EXHIBIT 4

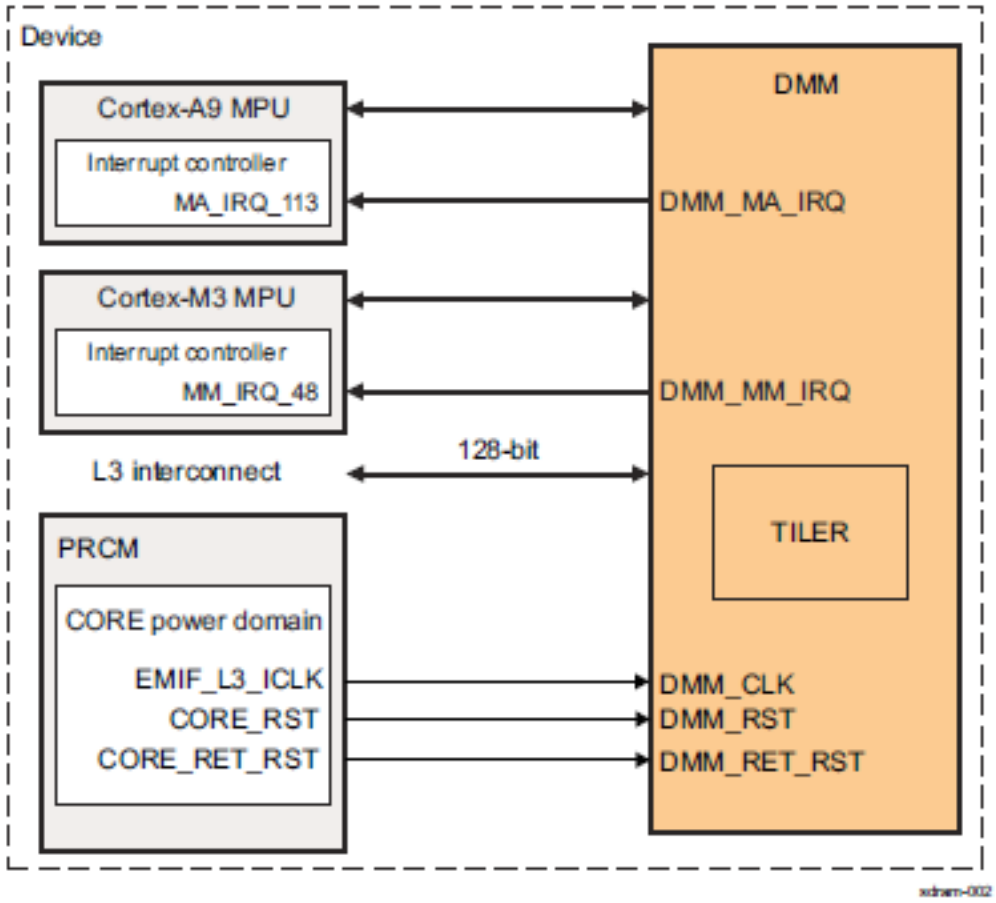
EXHIBIT E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

Claim 1	
A microprocessor, which comprises	On information and belief, each Accused Product listed in the attached list of Accused Products (Ex. A.4) 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/Exynos; 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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

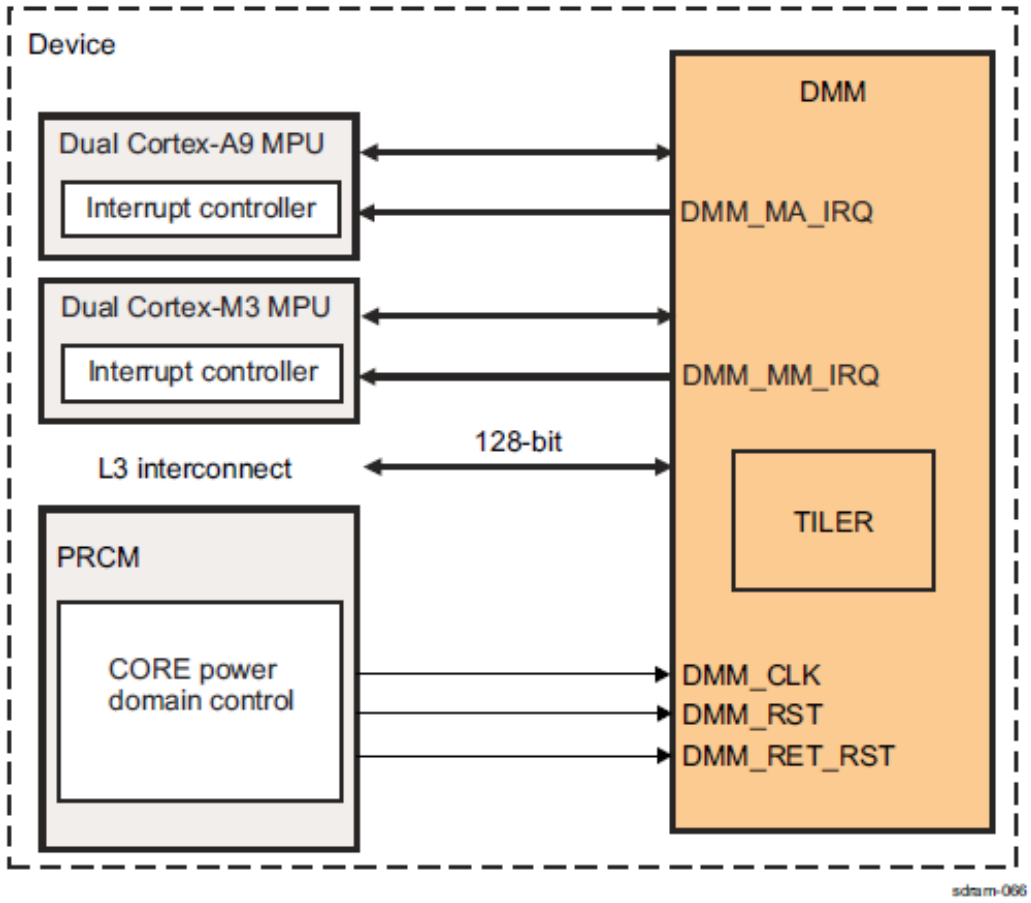
Figure 15-3. DMM Integration



OMAP4430 Technical Reference Manual, p. 3213.

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

Figure 16-2. DMM and Tiler Overview



OMAP4470 Technical Reference Manual, p. 3308.

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

1.3 Block Diagram

Figure 1-1 illustrates the block diagram of Exynos 5250.

