set used in Thumb state
- the 8-bit Java bytecodes used in Jazelle state.

For details of both the ARM and Thumb instruction sets, see the *ARM Architecture Reference Manual*. For full details of the ARM1176JZF-S Java instruction set, see the *Jazelle V1 Architecture Reference Manual*.

ARM11 TRM at 1-6 (32) [TPLBN035119], *see also* 1-10 (36) [TPLBN035123], 2-12 (85) [TPLBN035172].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

1.5.4 Memory system

The level-one memory system provides the core with:

- separate instruction and data caches
- separate instruction and data Tightly-Coupled Memories
- 64-bit datapaths throughout the memory system
- virtually indexed, physically tagged caches
- memory access controls and virtual memory management
- support for four sizes of memory page
- two-channel DMA into TCMs
- I-fetch, D-read/write interface, compatible with multi-layer AMBA AXI
- 32-bit dedicated peripheral interface
- export of memory attributes for second-level memory system.

ARM11 TRM at 1-12 (38) [TPLBN035125], *see also* 7-2 (373) [TPLBN035460].

The processor level two interconnect system uses the following 64-bit wide AXI interfaces:

- Instruction Fetch Interface
- Data Read/Write Interface
- DMA Interface.

Another interface is also provided, the Peripheral Interface. This is a 32-bit AXI interface.

ARM11 TRM at 8-2 (389) [TPLBN035476].

A5.1 Thumb instruction set encoding

The Thumb instruction stream is a sequence of halfword-aligned halfwords. Each Thumb instruction is either a single 16-bit halfword in that stream, or a 32-bit instruction consisting of two consecutive halfwords in that stream.

ARMv6 Reference Manual at A5-82 (82) [TPLBN035928]; *see also* A1-26 (26) [TPLBN035872], A3-63 (63)

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

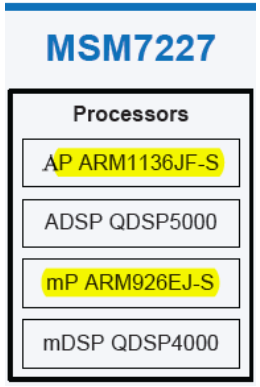
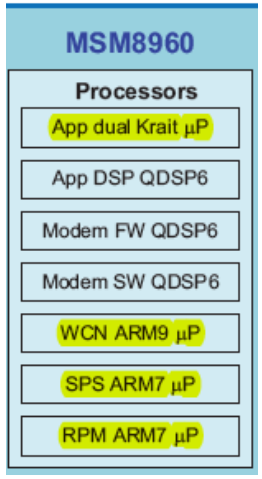
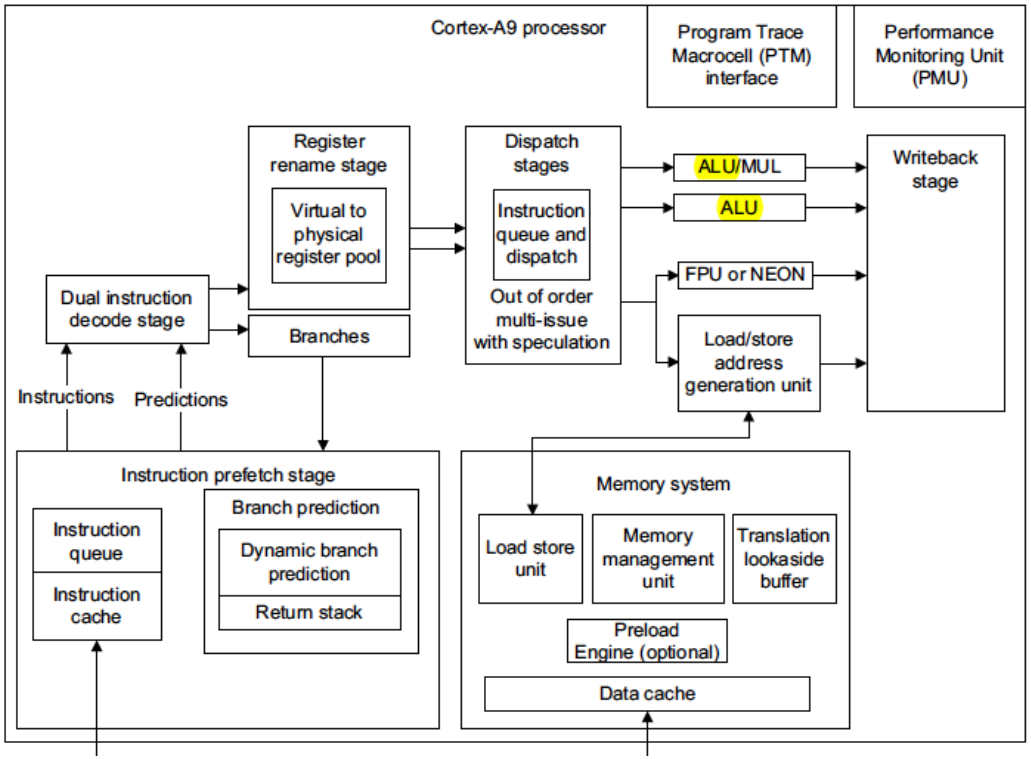
	[TPLBN035909], A6-137 (137) [TPLBN035983].
[1g] said central processing unit integrated circuit including an arithmetic logic unit and a first push down stack connected to said arithmetic logic unit,	On information and belief, each Accused Product has a central processing unit integrated circuit including an arithmetic logic unit and a first push down stack connected to said arithmetic logic unit. As to Accused Products with Qualcomm processors, *see, e.g.*:   MSM7227 Chipset Training at 7; MSM8960 Design Guidelines at 38; http://www.anandtech.com/show/4940/qualcomm-new-snapdragon-s4-msm8960-krait-architecture (Krait similar to ARM core and "capable of decoding any ARMv7-A instructions"). All Qualcomm processors, including those with ARM cores and other cores (*e.g.*, Scorpion, Krait, etc.) contain instruction registers and have pipelines that have multiple stages. *See* http://www.anandtech.com/show/4940/qualcomm-new-snapdragon-s4-msm8960-krait-architecture. *See also* the ARM core documentation cited below. As to Accused Products with ARM processors, *see, e.g.*:

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

Figure 2-1 shows a top-level diagram of the Cortex-A9 processor.



Cortex-A9 TRM at 2-2 (33) [TPLBN034890].

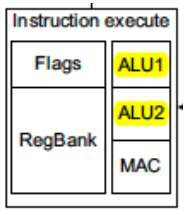
EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

11.3.1 Cortex-A9 specific events

Table 11-7 shows the Cortex-A9 specific events. In the value column of Table 11-7 Precise means the event is counted precisely. Events related to stalls and speculative instructions appear as Approximate entries in this column. s

0x70	Main execution unit instructions Counts the number of instructions being executed in the main execution pipeline of the processor, the multiply pipeline and arithmetic logic unit pipeline. The counted instructions are still speculative.	Approximate
0x71	Second execution unit instructions Counts the number of instructions being executed in the processor second execution pipeline (ALU). The counted instructions are still speculative.	Approximate

Cortex-A9 TRM at 11-7 – 11-8 (168-69) [TPLBN035025-26].



1.3.3 Instruction execute

The instruction execute unit consists of two symmetric *Arithmetic Logical Unit* (ALU) pipelines, an address generator for load and store instructions, and the multiply pipeline. The execute pipelines also perform register write back.

Cortex-A8 TRM at 1-4 – 1-6 (29-31) [TPLBN034306-08].

In addition, the ARM architecture gives you:

- control over both the *Arithmetic Logic Unit* (ALU) and shifter in every data-processing instruction to maximize the use of an ALU and a shifter

ARM Architecture Reference Manual at A1-2 [PIC00005214].

The Arithmetic Logic Unit (ALU) derives its two inputs from the 'Top item' (Rn) and the 'Next item' (Rm) of the 'Push Down Stack' from the 'General Purpose Registers' (the holding place for items placed there by stack operations) and directs its output (Rd) back to the 'General Purpose Registers'.

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

A8.6.2 ADC (register)

Add with Carry (register) adds a register value, the carry flag value, and an optionally-shifted register value, and writes the result to the destination register. It can optionally update the condition flags based on the result.

Encoding T1 ARMv4T, ARMv5T*, ARMv6*, ARMv7

```
ADCS <Rdn>,<Rm>                                   Outside IT block.
ADC<c> <Rdn>,<Rm>                                 Inside IT block.
```

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	0	0	1	0	1	Rm			Rdn		

```
d = UInt(Rdn);  n = UInt(Rdn);  m = UInt(Rm);  setflags = !InITBlock();
(shift_t, shift_n) = (SRType_LSL, 0);
```

Encoding T2 ARMv6T2, ARMv7

```
ADC{S}<c>.W <Rd>,<Rn>,<Rm>{,<shift>}
```

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	1	0	1	1	0	1	0	S	Rn				(0)	imm3			Rd				imm2		type		Rm			

```
d = UInt(Rd);  n = UInt(Rn);  m = UInt(Rm);  setflags = (S == '1');
(shift_t, shift_n) = DecodeImmShift(type, imm3:imm2);
if BadReg(d) || BadReg(n) || BadReg(m) then UNPREDICTABLE;
```

Encoding A1 ARMv4*, ARMv5T*, ARMv6*, ARMv7

```
ADC{S}<c> <Rd>,<Rn>,<Rm>{,<shift>}
```

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cond				0	0	0	0	1	0	1	S	Rn				Rd				imm5				type		0	Rm				

```
if Rd == '1111' && S == '1' then SEE SUBS PC, LR and related instructions;
d = UInt(Rd);  n = UInt(Rn);  m = UInt(Rm);  setflags = (S == '1');
(shift_t, shift_n) = DecodeImmShift(type, imm5);
```

ARMv7 Reference Manual at A8-16 (328) [TPLBN049616].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

Assembler syntax

```
ADC{S}<c><q>  {<Rd>,} <Rn>, <Rm> {,<shift>}
```

where:

S	If S is present, the instruction updates the flags. Otherwise, the flags are not updated.
<c><q>	See *Standard assembler syntax fields* on page A8-7.
<Rd>	The destination register.
<Rn>	The first operand register.
<Rm>	The optionally shifted second operand register.
<shift>	The shift to apply to the value read from <Rm>. If present, encoding T1 is not permitted. If absent, no shift is applied and any encoding is permitted. *Shifts applied to a register* on page A8-10 describes the shifts and how they are encoded.

ARMv7 Reference Manual at A8-17 (329) [TPLBN049617].

The Register File includes register R13, also known as SP or the Stack pointer, which is a pointer to the active stack.

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

A2.3 ARM core registers

In the application level view, an ARM processor has:

- thirteen general-purpose32-bit registers, R0 to R12
- three 32-bit registers, R13 to R15, that sometimes or always have a special use.

Registers R13 to R15 are usually referred to by names that indicate their special uses:

SP, the Stack Pointer

Register R13 is used as a pointer to the active stack.

In Thumb code, most instructions cannot access SP. The only instructions that can access SP are those designed to use SP as a stack pointer.

The use of SP for any purpose other than as a stack pointer is deprecated.

Note

Using SP for any purpose other than as a stack pointer is likely to break the requirements of operating systems, debuggers, and other software systems, causing them to malfunction.

LR, the Link Register

Register R14 is used to store the return address from a subroutine. At other times, LR can be used for other purposes.