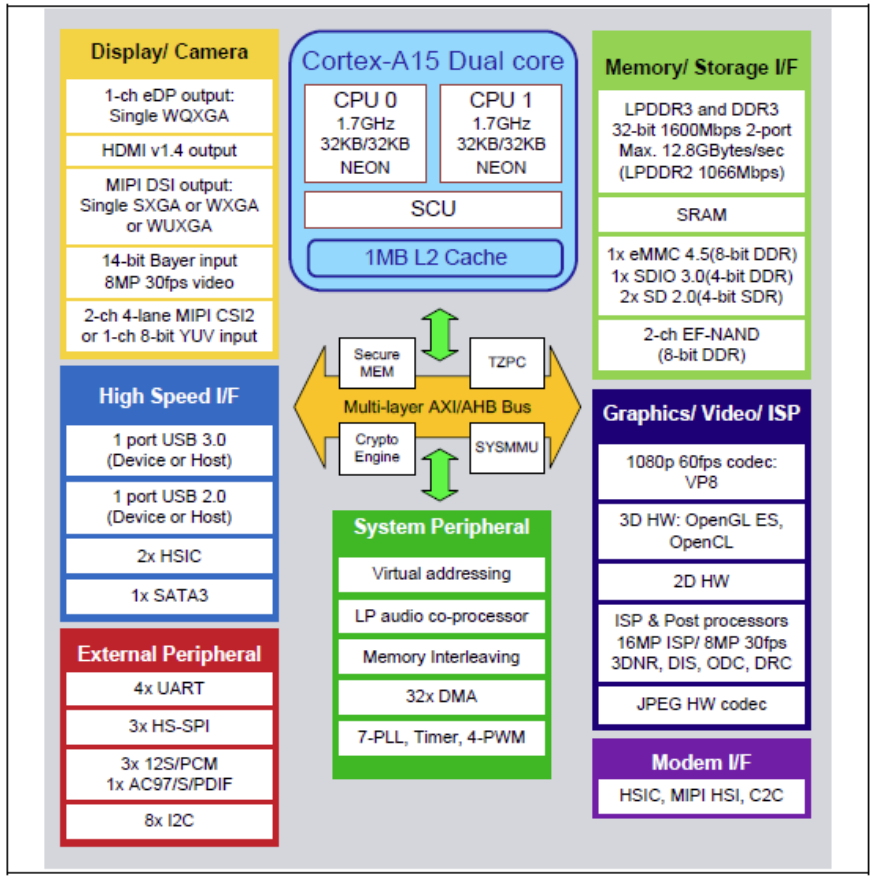


Figure 1-1 Block Diagram of Exynos 5250

Exynos 5 Dual Reference Manual, Fig. 1-1.

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

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

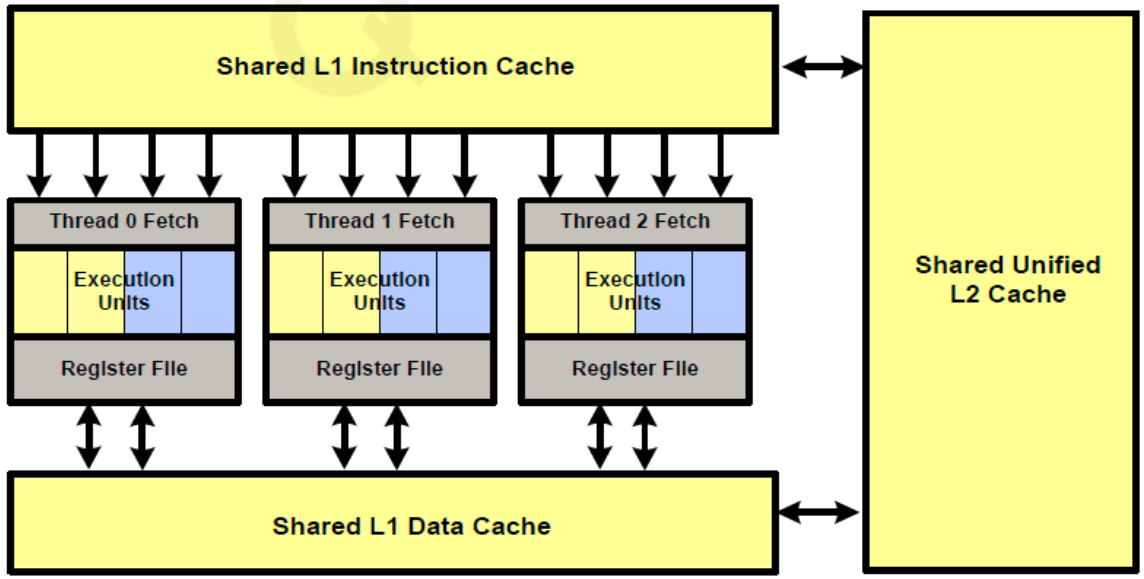
	access memory and fetch and execute instructions separately and independently of the main central processing unit.  Hexagon Programming: Processor Overview, p. 10.

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

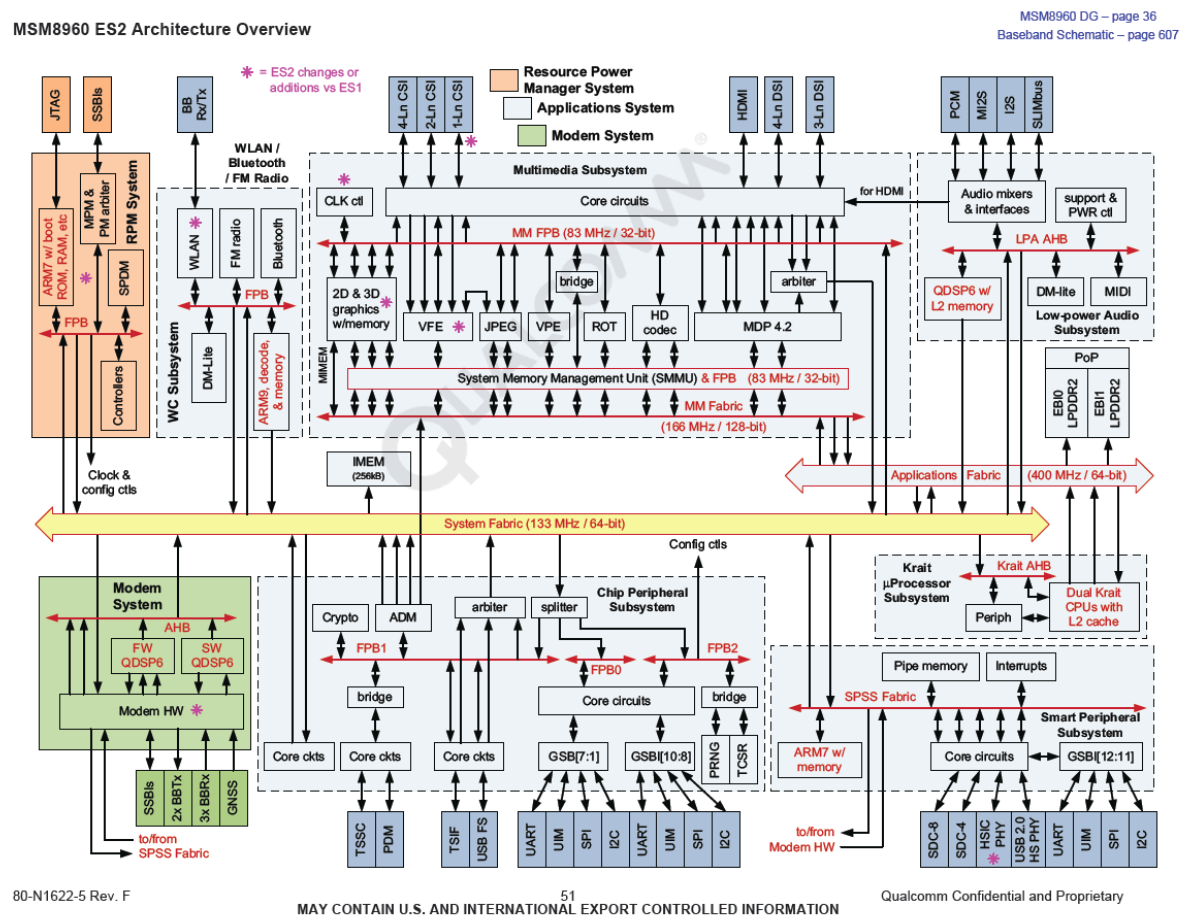
	 MSM8960 Chipset Schematics and Design Guidelines, p. 50. *See also.*, http://en.wikipedia.org/wiki/Qualcomm_Snapdragon; http://en.wikipedia.org/wiki/Exynos; http://en.wikipedia.org/wiki/OMAP.
said main central processing unit	ARM cores, such as those contained within the Accused Microprocessors, contain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

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

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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

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, Accused Microprocessors also include 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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

	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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

	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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

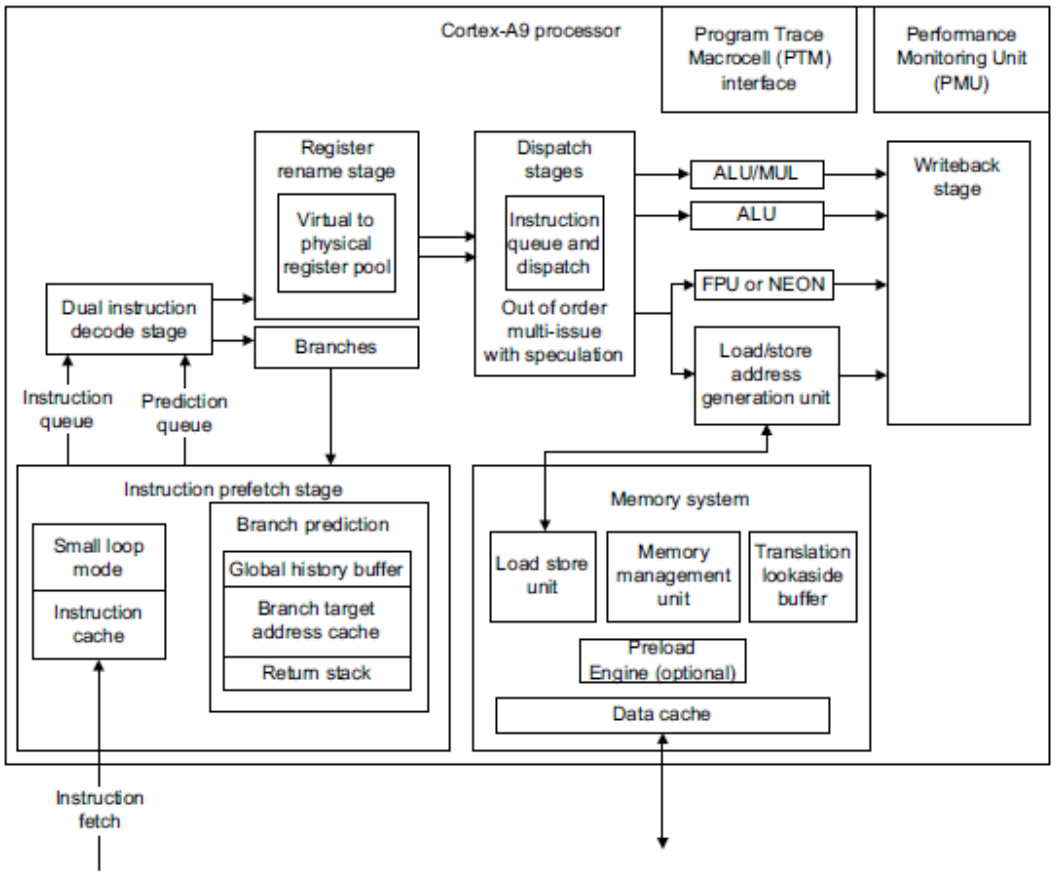
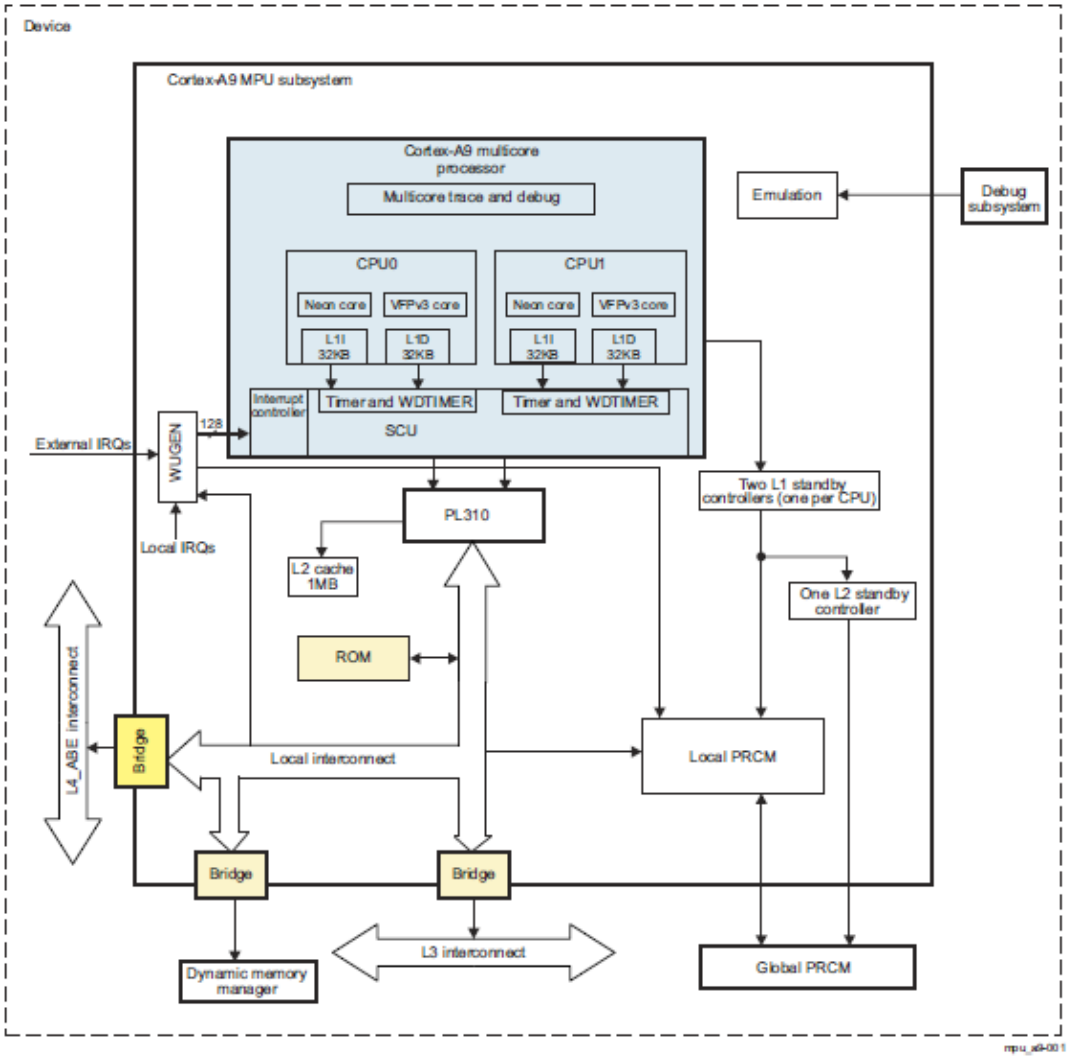