When a `BL` or `BLX` instruction performs a subroutine call, LR is set to the subroutine return address. To perform a subroutine return, copy LR back to the program counter. This is typically done in one of two ways, after entering the subroutine with a `BL` or `BLX` instruction:

- Return with a `BX LR` instruction.
- On subroutine entry, store LR to the stack with an instruction of the form:
  `PUSH {<registers>,LR}`
  and use a matching instruction to return:
  `POP {<registers>,PC}`

ThumbEE checks and handler calls use LR in a similar way. For details see Chapter A9 *ThumbEE*.

ARMv7 Reference Manual at A2-11 (43) [TPLBN049331]; *see also* Cortex-A9 TRM at 1-4 (21) [TPLBN034878], 2-2 (33) [TPLBN034890], 7-7 (121) [TPLBN034978].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

A6.1.3 Use of 0b1101 as a register specifier

R13 is defined in the Thumb instruction set so that its use is primarily as a stack pointer, and R13 is normally identified as SP in Thumb instructions. In 32-bit Thumb instructions, if you use R13 as a general-purpose register beyond the architecturally defined constraints described in this section, the results are UNPREDICTABLE.

The restrictions applicable to R13 are described in:

- *R13[1:0] definition*
- *32-bit Thumb instruction support for R13.*

See also *16-bit Thumb instruction support for R13* on page A6-5.

ARMv7 Reference Manual at A6-4 (242) [TPLBN049530].

The top two items in the stack are connected to provide inputs into the ALU by using a "POP".

A8.6.122 POP

Pop Multiple Registers loads multiple registers from the stack, loading from consecutive memory locations starting at the address in SP, and updates SP to point just above the loaded data.

ARMv7 Reference Manual at A8-246 (558) [TPLBN049846].

The output of the ALU is connected to the top of the stack and may be saved with a "PUSH."

A8.6.123 PUSH

Push Multiple Registers stores multiple registers to the stack, storing to consecutive memory locations ending just below the address in SP, and updates SP to point to the start of the stored data.

ARMv7 Reference Manual at A8-248 (560) [TPLBN049848].

In the alternative, ARM Cortex A9 processor also includes an operand stack for Jazelle DBX instructions.

A2.10.2 Jazelle direct bytecode execution support

From ARMv5TEJ, the architecture requires every system to include an implementation of the Jazelle extension. The Jazelle extension provides architectural support for hardware acceleration of bytecode execution by a *Java Virtual Machine* (JVM).

ARMv7 Reference Manual at A2-73 (105) [TPLBN049393].

Reads

Reads are defined as memory operations that have the semantics of a load.

The memory accesses of the following instructions are reads:

- LDR, LDRB, LDRH, LDRSB, and LDRSH
- LDRT, LDRBT, LDRHT, LDRSBT, and LDRSHT
- LDREX, LDREXB, LDREXD, and LDREXH
- LDM, LDRD, POP, and RFE
- LDC, LDC2, VLDM, VLDR, VLD1, VLD2, VLD3, and VLD4
- the return of status values by STREX, STREXB, STREXD, and STREXH
- in the ARM instruction set only, SWP and SWPB
- in the Thumb instruction set only, TBB and TBH.

Hardware-accelerated opcode execution by the Jazelle extension can cause a number of reads to occur, according to the state of the operand stack and the implementation of the Jazelle hardware acceleration.

Writes

Writes are defined as memory operations that have the semantics of a store.

The memory accesses of the following instructions are Writes:

- STR, STRB, and STRH
- STRT, STRBT, and STRHT
- STREX, STREXB, STREXD, and STREXH
- STM, STRD, PUSH, and SRS
- STC, STC2, VSTM, VSTR, VST1, VST2, VST3, and VST4
- in the ARM instruction set only, SWP and SWPB.

Hardware-accelerated opcode execution by the Jazelle extension can cause a number of writes to occur, according to the state of the operand stack and the implementation of the Jazelle hardware acceleration.

ARMv7 Reference Manual at A3-42 (156) [TPLBN049444].

The operand stack contains a top item register and a next item register connected to provide inputs to the ALU and the output of the ALU is connected to the top item register. The first four elements of the stack are held in

the register file in registers R0-R3. The stack pointer is held in register R6.

> In Java State, the processor assigns several ARM registers to functions specific to the Java machine (e.g., R6= stack pointer, R0-R3= top elements of stack, R4=local variable 0). This hardware reuse contributes to the small size of the additional logic (12k gates) required to implement the Java machine, and keeps all of the state required by the Jazelle extension in ARM registers, In addition, it ensures compatibility with existing operating systems, interrupt handlers and exception code.

ARM White Paper, Accelerating to Meet the Challenges of Embedded Java™ [TPLBN051478-TPLBN051482] ("ARM White Paper") at 3 [TPLBN051480].

> **Datapath**
>
> The datapath consists of three pipelines:
> - ALU, shift and Sat pipeline
> - MAC pipeline
> - load or store pipeline, see *Load Store Unit (LSU)* on page 1-11.
>
> ***ALU, shift or Sat pipe***
>
> The ALU, shift and Sat pipeline executes most of the ALU operations, and includes a 32-bit barrel shifter. It consists of three pipeline stages:
>
> **Shift** The Shift stage contains the full barrel shifter. This stage performs all shifts, including those required by the LSU.
>
> The Shift stage implements saturating left shift that doubles the value of an operand and saturates it.

ARM11 TRM at 1-10 (36) [TPLBN035123].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

	ALU The ALU stage performs all arithmetic and logic operations, and generates the condition codes for instructions that set these flags. The ALU stage consists of a logic unit, an arithmetic unit, and a flag generator. The pipeline logic evaluates the flag settings in parallel with the main adder in the ALU. The flag generator is enabled only on flag-setting operations. The ALU stage separates the carry chains of the main adder for 8-bit and 16-bit SIMD instructions. **Sat** The Sat stage implements the saturation logic required by the various classes of DSP instructions. ARM11 TRM at 1-11 (37) [TPLBN035124]. **Core** A core is that part of a processor that contains the ALU, the datapath, the general-purpose registers, the Program Counter, and the instruction decode and control circuitry. ARM11 TRM at Glossary-8 (747) [TPLBN035834]. Plaintiffs contend that this claim element is literally present as described above. In the event that this claim element is not found to be literally present, Plaintiffs contend that the above-identified stacks are equivalent to "a first push down stack connected to said arithmetic logic unit," and any differences are insubstantial. In particular, the stacks perform the same function (*i.e.*, input and output to the ALU), in substantially the same way (*i.e.*, by providing a last-in, first-out data structure), and have the same result (*i.e.*, the ALU performs operations on inputs and provides output).
[1h] said first push down stack including means for storing a top item connected to	On information and belief, in each Accused Product, the first push down stack includes 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. On information and belief, each Accused Product has 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. The Arithmetic Logic Unit (ALU) derives its two inputs from the 'Top item' (Rn) and the 'Next item' (Rm) of

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

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,	the 'Push Down Stack' from the 'General Purpose Registers' (the holding place for items placed there by stack operations) and directs its output (Rd) back to the 'General Purpose Registers'. *See also* the evidence shown above for item 1g.

A8.6.2 ADC (register)

Add with Carry (register) adds a register value, the carry flag value, and an optionally-shifted register value, and writes the result to the destination register. It can optionally update the condition flags based on the result.

Encoding T1 ARMv4T, ARMv5T*, ARMv6*, ARMv7

```
ADCS <Rdn>,<Rm>                                   Outside IT block.
ADC<c> <Rdn>,<Rm>                                 Inside IT block.
```

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	0	0	1	0	1	Rm			Rdn		

```
d = UInt(Rdn);  n = UInt(Rdn);  m = UInt(Rm);  setflags = !InITBlock();
(shift_t, shift_n) = (SRType_LSL, 0);
```

Encoding T2 ARMv6T2, ARMv7

```
ADC{S}<c>.W <Rd>,<Rn>,<Rm>{,<shift>}
```

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	1	0	1	1	0	1	0	S	Rn				(0)	imm3			Rd				imm2		type		Rm			

```
d = UInt(Rd);  n = UInt(Rn);  m = UInt(Rm);  setflags = (S == '1');
(shift_t, shift_n) = DecodeImmShift(type, imm3:imm2);
if BadReg(d) || BadReg(n) || BadReg(m) then UNPREDICTABLE;
```

Encoding A1 ARMv4*, ARMv5T*, ARMv6*, ARMv7

```
ADC{S}<c> <Rd>,<Rn>,<Rm>{,<shift>}
```

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cond				0	0	0	0	1	0	1	S	Rn				Rd				imm5				type		0	Rm				

```
if Rd == '1111' && S == '1' then SEE SUBS PC, LR and related instructions;
d = UInt(Rd);  n = UInt(Rn);  m = UInt(Rm);  setflags = (S == '1');
(shift_t, shift_n) = DecodeImmShift(type, imm5);
```

ARMv7 Reference Manual at A8-16 (328) [TPLBN049616]; *see also* ARMv6 Reference Manual at A6-106

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

(106) [TPLBN035952].

Assembler syntax

```
ADC{S}<c><q>  {<Rd>,} <Rn>, <Rm> {,<shift>}
```

where:

S	If S is present, the instruction updates the flags. Otherwise, the flags are not updated.
<c><q>	See *Standard assembler syntax fields* on page A8-7.
<Rd>	The destination register.
<Rn>	The first operand register.
<Rm>	The optionally shifted second operand register.
<shift>	The shift to apply to the value read from <Rm>. If present, encoding T1 is not permitted. If absent, no shift is applied and any encoding is permitted. *Shifts applied to a register* on page A8-10 describes the shifts and how they are encoded.

ARMv7 Reference Manual at A8-17 (329) [TPLBN049617]; *see also* ARMv6 Reference Manual at A6-106 (106) [TPLBN035952].

On information and belief, the operand stack contains a top item register and a next item register connected to provide inputs to the ALU and the output of the ALU is connected to the top item register.

The top item in the stack is connected to provide input to an internal data bus by using a “POP” which loads it into a register.

A8.6.122 POP

Pop Multiple Registers loads multiple registers from the stack, loading from consecutive memory locations starting at the address in SP, and updates SP to point just above the loaded data.

ARMv7 Reference Manual at A8-246 (558) [TPLBN049846]; *see also* ARMv6 Reference Manual at A6-165 (165) [TPLBN036011].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

	The output of the ALU is connected to the top of the stack and may be saved with a "PUSH." **A8.6.123 PUSH** Push Multiple Registers stores multiple registers to the stack, storing to consecutive memory locations ending just below the address in SP, and updates SP to point to the start of the stored data. ARMv7 Reference Manual at A8-248 (560) [TPLBN049848]; *see also* ARMv6 Reference Manual at A6-167 (167) [TPLBN036013]. The top item of the stack is also located in the data cache within the "memory system." The register file is connected through the memory system which is bidirectionally connected to the data bus of the processors of each of the Accused Products, as shown above.