Figure 2-1 Cortex-A9 processor top-level diagram

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

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

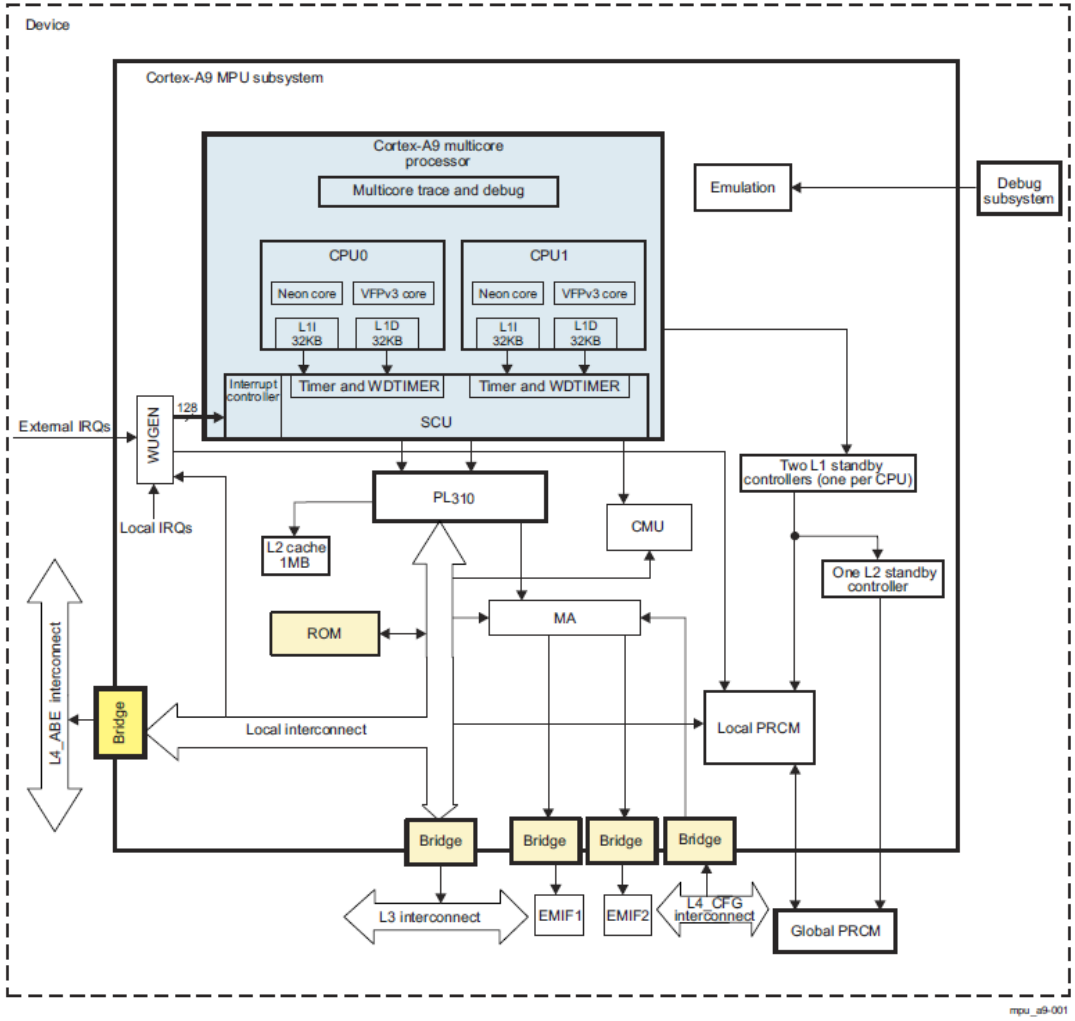
Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4430 Technical Reference Manual, p. 1077.

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

Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4470 Technical Reference Manual, p. 1091.

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

1.3 Block Diagram

Figure 1-1 illustrates the block diagram of Exynos 5250.

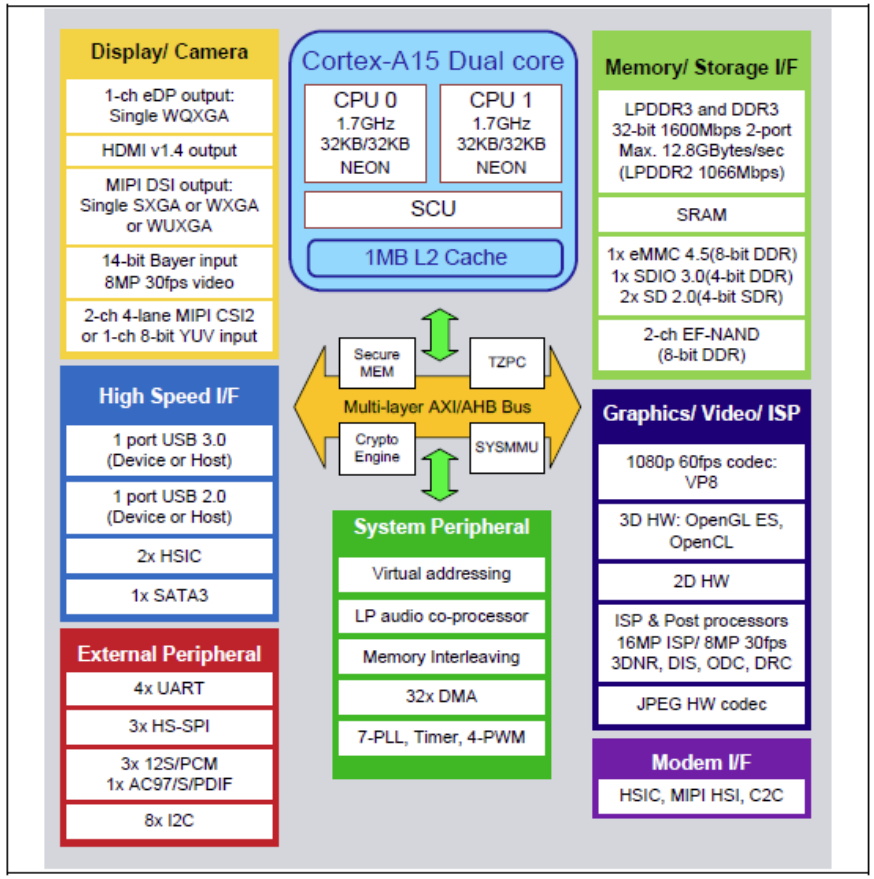
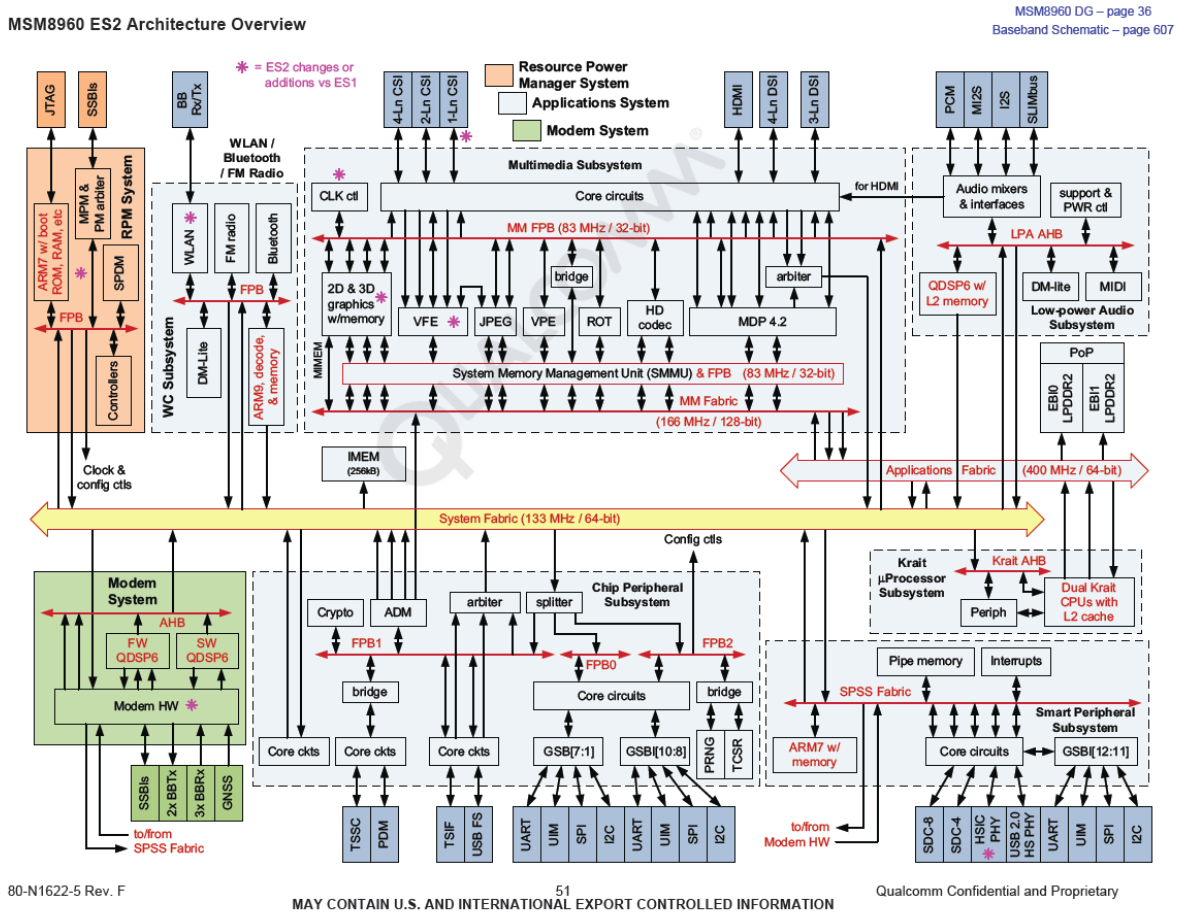