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

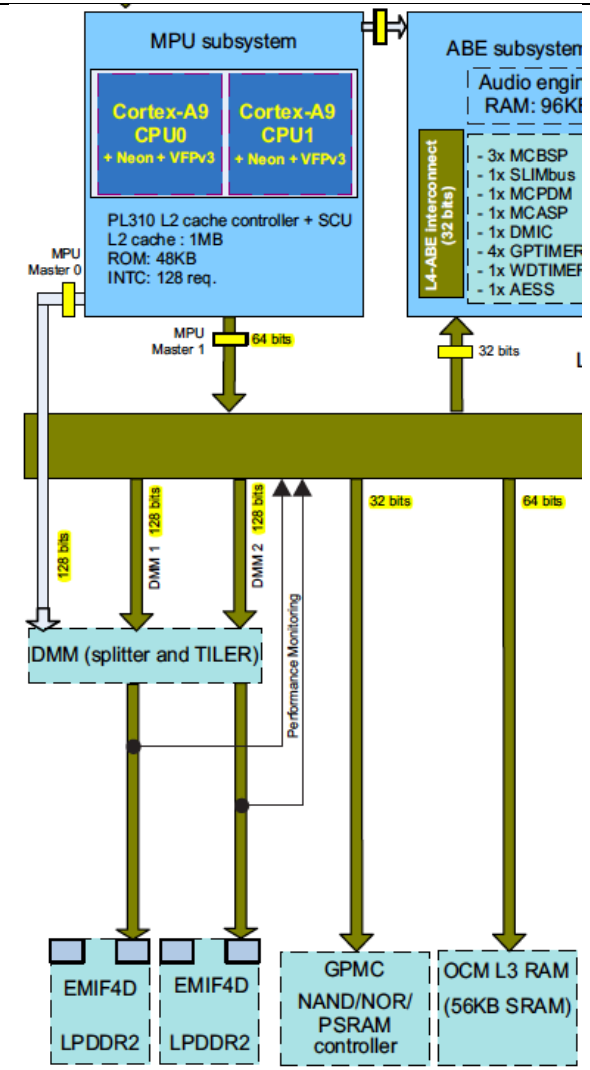
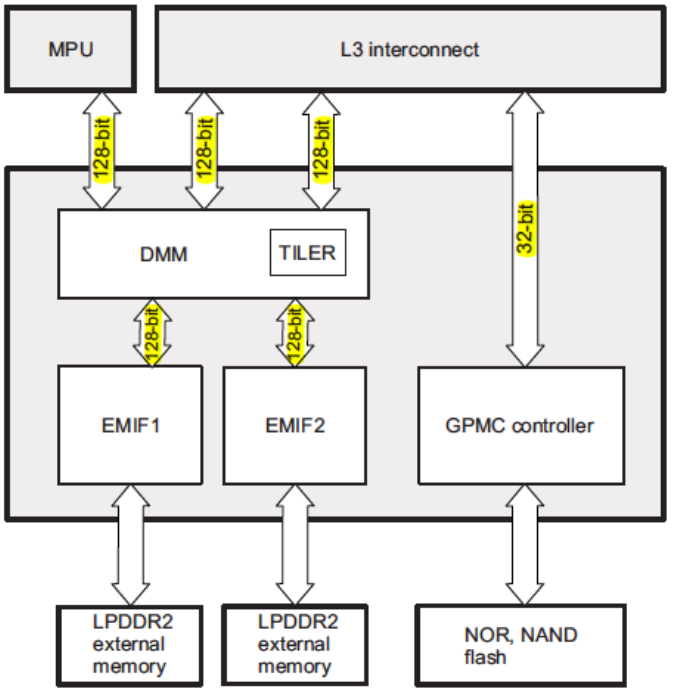
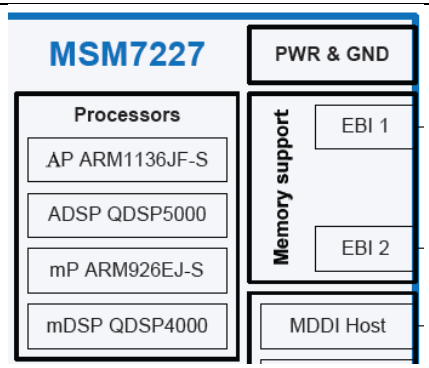


Figure 15-1. Memory Subsystem Functional Diagram



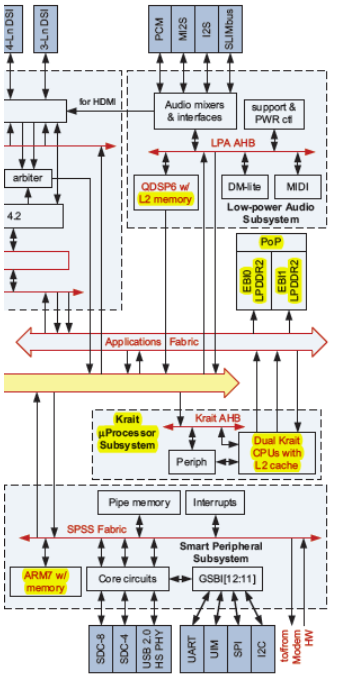
OMAP4430 TRM at 275 [TPLBN008774], 3207 [TPLBN011706]; *see also* OMAP4460 TRM at 263, 2982; OMAP4470 TRM at 276 [TPLBN021476], 1090 [TPLBN022290], 3280 [TPLBN024480]; OMAP4430 TRM at 275 [TPLBN008774], 1077 [TPLBN009576], 3207 [TPLBN011706].

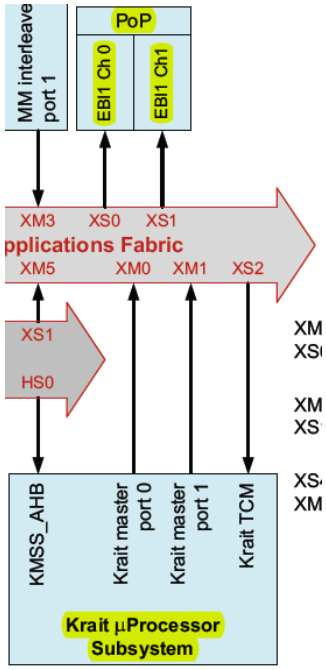
EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG



- Improvements compared to MSM7225 and MSM7200A devices
  - Bus/processor speed enhancements
    - 200 MHz AXI and AHB bus
    - 400 MHz ARM9™
    - 600 MHz ARM11™
  - 256 kB ARM11 L2 cache
  - ARM11 floating point

MSM7227 Chipset Training at 5, 7. *See also* the ARM core documentation cited below.





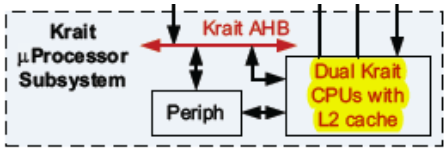


EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

Key DDR memory controller features

- Supports DDR on address bus
- Two major clock domains (AXI slave and DDR controller)
- Multiple AXI-to-DDR clock modes
  – Synchronous (1:1), iso-synchronous (1:2), asynchronous
- Highly flexible with many configurable parameters, including interface timing
- Flexible memory page management with various page open/close policies
- Out-of-order command execution and read data return
- Multiple high-priority tiers for sophisticated handling of priority requests

Major MSM8960 functional blocks (left) are shared by its internal systems, subsystem, and busses – the 'architecture' (right)

MSM8960 Design Guidelines at 38, 44, 51, 94. *See also* the ARM core documentation cited below.

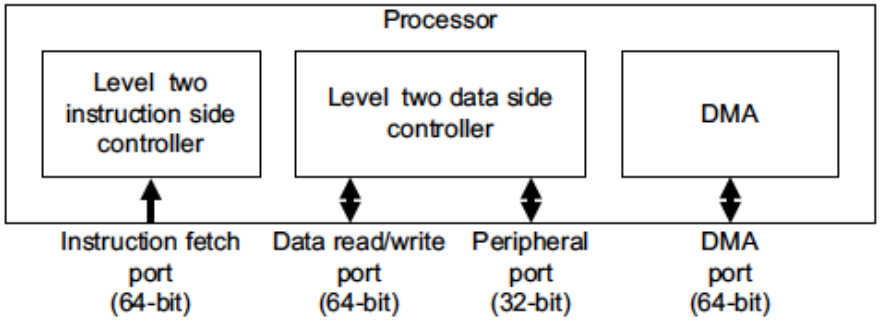


Figure 8-1 Level two interconnect interfaces

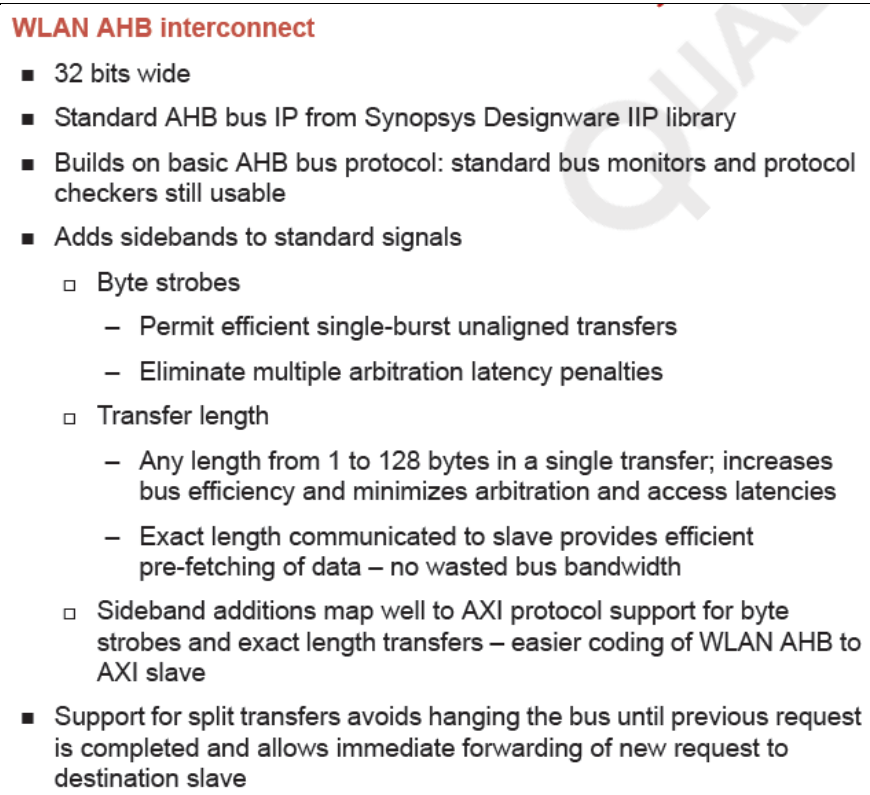
ARM11 TRM at 8-2 (389) [TPLBN035476].

As discussed in connection with claim element 1g, Plaintiffs contend that this claim element is literally present as described above. In the event that this claim element is not found to be literally present, Plaintiffs contend that the above-identified stacks are equivalent to "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" and any differences are insubstantial. In particular, the stacks perform the same function (*i.e.*, input and output to the ALU), in substantially the same way (*i.e.*, by providing a last-in, first-out data structure),

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

	and have the same result (*i.e.*, the ALU performs operations on inputs and provides output). *See* element 1g, above.
[1i] 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,	On information and belief, each Accused Product has 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. As discussed above in connection with element 1g, the Arithmetic Logic Unit (ALU) derives its two inputs from the 'Top item' (Rn) and the 'Next item' (Rm) of the 'Push Down Stack' from the 'General Purpose Registers' (the holding place for items placed there by stack operations) and directs its output (Rd) back to the 'General Purpose Registers'. *See* element 1g, above. As to Accused Products with Qualcomm processors, *see e.g.*: ▪ Improvements compared to MSM7225 and MSM7200A devices • Bus/processor speed enhancements ◆ 200 MHz AXI and AHB bus ◆ 400 MHz ARM9™ ◆ 600 MHz ARM11™ • 256 kB ARM11 L2 cache • ARM11 floating point Instruction Cache Controller The instruction cache controller fetches the instructions from memory depending on the program flow predicted by the prefetch unit.

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG



MSM7227 Chipset Training at 7; *see also* MSM8960 Design Guidelines at 454, 86, 88; http://www.anandtech.com/show/4940/qualcomm-new-snapdragon-s4-msm8960-krait-architecture (showing fetch and decode stages for the Krait core; "The architecture can fetch and decode three instructions per clock. The decoders are equally capable of decoding any ARMv7-A instructions."); *id* (showing decoding of multiple instructions in parallel). *See also* the ARM core documentation cited below.

Each ARM processor has a set of multiple registers with 32 bits:

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

2.13 Registers

The processor has a total of 40 registers:

- 33 general-purpose 32-bit registers
- seven 32-bit status registers.

These registers are not all accessible at the same time. The processor state and mode of operation determine the registers that are available to the programmer.

2.13.1 The state register set

In ARM state, 16 data registers and one or two status registers are accessible at any time. In privileged modes, mode-specific banked registers become available. Figure 2-10 on page 2-19 shows the registers that are available in each mode.

Thumb and ThumbEE state give access to the same set of registers as ARM state. However, the 16-bit instructions provide only limited access to some of the registers. No such limitations exist for 32-bit Thumb-2 and ThumbEE instructions.

Registers r0 through r13 are general-purpose registers used to hold either data or address values.

Cortex-A8 TRM at 2-18 (58) [TPLBN034335]; *see also* ARM11 TRM at 2-18 (91) [TPLBN035178].

The scheme maps the 32 ARM architectural registers to a pool of 56 physical 32-bit registers, and renames the flags (N, Z, C, V, Q, and GE) of the CPSR using a dedicated pool of eight physical 9-bit registers.

Cortex-A9 TRM at 2-2 (33) [TPLBN034890].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

Figure 2-11 shows an alternative view of the ARM registers.

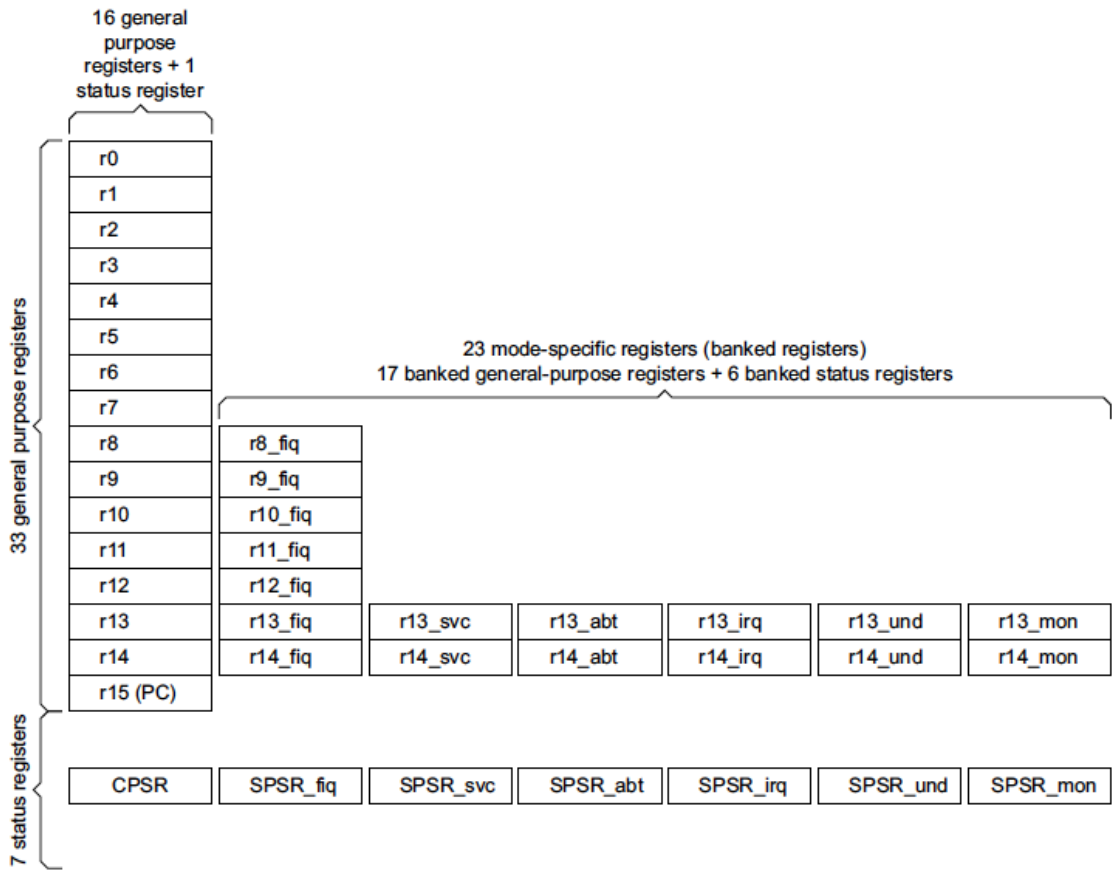


Figure 2-11 Processor register set showing banked registers

Cortex-A8 TRM at 2-20 (60) [TPLBN034337]; *see also* ARM11 TRM at 2-21 (94) [TPLBN035181]; A9 TRM at 7-7 (121) [TPLBN034978].

As discussed in connection with claim element 1g and 1h, Plaintiffs contend that this claim element is literally present as described above; however, for the reasons stated above, in the event that this claim element is not found to be literally present, Plaintiffs contend that the above-identified stacks are equivalent to this claim

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

	element. *See* elements 1g and 1h, above. As discussed in connection with claim element 1g and 1h, Plaintiffs contend that this claim element is literally present as described above; however, for the reasons stated above, in the event that this claim element is not found to be literally present, Plaintiffs contend that the above-identified stacks are equivalent to this claim element. *See* elements 1g and 1h, above.
[1j] said arithmetic logic unit having an output connected to said means for storing a top item;	On information and belief, in each Accused Product, said arithmetic logic unit has an output connected to said means for storing a top item. As discussed above in connection with element 1g, the Arithmetic Logic Unit (ALU) derives its two inputs from the 'Top item' (Rn) and the 'Next item' (Rm) of the 'Push Down Stack' from the 'General Purpose Registers' (the holding place for items placed there by stack operations) and directs its output (Rd) back to the 'General Purpose Registers'. *See* element 1g, above; *see also* elements 1h, and 1i, above. The output is stored in these “General Purpose Registers.” *See also* http://www.anandtech.com/show/4940/qualcomm-new-snapdragon-s4-msm8960-krait-architecture (showing fetch and decode stages for the Krait core; “The architecture can fetch and decode three instructions per clock. The decoders are equally capable of decoding any ARMv7-A instructions.”); *id* (showing decoding of multiple instructions in parallel). *See also* the ARM core documentation cited herein.
[1k] wherein the microprocessor system comprises an instruction register configured to store the multiple sequential instructions and from which	On information and belief, in each Accused Product, the microprocessor system comprises an instruction register configured to store the multiple sequential instructions and from which instructions are accessed and decoded. As to Accused Products with ARM processors, *see, e.g.*:

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

instructions are accessed and decoded; and	***1.3.1 Cortex-A9 MPU Subsystem Description*** The Cortex-A9 MPU subsystem integrates the following submodules: • ARM Cortex-A9 MPCore – Two ARM Cortex-A9 central processing units (CPUs) – ARM Version 7 ISA™: Standard ARM instruction set plus Thumb®-2, Jazelle® RCT and Jazelle DBX Java™ accelerators – Neon™ SIMD coprocessor and VFPv3 per CPU – Interrupt controller (Cortex-A9 MPU INTC) with up to 128 interrupt requests – One general-purpose timer and one watchdog timer per CPU – Debug and trace features – 32-KB instruction and 32-KB data level 1 (L1) caches per CPU • Shared 1-MB level 2 (L2) cache • 48 KB bootable ROM • Local power, reset, and clock managment (PRCM) module • Emulation features • Digital phase-locked loop (DPLL) OMAP4430 TRM at 276 [TPLBN008775]; *see also* ARMv7 Reference Manual at A3-3 (117) [TPLBN049405] re: "prefetch[ing] instructions," A3-28 (142) [TPLBN049430] re: "instruction fetches".

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

	4.3.1 Cortex-A9 MPU Subsystem Block Diagram The Cortex-A9 MPU subsystem integrates the following group of submodules: • Two ARM Cortex-A9 CPUs. Each CPU contains: – ARM version 7 ISA™: Standard ARM instruction set plus Thumb®-2 , Jazelle® RCT and Jazelle DBX Java™ accelerator – Neon SIMD coprocessor and VFPv3 • INTC: Handles module interrupts (For details, see Chapter 17, *Interrupt Controller.*) • PL310 L2 cache controller (revision r2p0) with 1MB cache and two 64-bit slave and two 64-bit master ports (For details about L2 cache controller, see the ARM PL310 Cache Controller TRM, available at infocenter.arm.com/help/index.jsp). • Local interconnect: Connects the ARM Cortex-A9 multicore processor to the level 3 (L3) interconnect, dynamic memory manager, ABE interconnect, local PRCM, PL310 L2 cache controller, on-chip ROM memory, and the WUGEN. • Power, clock and reset manager • On-chip ROM memory – CPU0 (the master CPU) can boot from this memory. The ROM memory size is 48KB, the address range is from 0x4002 8000 to 0x4003 3FFF. For more information, see Chapter 27, *Initialization*. • WUGEN: Responsible for waking up the CPUs, used by the ROM code and OS during SMP boot. Two internal memory-mapped registers, AUX_CORE_BOOT_0 and AUX_CORE_BOOT_1, are available to the OS for communicating start-up information. For more information, see Chapter 27, *Initialization*. • Standby controllers – handle the power transitions inside the Cortex-A9 MPU subsystem. • SCU for L1 cache coherency • One timer and one watchdog unit per core • Caches memories: – 32-KB L1 instruction and 32-KB data caches – 4-way associative on each core – 1-MB L2 unified cache – 16-way associative OMAP4430 TRM at 1084 [TPLBN009583].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

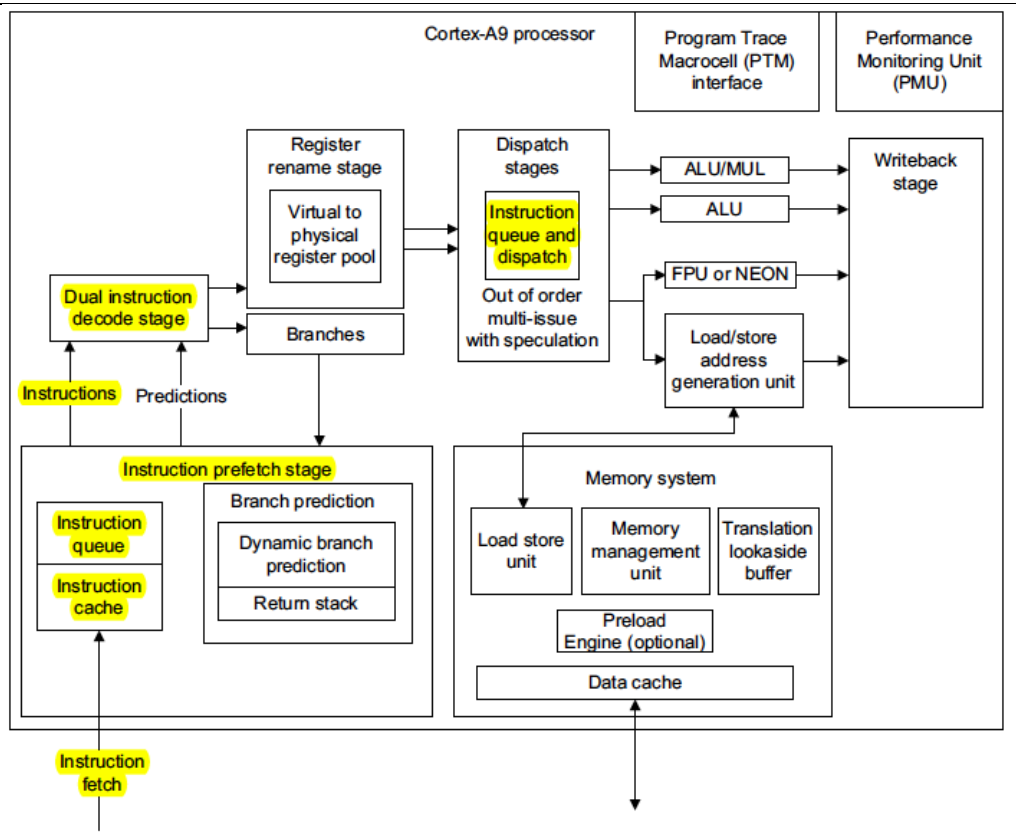


Figure 2-1 Cortex-A9 processor top-level diagram

Cortex-A9 TRM at 2-2 (33) [TPLBN034890].