Figure 1-1 Block Diagram of Exynos 5250

Exynos 5 Dual Reference Manual, Fig. 1-1.

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



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 OMAP4430's or OMAP4470's L3 interconnect, the MPU local interconnect, and other internal buses.

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

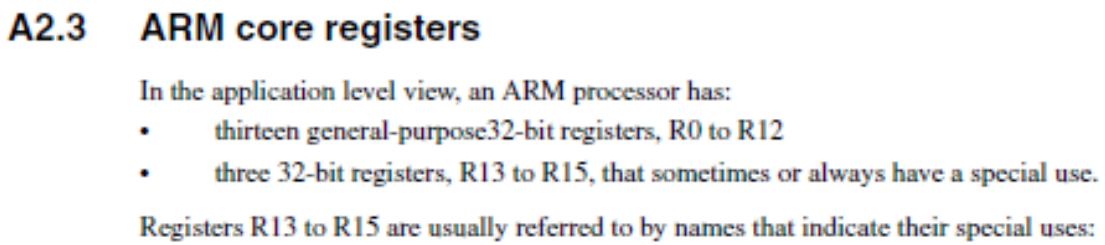
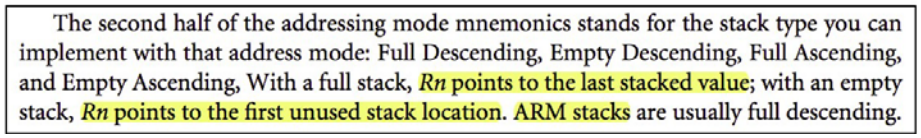
	Plaintiffs contend that this claim element is literally present as described above. In the event that this claim element is not found to be literally present, Plaintiffs contend that the above identified buses, either alone or in combination, are equivalent to "an internal data bus" and any differences 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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

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

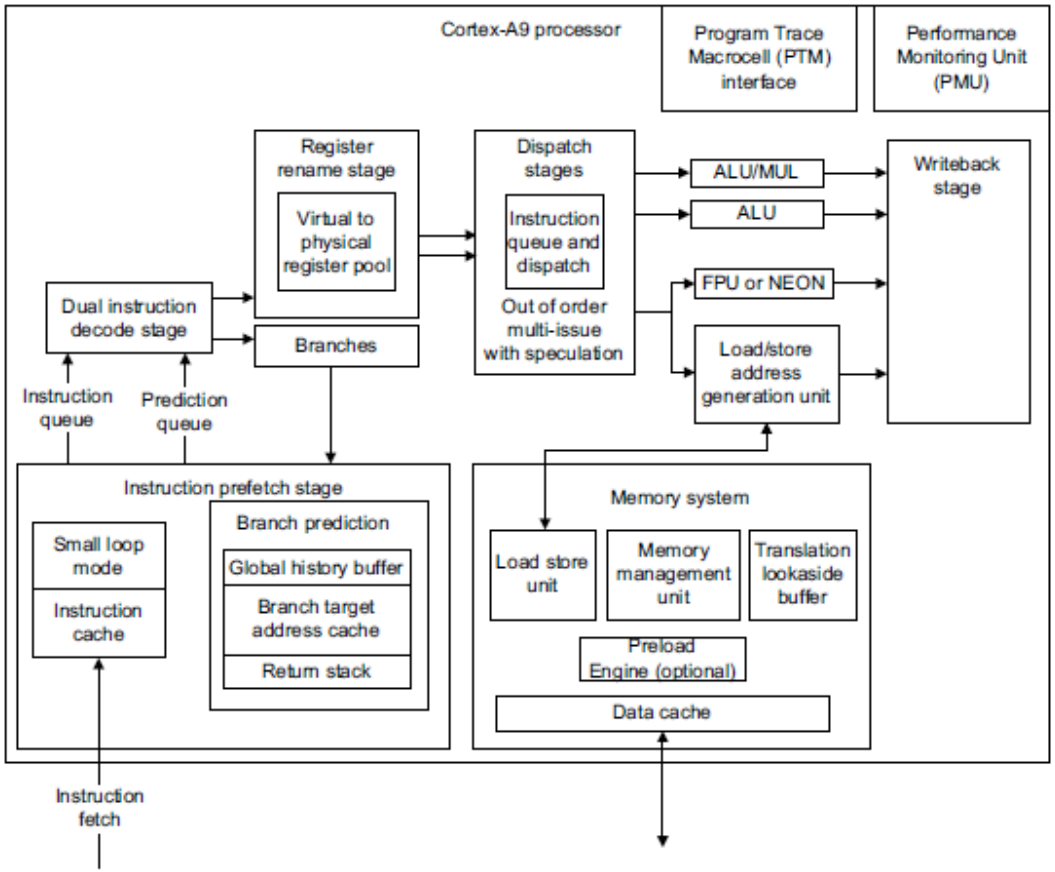
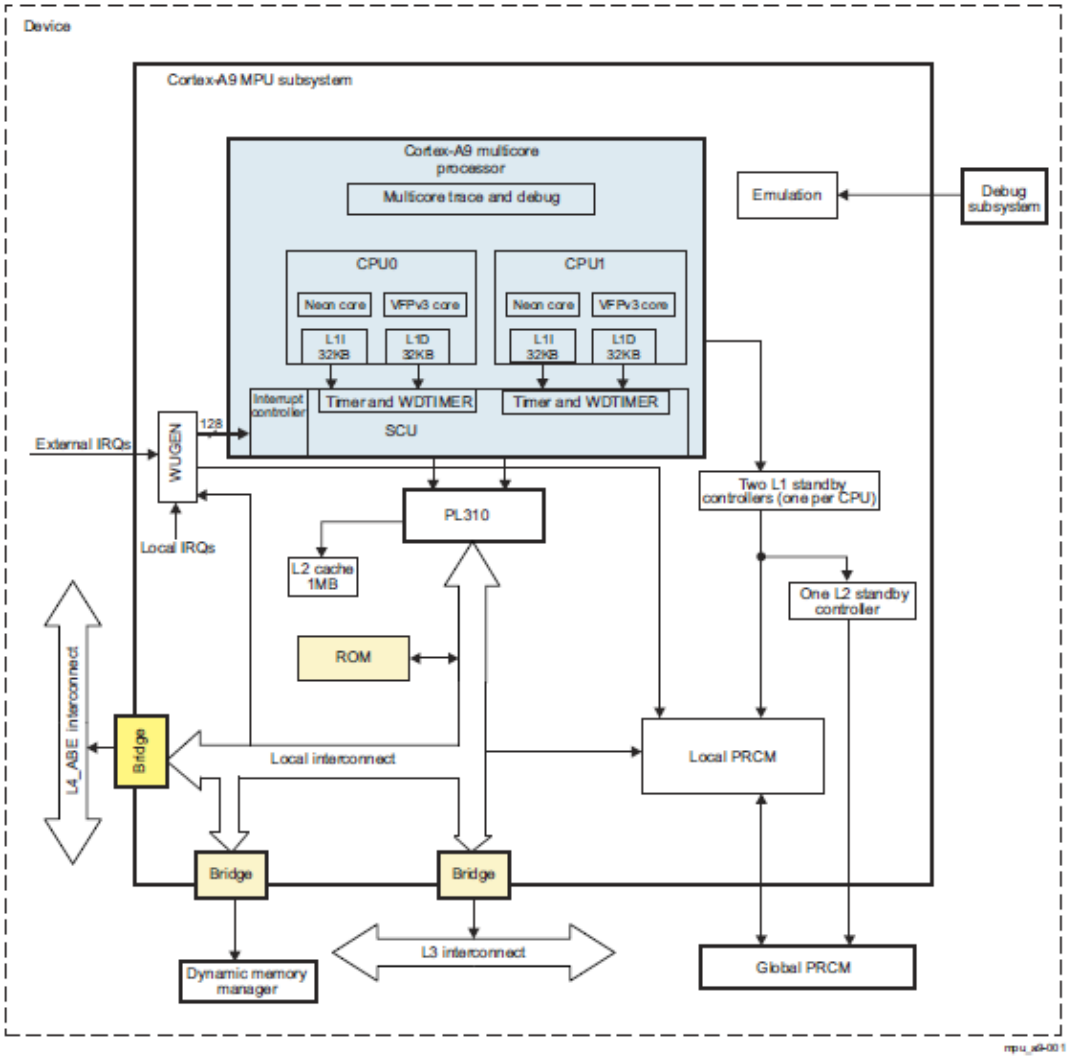