2.1.2 Instruction queue

In the instruction queue small loop mode provides low power operation while executing small instruction loops. See *Energy efficiency features* on page 2-10.

Cortex-A9 TRM at 2-2 (33) [TPLBN034890]; ARM11 TRM at 11-12 (457) [TPLBN035544].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

	2.1.3 Dynamic branch prediction The Prefetch Unit implements two-level dynamic branch prediction with a *Global History Buffer* (GHB), a *Branch Target Address Cache* (BTAC) and a return stack. See *About the L1 instruction side memory system* on page 7-5. Cortex-A9 TRM at 2-3 (34) [TPLBN034891]; ARM11 TRM at 1-11 (37) [TPLBN035124]. **2.4.1 Energy efficiency features** The features of the Cortex-A9 processor that improve energy efficiency include: • accurate branch and return prediction, reducing the number of incorrect instruction fetch and decode operations • the use of physically addressed caches, reducing the number of cache flushes and refills, saving energy in the system • the use of micro TLBs reduces the power consumed in translation and protection look-ups for each cycle • caches that use sequential access information to reduce the number of accesses to the tag RAMs and to unnecessary accesses to data RAMs • instruction loops that are smaller than 64 bytes often complete without additional instruction cache accesses, so lowering power consumption. Cortex-A9 TRM at 2-10 (41) [TPLBN034898]; ARM11 TRM at 1-23 (49) [TPLBN035136].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

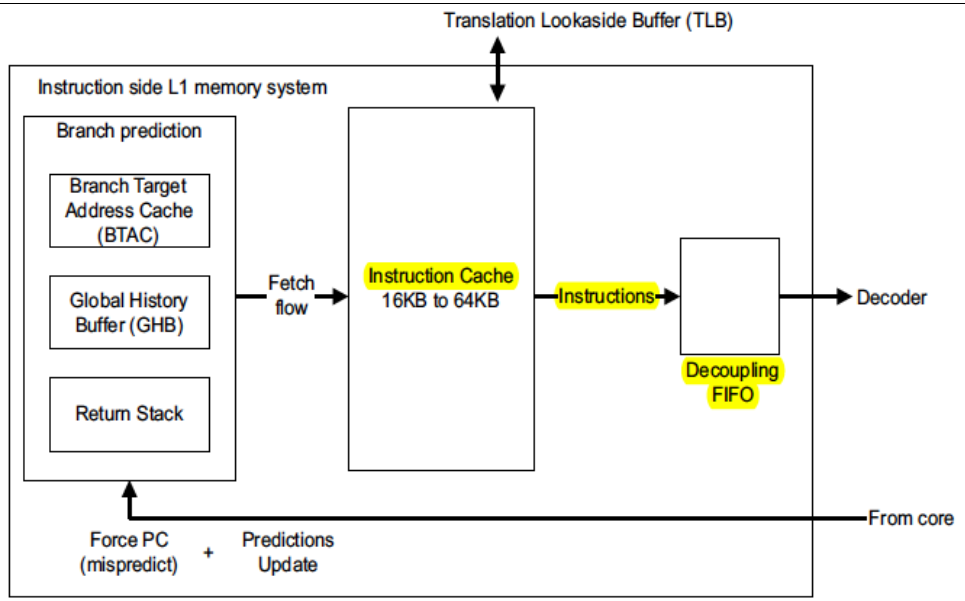


Figure 7-1 Branch prediction and instruction cache

Cortex-A9 TRM at 7-5 (119) [TPLBN034976]; *see also* Cortex-A8 TRM at 1-4 (29) [TPLBN034306]; ARM11 TRM at 1-26 (52) [TPLBN035139].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

Cache features

The Cortex-A9 processor has separate instruction and data caches. The caches have the following features:

- Each cache can be disabled independently. See *System Control Register* on page 4-15.
- Cache replacement policy is either pseudo round-robin or pseudo random.
- Both caches are 4-way set-associative.
- The cache line length is eight words.
- On a cache miss, critical word first filling of the cache is performed.
- You can configure the instruction and data caches independently during implementation to sizes of 16KB, 32KB, or 64KB.
- To reduce power consumption, the number of full cache reads is reduced by taking advantage of the sequential nature of many cache operations. If a cache read is sequential to the previous cache read, and the read is within the same cache line, only the data RAM set that was previously read is accessed.

Instruction cache features

The instruction cache is virtually indexed and physically tagged.

Cortex-A9 TRM at 7-2 (116) [TPLBN034898]; *see* ARM11 TRM at 1-8, 1-11, 1-16 (34, 37, 42) [TPLBN035121, TPLBN035124, TPLBN035129].

Prefetching	In pipelined processors, the process of fetching instructions from memory to fill up the pipeline before the preceding instructions have finished executing. Prefetching an instruction does not mean that the instruction must be executed.

Cortex-A9 TRM at Glossary-12 (226) [TPLBN035083]; ARM11 TRM at Glossary-15 (754) [TPLBN035841].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

A3.9.2 Memory hierarchy

Memory close to a processor has very low latency, but is limited in size and expensive to implement. Further from the processor it is easier to implement larger blocks of memory but these have increased latency. To optimize overall performance, an ARMv7 memory system can include multiple levels of cache in a hierarchical memory system. Figure A3-5 shows such a system, in an ARMv7-A implementation of a VMSA, supporting virtual addressing.

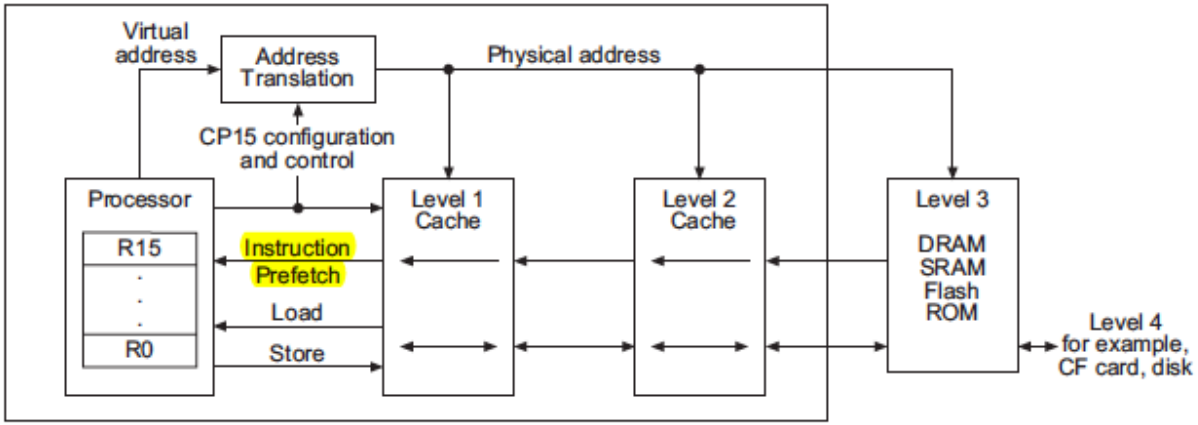


Figure A3-5 Multiple levels of cache in a memory hierarchy

ARMv7 Reference Manual at A3-52 (166) [TPLBN049454], A3-53 – A3-54 (167-68) [TPLBN049455-56]; *see also* ARMv6 Reference Manual at A3-63 (63) [TPLBN035909].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

For implicit accesses:

- Cache linefills and evictions have no effect on the single-copy atomicity of explicit transactions or instruction fetches.
- Instruction fetches are single-copy atomic for each instruction fetched.

Note

32-bit Thumb instructions are fetched as two 16-bit items.

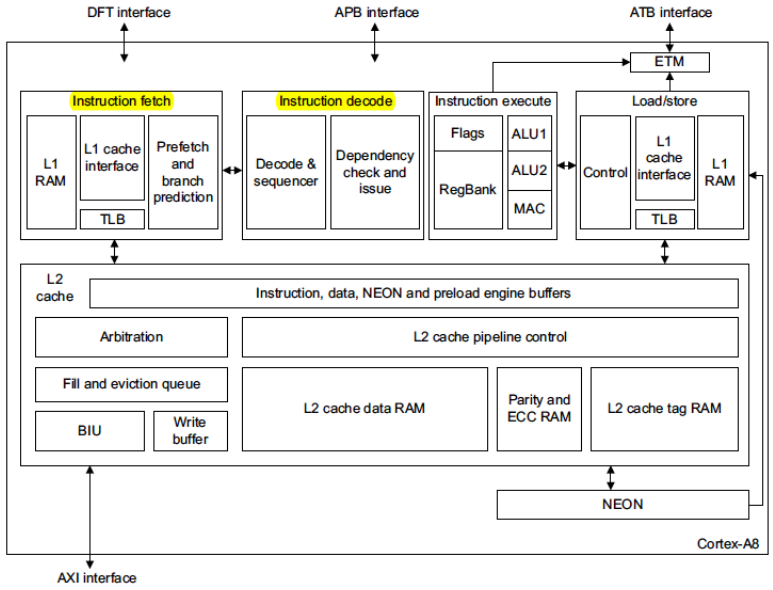
- Translation table walks are performed as 32-bit accesses aligned to 32 bits, each of which is single-copy atomic.

ARMv7 Reference Manual at A3-27 – 3-28 (141-142) [TPLBN049429-30]; *see also* ARMv6 at A3-50 (50) [TPLBN035896].

1.3 Components of the processor

The main components of the processor are:

- *Instruction fetch*
- *Instruction decode* on page 1-5
- *Instruction execute* on page 1-5
- *Load/store* on page 1-5
- *L2 cache* on page 1-5
- *NEON* on page 1-6
- *ETM* on page 1-6.



Cortex-A8 TRM at 1-4 (29) [TPLBN034306].

	1.3.1 Instruction fetch The instruction fetch unit predicts the instruction stream, fetches instructions from the L1 instruction cache, and places the fetched instructions into a buffer for consumption by the decode pipeline. The instruction fetch unit also includes the L1 instruction cache. Cortex-A8 TRM at 1-4 (29) [TPLBN034306]. **1.3.2 Instruction decode** The instruction decode unit decodes and sequences all ARM and Thumb-2 instructions including debug control coprocessor, CP14, instructions and system control coprocessor, CP15, instructions. See Chapter 12 *Debug* for information on the CP14 coprocessor and Chapter 3 *System Control Coprocessor* for information on the CP15 coprocessor. The instruction decode unit handles the sequencing of: • exceptions • debug events • reset initialization • *Memory Built-In Self Test* (MBIST) • wait-for-interrupt • other unusual events. Cortex-A8 TRM at 1-5 (30) [TPLBN034307].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

	For ARMv6-M, instruction fetches are always halfword-aligned and data accesses are always naturally aligned. Address calculations are normally performed using ordinary integer instructions. This means that they wrap around if they overflow or underflow the address space. Another way of describing this is that any address calculation is reduced modulo 2^{32}. Normal sequential execution of instructions effectively calculates: `(address_of_current_instruction) + (size_of_executed_instruction)` after each instruction to determine the instruction to execute next. If this calculation overflows the top of the address space, the result is UNPREDICTABLE. In ARMv6-M this condition cannot occur because the top of memory is defined to always have the *eXecute Never* (XN) memory attribute associated with it. See *The system address map* on page B3-258 for more information. An access violation is reported if this scenario occurs. The information in this section only applies to instructions that are executed, including those that fail their condition code check. Most ARM implementations prefetch instructions ahead of the currently-executing instruction. ARMv6 Reference Manual at A3-42 (42) [TPLBN035888].
[1L] 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	On information and belief, in each Accused Product, the means for fetching instructions are 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. As discussed above in connection with element 1e, the means for fetching instructions is 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. *See* discussion and evidence above. *See* element 1e, above. In addition, the single memory cycle during which it is supplied to the CPU is the same memory cycle in which it is fetched. *See, e.g.*:

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

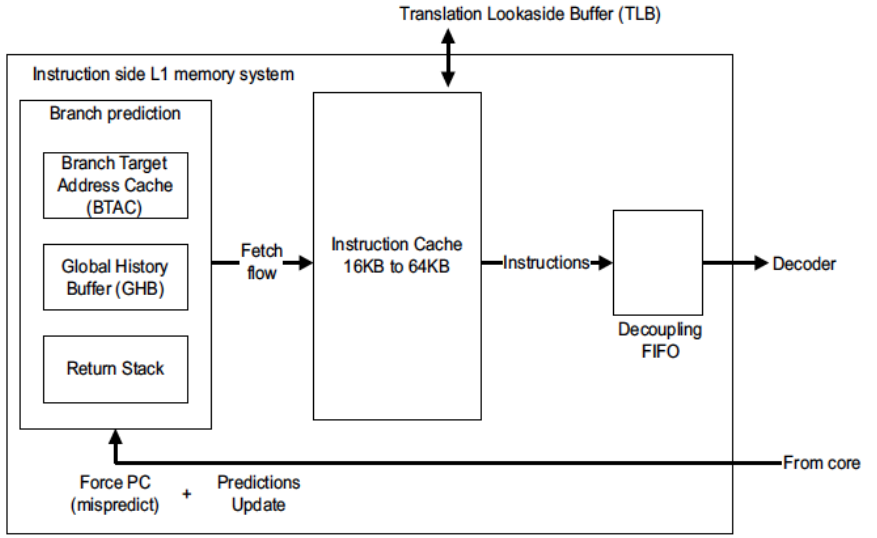
central processing unit integrated circuit during a single memory cycle comprises supplying the multiple sequential instructions in parallel to said instruction register during the same memory cycle in which the multiple sequential instructions are fetched.	**Instruction Cache Controller** The instruction cache controller fetches the instructions from memory depending on the program flow predicted by the prefetch unit. 15.1.4.1 GPMC Features The GPMC is the external memory controller of the device. The GPMC data access engine provides a flexible programming model for communication with all standard memories. The GPMC supports various accesses: • Asynchronous read/write access • Asynchronous read page access (4, 8, and 16 Word16) Cortex-A9 TRM at 7-5 – 7-6 (119-20) [TPLBN034976-77].  **Figure 7-1 Branch prediction and instruction cache** Cortex-A9 TRM at 7-5 (119) [TPLBN034976]

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

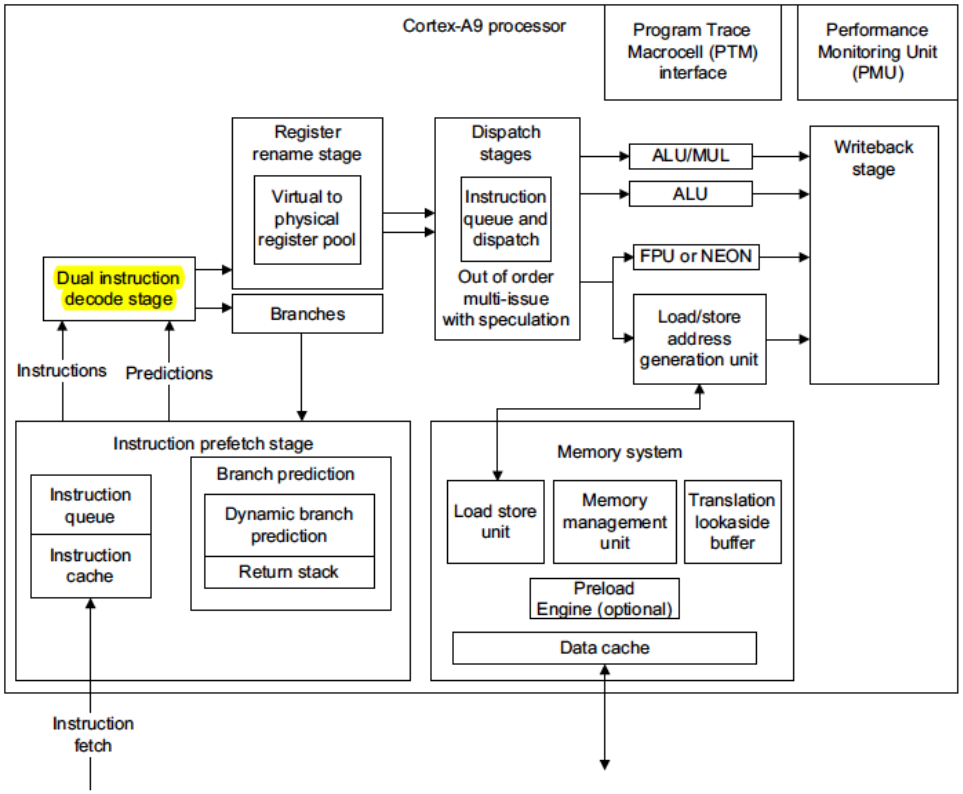


Figure 2-1 Cortex-A9 processor top-level diagram

Cortex-A9 TRM at 2-2 (33) [TPLBN034890].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

Instruction Cache Controller

The instruction cache controller fetches the instructions from memory depending on the program flow predicted by the prefetch unit.

15.1.4.1 GPMC Features

The GPMC is the external memory controller of the device. The GPMC data access engine provides a flexible programming model for communication with all standard memories. The GPMC supports various accesses:

- Asynchronous read/write access
- Asynchronous read page access (4, 8, and 16 Word16)

Cortex-A9 TRM at 7-5 – 7-6 (119-20) [TPLBN034976-77].

7.2 Cache organization

Each cache is implemented as a four-way set associative cache of configurable size. The caches are virtually indexed and physically tagged. You can configure the cache sizes in the range of 4 to 64KB. Both the Instruction Cache and the Data Cache can provide two words per cycle for all requesting sources.

ARM11 TRM at 7-3 (374) [TPLBN035461]; Cortex-A8 TRM at 7-3 (244) [TPLBN034521].

Execution stream

The stream of instructions that would have been executed by sequential execution of the program.

ARMv7 Reference Manual at Glossary-5 (2149) [TPLBN051437]; ARMv6 Reference Manual at Glossary-428 (428) [TPLBN036274].

See also Cortex-A8 TRM at 16-13 (504) [TPLBN034781]; Cortex-A9 TRM at B-1 – B-9 (198-207) [TPLBN035055-TPLBN035065]; Cortex-A8 TRM at 2-3 (43) [TPLBN034320].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

Claim 9 (not asserted)[3]	
[9 preamble] A microprocessor system, comprising	On information and belief, each Accused Product contains one of the Accused Microprocessors listed below. *See* claim chart for claim 1, above.
[9a] a central processing unit,	On information and belief, each Accused Product contains one of the Accused Microprocessors, each of which contains one or more central processing units. *See* element 1a, above.
[9b] a dynamic random access memory,	On information and belief, each Accused Product has a dynamic random access memory. *See* element 1b, above.
[9c] a bus connecting said central processing unit to said dynamic random access memory, and	On information and belief, each Accused Product has a bus connecting said central processing unit to said dynamic random access memory. *See* element 1c, above.
[9d] multiplexing means on said bus	On information and belief, each Accused Product has multiplexing means on said bus between said central processing unit and said dynamic random access memory.

[3] Claim 9 is not asserted, but is included in this chart because it is the independent claim from which dependent claim 59 depends.

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

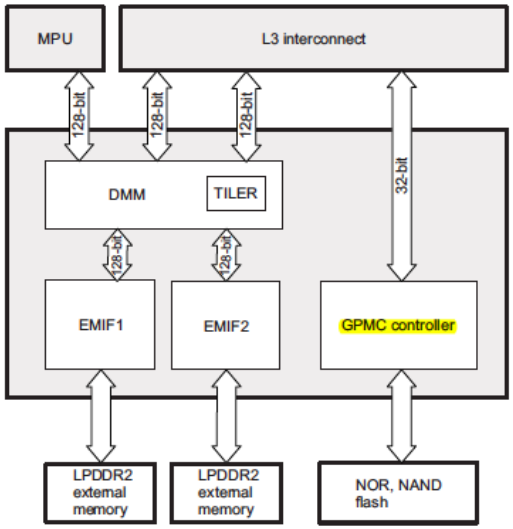
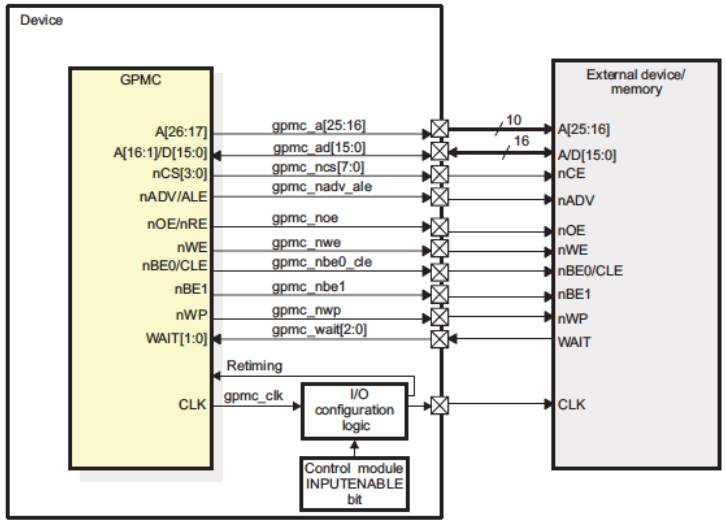
between said central processing unit and said dynamic random access memory,	*See* elements 1c and 1d, above. As to Accused Products with ARM processors, *see, e.g.*: The OMAP4430 supports multiplexed memory using a multiplexing means connected to the bus between the CPU and the DRAM. **Figure 15-1. Memory Subsystem Functional Diagram**  **Figure 15-50. GPMC to 16-Bit Address/Data-Multiplexed Memory** 

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

15.1.4.1 GPMC Features

The GPMC is the external memory controller of the device. The GPMC data access engine provides a flexible programming model for communication with all standard memories. The GPMC supports various accesses:

- Asynchronous read/write access
- Asynchronous read page access (4, 8, and 16 Word16)
- Synchronous read/write access
- Synchronous read/write burst access without wrap capability (4, 8, and 16 Word16)
- Synchronous read/write burst access with wrap capability (4, 8, and 16 Word16)
- Address/data-multiplexed access
- Little- and big-endian access

OMAP4430 TRM at 3207 [TPLBN011706], 3209-10 [TPLBN011708-10], 3366 [TPLBN011865]; *see also* 3368 [TPLBN011865]; 3376 [TPLBN011875].