Figure 2-1 Cortex-A9 processor top-level diagram

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

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

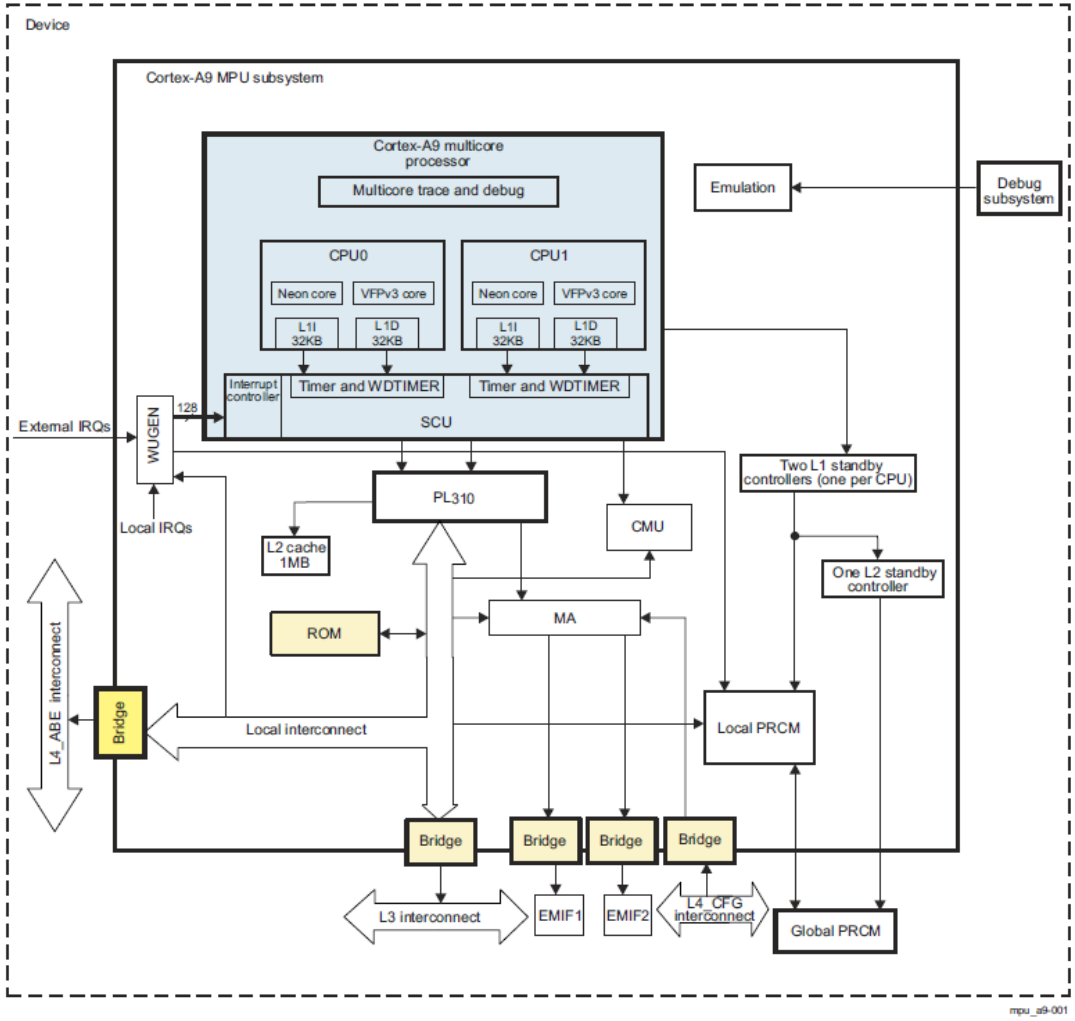
Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4430 Technical Reference Manual, p. 1077.

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

Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4470 Technical Reference Manual, p. 1091.

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

1.3 Block Diagram

Figure 1-1 illustrates the block diagram of Exynos 5250.