15.4.4 GPMC Functional Description

The GPMC basic programming model offers maximum flexibility to support various access protocols for each of the eight configurable chip-selects. Use optimal chip-select settings, based on the characteristics of the external device:

- Different protocols can be selected to support generic asynchronous or synchronous random-access devices (NOR flash, SRAM) or to support specific NAND devices.
- The address and the data bus can be multiplexed on the same external bus.
- Read and write access can be independently defined as asynchronous or synchronous.
- System requests (byte, 16-bit word, burst) are performed through single or multiple accesses. External access profiles (single, multiple with optimized burst length, native- or emulated-wrap) are based on external device characteristics (supported protocol, bus width, data buffer size, native-wrap support).
- System burst read or write requests are synchronous-burst (multiple-read or multiple-write). When neither burst nor page mode is supported by external memory or ASIC devices, system burst read or write requests are translated to successive single synchronous or asynchronous accesses (single reads or single writes). 8-bit wide devices are supported only in single synchronous or single asynchronous read or write mode.
- To simulate a programmable internal-wait state, an external wait pin can be monitored to dynamically control external access at the beginning (initial access time) of and during a burst access.

OMAP4430 TRM at 3372 [TPLBN011871].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

	15.4.4.8 GPMC Address and Data Bus The current application supports GPMC connection to NAND devices and to address/data-multiplexed memories or devices. Connection to address/data-nonmultiplexed memories or devices is supported with a limited address range of 2 Kbytes. Depending on the GPMC configuration of each chip-select, address and data bus lines that are not required for a particular access protocol are not updated (changed from current value) and are not sampled when input (input data bus). • When the GPMC_CONFIG[1] LIMITEDADDRESS bit is set to 1, A26-A11 are not modified during an external memory access. This limits the memory address space to 2K-byte regardless of the memory size. • For address/data-multiplexed and AAD-multiplexed NOR devices, the address is multiplexed on the data bus. • 8-bit wide NOR devices do not use GPMC I/O: gpmc_ad[15:8] for data (they are used for address if needed). • 16-bit wide NAND devices do not use GPMC I/O: gpmc_a[25:16]. • 8-bit wide NAND devices do not use GPMC I/O: gpmc_a[25:16] and GPMC I/O: gpmc_ad[15:8]. OMAP4430 TRM at 3376 [TPLBN011875]. ***15.4.4.9.2.3 Address/Data-Multiplexing Interface*** For random synchronous or asynchronous memory interfacing (DEVICETYPE = 0b00), an address- and data-multiplexing protocol can be selected through the GPMC_CONFIG1_i[9:8] MUXADDDATA bit field (i = 0 to 7). The nADV signal must be used as the external device address latch control signal. For the associated chip-select configuration, nADV assertion and deassertion time and nOE assertion time must be set to the appropriate value to meet the address latch setup/hold time requirements of the external device. See Section 15.4.3, *GPMC Integration*. OMAP4430 TRM at 3380 [TPLBN011879].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

	15.3.1.1 EMIF Module Main Features The EMIF module supports the following features: ▪ JEDEC standard compliant LPDDR2-SDRAM (S2 and S4) and LPDDR2-NVM devices. ▪ Two GB SDRAM address range over two chip-selects (1GB per CS) (configurable with the DMM module, see Section 15.2, *Dynamic Memory Manager* for more information) ▪ Two independent chip-selects with their corresponding register sets and independent page tracking ▪ Both chip-selects must have the same memory type and size if they are both SDRAM or both NVM. LPDDR2-SDRAM can be used in parallel with LPDDR2-NVM and can have a different size. ▪ Flexible address muxing scheme which permit to choose different bank mapping allocation by configuring the bank, column and row address decoding ordering ▪ 16 or -32-bit data path to external SDRAM memory ▪ LPDDR2 devices with 1, 2. 4 or 8 of internal banks ▪ Data bus widths: – 128-bit L3 interconnect Data Bus Width – 16 and 32-bit SDRAM Data Bus Width ▪ CAS latencies: 3, 4, 5, 6, 7 and 8 ▪ 256, 512, 1024, and 2048-word page sizes ▪ Burst lenghts: 8 ▪ Sequential burst type ▪ SDRAM auto initialization from reset or configuration change ▪ Bank Interleaving across both the chip-selects if same memory type OMAP4430 TRM at 3292 [TPLBN011791]. On information and belief, the DDR memory components shown in the OMAP4430 Accused Products requires a multiplexing means connected to the bus between the memory and the CPU. *See, e.g.*, http://www.jedec.org; *see also* JESD209-2E, available online at http://www.jedec.org/sites/default/files/docs/JESD209-2E.pdf at 3; JEDEC Standard No. 21-C, available online at http://www.jedec.org/sites/default/files/2_00R20.pdf.
[9e] said multiplexing means being connected and configured to	On information and belief, each Accused Product has multiplexing means connected and configured to provide multiplexed row addresses, column addresses and data on said bus from said central processing unit to said dynamic random access memory and to provide data from said dynamic random access memory to said central processing unit. *See* element 9d, above.

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

provide multiplexed row addresses, column addresses and data on said bus from said central processing unit to said dynamic random access memory and to provide data from said dynamic random access memory to said central processing unit, and	The multiplexing means multiplexes row addresses, column addresses, and data on the bus from the CPU to the DRAM and to provide the data from the DRAM. The controllers described in connection with element 9d above are all connected to the bus system for the various microprocessor chips in the Accused Products, and are all capable of passing multiplexed information between the CPU and memory. As shown in the documents cited above in connection with element 9d, the multiplexed information includes multiplexing of row/column addresses and data. **15.4.4.8 GPMC Address and Data Bus** The current application supports GPMC connection to NAND devices and to address/data-multiplexed memories or devices. Connection to address/data-nonmultiplexed memories or devices is supported with a limited address range of 2 Kbytes. Depending on the GPMC configuration of each chip-select, address and data bus lines that are not required for a particular access protocol are not updated (changed from current value) and are not sampled when input (input data bus). • When the GPMC_CONFIG[1] LIMITEDADDRESS bit is set to 1, A26-A11 are not modified during an external memory access. This limits the memory address space to 2K-byte regardless of the memory size. • For address/data-multiplexed and AAD-multiplexed NOR devices, the address is multiplexed on the data bus. • 8-bit wide NOR devices do not use GPMC I/O: gpmc_ad[15:8] for data (they are used for address if needed). • 16-bit wide NAND devices do not use GPMC I/O: gpmc_a[25:16]. • 8-bit wide NAND devices do not use GPMC I/O: gpmc_a[25:16] and GPMC I/O: gpmc_ad[15:8]. OMAP4430 TRM at 3376 [TPLBN011875].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

	15.4.4 GPMC Functional Description The GPMC basic programming model offers maximum flexibility to support various access protocols for each of the eight configurable chip-selects. Use optimal chip-select settings, based on the characteristics of the external device: • Different protocols can be selected to support generic asynchronous or synchronous random-access devices (NOR flash, SRAM) or to support specific NAND devices. • The address and the data bus can be multiplexed on the same external bus. • Read and write access can be independently defined as asynchronous or synchronous. • System requests (byte, 16-bit word, burst) are performed through single or multiple accesses. External access profiles (single, multiple with optimized burst length, native- or emulated-wrap) are based on external device characteristics (supported protocol, bus width, data buffer size, native-wrap support). • System burst read or write requests are synchronous-burst (multiple-read or multiple-write). When neither burst nor page mode is supported by external memory or ASIC devices, system burst read or write requests are translated to successive single synchronous or asynchronous accesses (single reads or single writes). 8-bit wide devices are supported only in single synchronous or single asynchronous read or write mode. • To simulate a programmable internal-wait state, an external wait pin can be monitored to dynamically control external access at the beginning (initial access time) of and during a burst access. OMAP4430 TRM at 3372 [TPLBN011871].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

3.2.55 c9, L2 Cache Auxiliary Control Register

The purpose of the L2 Cache Auxiliary Control Register is to enable you to configure the L2 cache behavior.

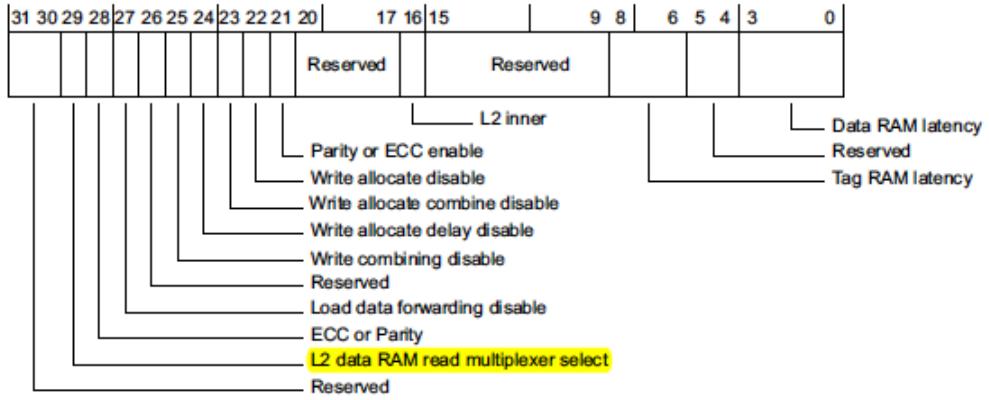
The L2 Cache Auxiliary Control Register is:

- a read register, common to Secure and Nonsecure states
- a write register in Secure state only
- accessible in privileged modes only.

Note

If bit [24] of the L2 Cache Auxiliary Control Register is not set to 1, the L2 cache does not perform a security check for the data that is placed in the L2 cache. If software requires a higher level of security within the processor, then you must set bit [24] to 1 of this register. By setting bit [24] to 1, the L2 cache performs an external linefill, and the AXI slave performs the security check on that linefill.

Figure 3-49 shows the bit arrangement of the L2 Cache Auxiliary Control Register.