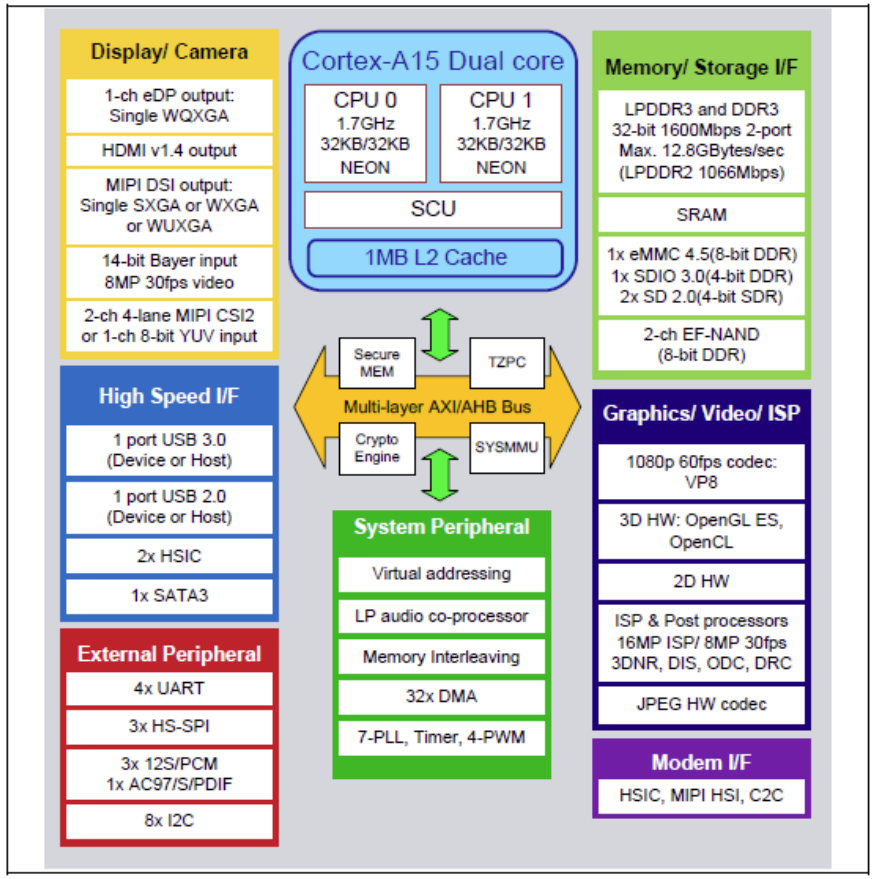


Figure 1-1 Block Diagram of Exynos 5250

Exynos 5 Dual Reference Manual, Fig. 1-1.

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

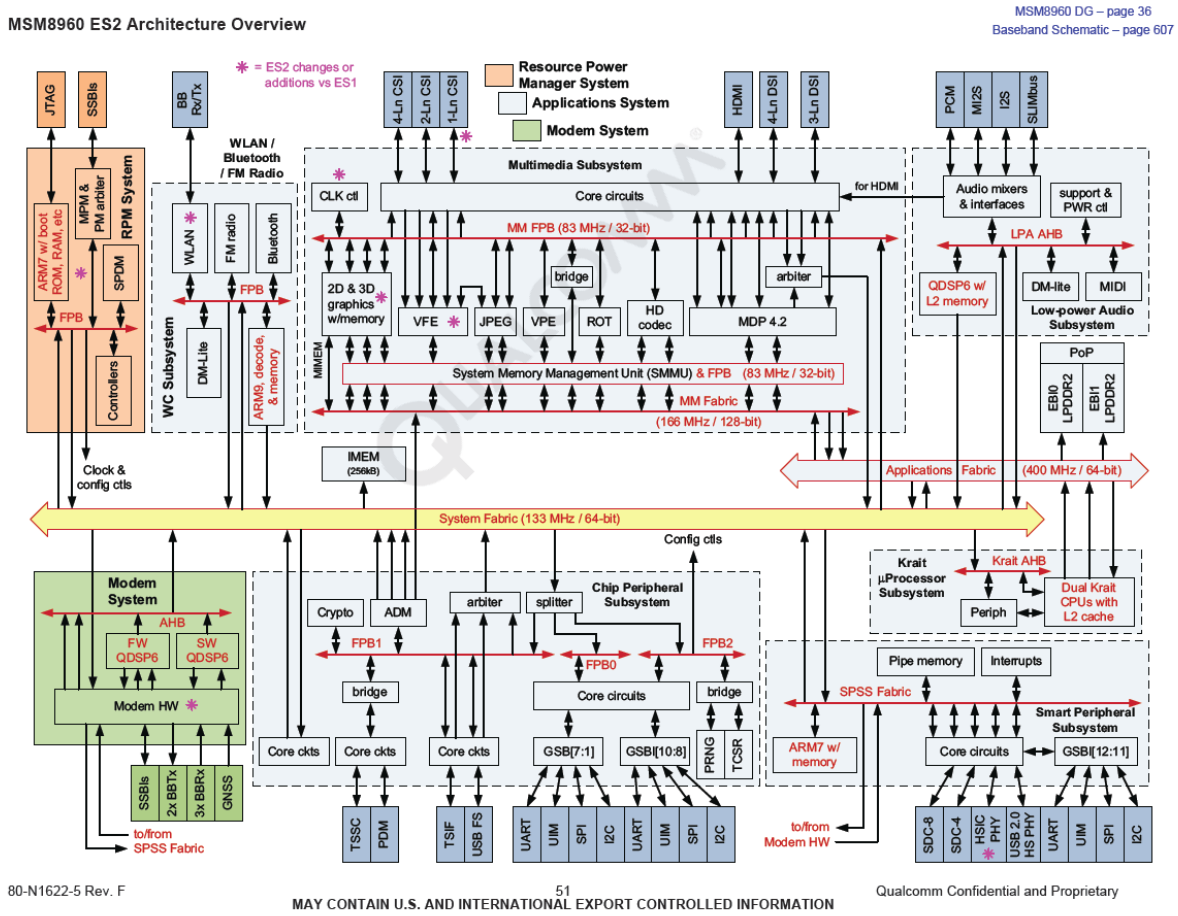
	 MSM8960 Chipset Schematics and Design Guidelines, p. 50. On information and belief, when executing Jazelle instructions within a loop, the processor maintains a loop counter that is bidirectionally connected to the Accused Microprocessor's internal buses.
said loop counter being connected to	Each ARM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

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

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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

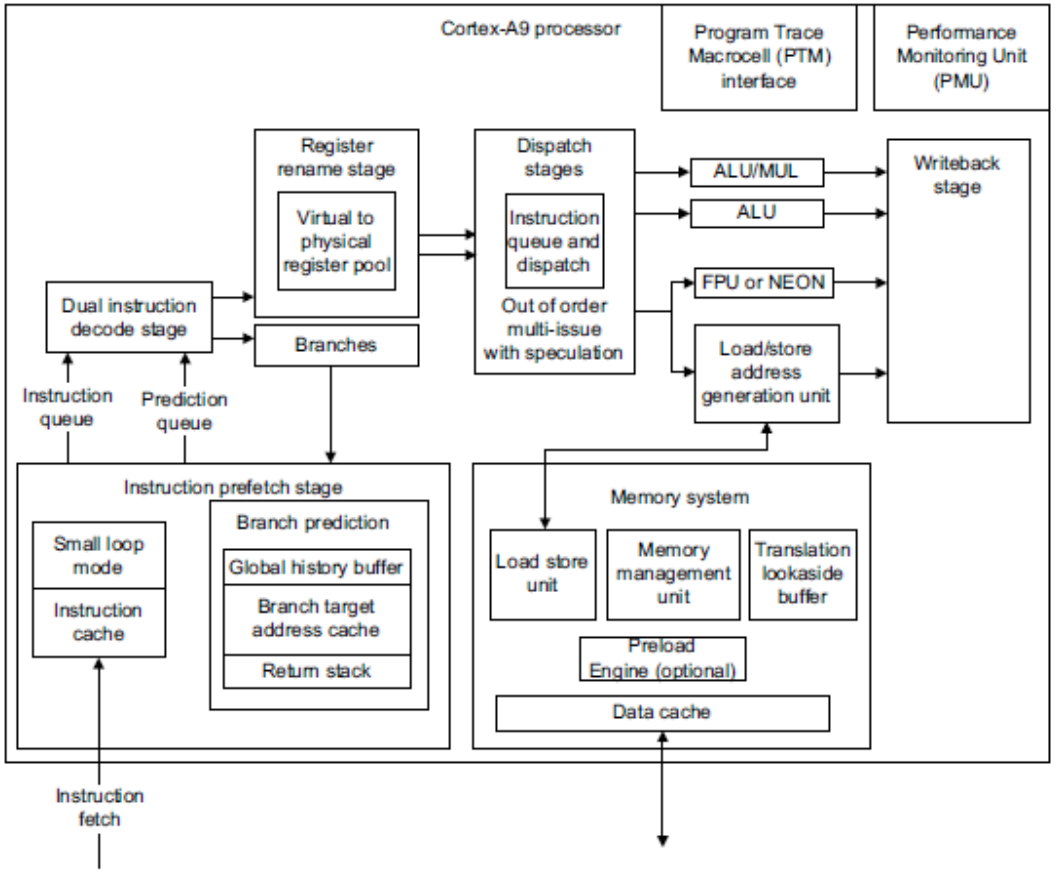
	 **Figure 2-1 Cortex-A9 processor top-level diagram** Cortex-A9 Technical Reference Manual, p.2-2.
said internal data bus being bidirectionally connected to a stack pointer, return stack pointer, mode register and instruction register,	Each ARM core has a stack pointer.