Cortex A8 TRM at 3-95 (173) [TPLBN034450].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

	<table><tr><th>Bits</th><th>Field</th><th>Function</th></tr><tr><td>[31:30]</td><td>-</td><td>Reserved. UNP, SBZP.</td></tr><tr><td>[29]</td><td>L2 data RAM read multiplexer select</td><td>Configures the timing of the read data multiplexer select between one or two cycles for all L2 data RAM read operations: 0 = two cycles, default 1 = one cycle.</td></tr></table> Cortex A8 TRM at 3-95 (173) [TPLBN034450]. In addition, on information and belief, the additional discovery will show that each of the memory devices in the accused products sends and receives information that is multiplexed to and from the CPU by way of a multiplexing means connected to the bus that multiplexes row/column addresses and data.
[9f] means connected to said bus for fetching instructions for said central processing unit on said bus from said dynamic random access memory,	On information and belief, each Accused Product has multiplexing means connected to said bus for fetching instructions for said central processing unit on said bus from said dynamic random access memory. *See* element 1d, above.
[9g] said means for fetching instructions being configured to fetch	On information and belief, in each Accused Product, said means for fetching instructions is configured to fetch multiple sequential instructions from said dynamic random access memory in parallel and supply the multiple instructions to said central processing unit during a single memory cycle. *See* element 1e, above.

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

multiple sequential instructions from said dynamic random access memory in parallel and supply the multiple instructions to said central processing unit during a single memory cycle,	
[9h] said central processing unit including an arithmetic logic unit and a first push down stack connected to said arithmetic logic unit,	On information and belief, in each Accused Product, said central processing unit includes an arithmetic logic unit and a first push down stack connected to said arithmetic logic unit. *See* element 1g, above.
[9i] 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	On information and belief, each Accused Product has a 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. *See* elements 1h and 1i, above.

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

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,	
[9j] 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,	On information and belief, in each Accused Product, a remainder of said first push down stack is 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. *See* elements 1h and 1i, above.
[9k] said arithmetic logic unit having an output connected to said means for storing a top item.	On information and belief, each Accused Product has an arithmetic logic unit having an output connected to said means for storing a top item. *See* elements 1g, 1h, 1i, and 1j, above.

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

Claim 43	
[43 preamble] The microprocessor system of claim 1	As shown in connection with claim 1, On information and belief, each Accused Product contains one of the Accused Microprocessors and includes all elements of claim 1. *See* claim chart for claim 1, above.
[43a] wherein said central processing unit integrated circuit a program counter comprising address bits,	On information and belief, each Accused Product includes a central processing unit integrated circuit a program counter comprising address bits. As discussed above in connection with claim 1, On information and belief, each Accused Product includes a central processing unit integrated circuit. *See* element 1a, above. The Accused Products include a program counter (PC):

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

	7.1.2 Features The dual Cortex-M3 MPU subsystem integrates the following: • Two ARM Cortex-M3 microprocessors: – ARMv7-M and Thumb®-2 instruction set architecture – Hardware division and single-cycle multiplication computational acceleration – Integrated nested vector interrupt controller (NVIC) – Integrated bus matrix – Registers: ▪ Thirteen general-purpose 32-bit registers ▪ Link register (LR) ▪ Program counter (PC) ▪ Program status register, xPSR ▪ Two banked SP registers – Integrated power management – Extensive debug capabilities • Shared cache interface: – Instruction and data interface – Supports paralleled accesses OMAP4430 TRM at 1457 [TPLBN009956].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

2.13.1 The state register set

In ARM state, 16 data registers and one or two status registers are accessible at any time. In privileged modes, mode-specific banked registers become available. Figure 2-10 on page 2-19 shows the registers that are available in each mode.

Thumb and ThumbEE state give access to the same set of registers as ARM state. However, the 16-bit instructions provide only limited access to some of the registers. No such limitations exist for 32-bit Thumb-2 and ThumbEE instructions.

Registers r0 through r13 are general-purpose registers used to hold either data or address values.

Registers r14 and r15 have the following special functions:

Link Register Register r14 is used as the subroutine *Link Register* (LR).

Register r14 receives the return address when the processor executes a *Branch with Link* (BL or BLX) instruction.

You can treat r14 as a general-purpose register at all other times. Similarly, the corresponding banked registers r14_mon, r14_svc, r14_irq, r14_fiq, r14_abt, and r14_und hold the return values when the processor receives interrupts and exceptions, or when it executes the BL or BLX instructions within interrupt or exception routines.

Program Counter Register r15 holds the PC:

- in ARM state, this is word-aligned
- in Thumb state, this is halfword-aligned
- in ThumbEE state, this is halfword-aligned.

Cortex-A8 TRM at 58 [TPLBN034335]; *see also* 59-60 [TPLBN034336-37]; Cortex-A9 TRM at 34 [TPLBN034891]; ARM11 TRM at 91 [TPLBN035177], 93-94 [TPLBN035180-81]; ARM Architecture Reference Manual at A1-3 (25) [PIC00005215], A1-6 (28) [PIC00005218], A2-4 (36) [PIC00005226].

The program counter in the Accused Products stores address bits.

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

2.15.1 Exception entry and exit summary

Table 2-12 summarizes the PC value preserved in the relevant r14 on exception entry and the recommended instruction for exiting the exception handler.

Table 2-12 Exception entry and exit

Exception or entry	Return instruction	Previous state		Notes
		ARM r14_x	Thumb r14_x	
SVC	MOVS PC, R14_svc	PC + 4	PC+2	Where the PC is the address of the SVC, SMC, or Undefined instruction
SMC	MOVS PC, R14_mon	PC + 4	-	
UNDEF	MOVS PC, R14_und	PC + 4	PC+2	
PABT	SUBS PC, R14_abt, #4	PC + 4	PC+4	Where the PC is the address of instruction that had the prefetch abort
FIQ	SUBS PC, R14_fiq, #4	PC + 4	PC+4	Where the PC is the address of the instruction that was not executed because the FIQ or IRQ took priority
IRQ	SUBS PC, R14_irq, #4	PC + 4	PC+4	
DABT	SUBS PC, R14_abt, #8	PC + 8	PC+8	Where the PC is the address of the load or store instruction that generated the data abort
RESET	-	-	-	The value saved in r14_svc on reset is Unpredictable
BKPT	SUBS PC, R14_abt, #4	PC + 4	PC+4	Software breakpoint

Cortex-A8 TRM at 67 [TPLBN034344]; *see also* Cortex-A9 TRM at 34 [TPLBN034891], 58-62 [TPLBN034915-19] (Table 4-1); ARM11 TRM at 110 [TPLBN035197]; ARM Architecture Reference Manual at A2-7 – A2-8 (39-40) [PIC00005229-30].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

LR, the Link Register

Register R14 is used to store the return address from a subroutine. At other times, LR can be used for other purposes.

When a BL or BLX instruction performs a subroutine call, LR is set to the subroutine return address. To perform a subroutine return, copy LR back to the program counter. This is typically done in one of two ways, after entering the subroutine with a BL or BLX instruction:

- Return with a `BX LR` instruction.
- On subroutine entry, store LR to the stack with an instruction of the form:
  `PUSH {<registers>,LR}`
  and use a matching instruction to return:
  `POP {<registers>,PC}`

ThumbEE checks and handler calls use LR in a similar way. For details see Chapter A9 *ThumbEE*.

PC, the Program Counter

Register R15 is the program counter:

- When executing an ARM instruction, PC reads as the address of the current instruction plus 8.
- When executing a Thumb instruction, PC reads as the address of the current instruction plus 4.
- Writing an address to PC causes a branch to that address.

In Thumb code, most instructions cannot access PC.

ARMv7 Reference Manual at A2-11 (43) [TPLBN049331]; *see also* B1-9 (1159) [TPLBN050447], ARMv6 Reference Manual at A2-36 (36) [TPLBN035882], B1-211 (211) [TPLBN036057], B1-216 – B1-217 (216-217) [TPLBN036062-63].

PC Program Counter, see *Use of 0b1111 as a register specifier* on page A5-82 for more information. The PC is loaded with the Reset handler start address on reset. PC is sometimes referred to as R15.

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

	ARMv6 Reference Manual at A2-36 (36) [TPLBN035882]; *see also* ARMv7 Reference Manual at A2-11 (43) [TPLBN049331].
[43b] said fetching means configured to locate the multiple sequential instructions using the address bits from the program counter.	On information and belief, each Accused Product includes a fetching means configured to locate the multiple sequential instructions using the address bits from the program counter. As discussed above in connection with claim 1, On information and belief, each Accused Product includes a means connected to the bus for fetching instructions for the central processing unit integrated circuit on the bus from the memory. *See* element 1d, above. As further discussed above, the means for fetching instructions is configured and connected to fetch multiple sequential instructions from the memory in parallel and supply the multiple sequential instructions to the central processing unit integrated circuit during a single memory cycle. *See* element 1e, above. As further discussed above, the means for fetching instructions is 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 the instruction register during the same memory cycle in which the multiple sequential instructions are fetched. *See* element 1L, above. As discussed above in connection with element 43a, the Accused Products all include a program counter that is stores address bits during operation. *See* element 43a, above. In addition, the documents cited above show that the addresses stored by the program counter are addresses of instructions. The program counters in the Accused Products are used to store the address of the current instruction and used when executing a program/subroutine.

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

LR, the Link Register

Register R14 is used to store the return address from a subroutine. At other times, LR can be used for other purposes.

When a `BL` or `BLX` instruction performs a subroutine call, LR is set to the subroutine return address. To perform a subroutine return, copy LR back to the program counter. This is typically done in one of two ways, after entering the subroutine with a `BL` or `BLX` instruction:

- Return with a `BX LR` instruction.
- On subroutine entry, store LR to the stack with an instruction of the form:
  `PUSH {<registers>,LR}`
  and use a matching instruction to return:
  `POP {<registers>,PC}`

ThumbEE checks and handler calls use LR in a similar way. For details see Chapter A9 *ThumbEE*.

PC, the Program Counter

Register R15 is the program counter:

- When executing an ARM instruction, PC reads as the address of the current instruction plus 8.
- When executing a Thumb instruction, PC reads as the address of the current instruction plus 4.
- Writing an address to PC causes a branch to that address.

In Thumb code, most instructions cannot access PC.

ARMv7 Reference Manual at A2-11 (43) [TPLBN049331]; *see also* B1-9 (1159) [TPLBN050447], ARMv6 Reference Manual at A2-36 (36) [TPLBN035882], B1-211 (211) [TPLBN036057], B1-216 – B1-217 (216-217) [TPLBN036062-63].

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

Claim 59	
[59 preamble] The microprocessor system of claim 9	As shown in connection with claim 9, On information and belief, each Accused Product contains one of the Accused Microprocessors and includes all elements of claim 9. *See* claim chart for claim 9, above.; *see also* claim chart for claim 1, above.
[59a] wherein the microprocessor system comprises an instruction register configured to store the multiple sequential instructions and from which instructions are accessed and decoded; and	On information and belief, each Accused Product includes an instruction register configured to store the multiple sequential instructions and from which instructions are accessed and decoded. As discussed above in connection with claim 1, On information and belief, each Accused Product includes an instruction register configured to store the multiple sequential instructions from which instructions are accessed and decoded. *See* element 1k, above.
[59b] wherein the means for fetching instructions being configured and connected to fetch multiple	On information and belief, each Accused Product includes a means for fetching instructions that is configured and connected to fetch multiple sequential instructions from memory in parallel and supply the multiple sequential instructions to the central processing unit during a single memory cycle comprises supplying the multiple sequential instructions in parallel to the instruction register during the same memory cycle in which the multiple sequential instructions are fetched As discussed above in connection with claim 1, On information and belief, each Accused Product includes the claimed means for fetching. *See* element 1L, above.

EXHIBIT G-3 – CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,440,749 By LG

sequential instructions from said memory in parallel and supply the multiple sequential instructions to the central processing unit 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.	