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

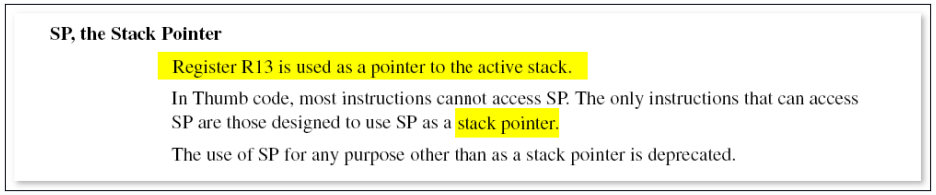
	 ARM Architecture Reference Manual - ARMv7-A and ARMv7-R p. A2-11 On information and belief, the Register File of each Accused Microprocessor is bidirectionally connected to the data bus. The Register File includes Register R13, also known as SP or the stack pointer, which points to the active stack and Register R14, also known as LR, the Link Register, which contains an identical memory address as the top of the return stack.

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

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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

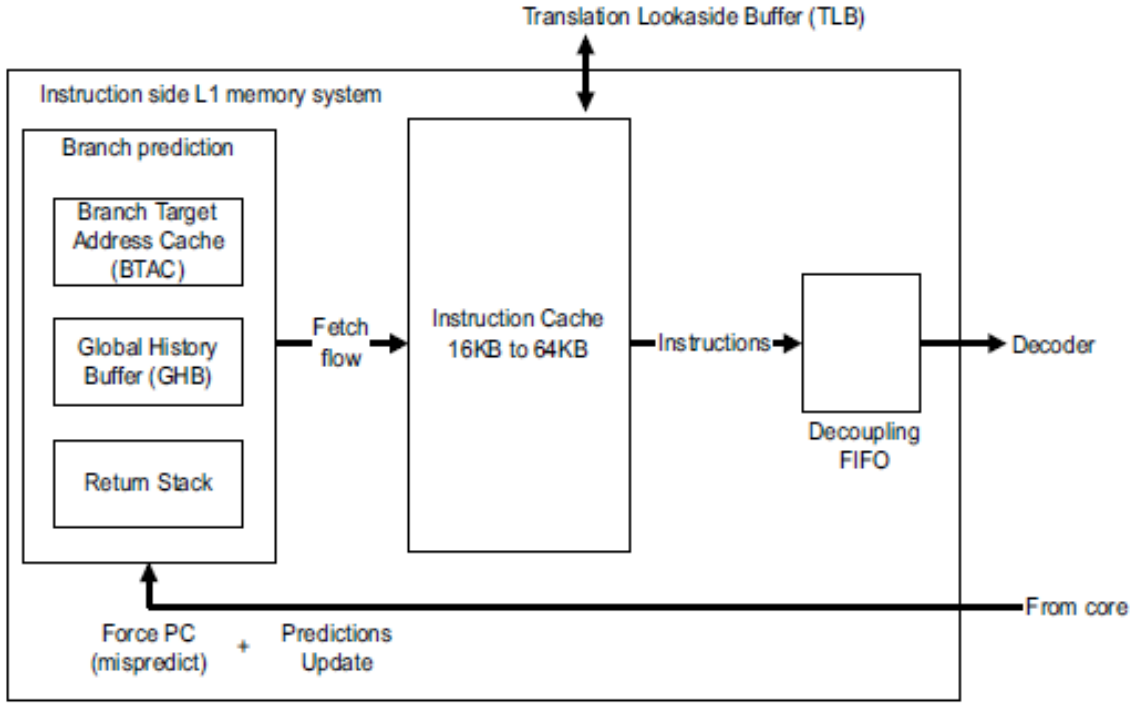


Figure 6-1 Branch prediction and instruction cache

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

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

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

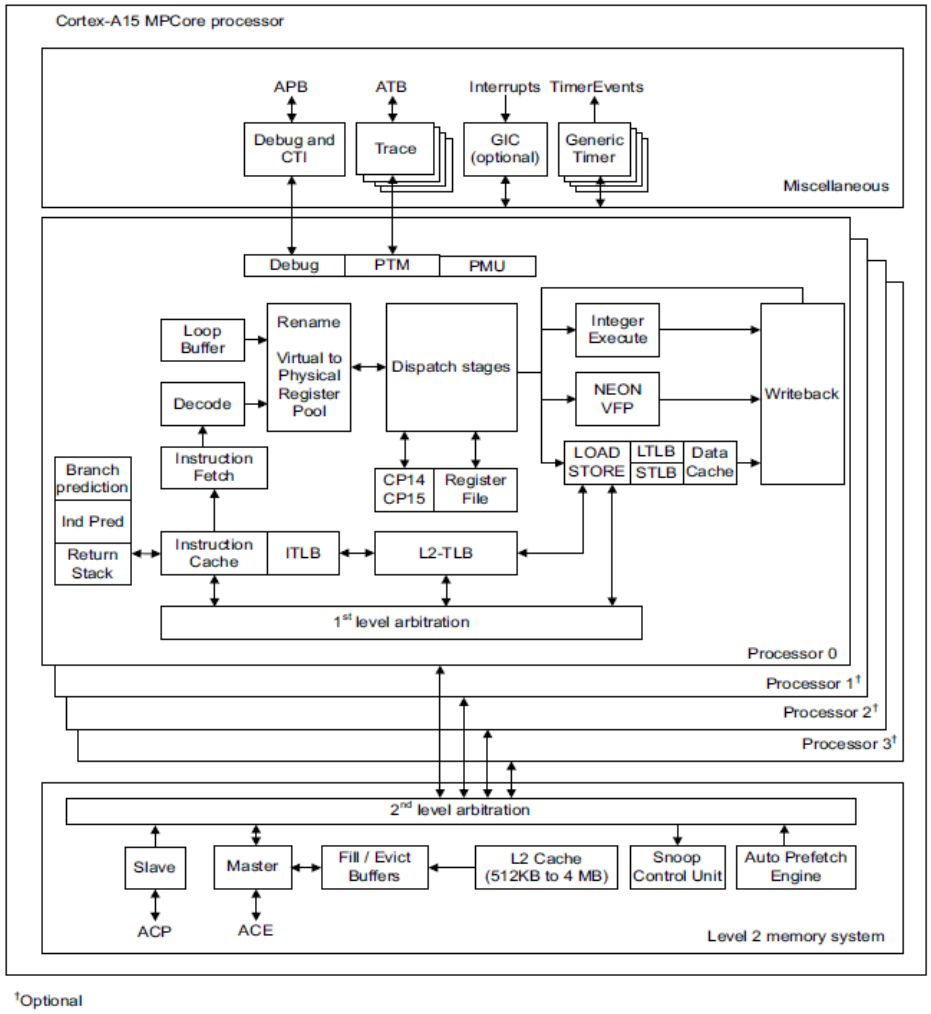


Figure 2-1 Block diagram

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

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

	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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

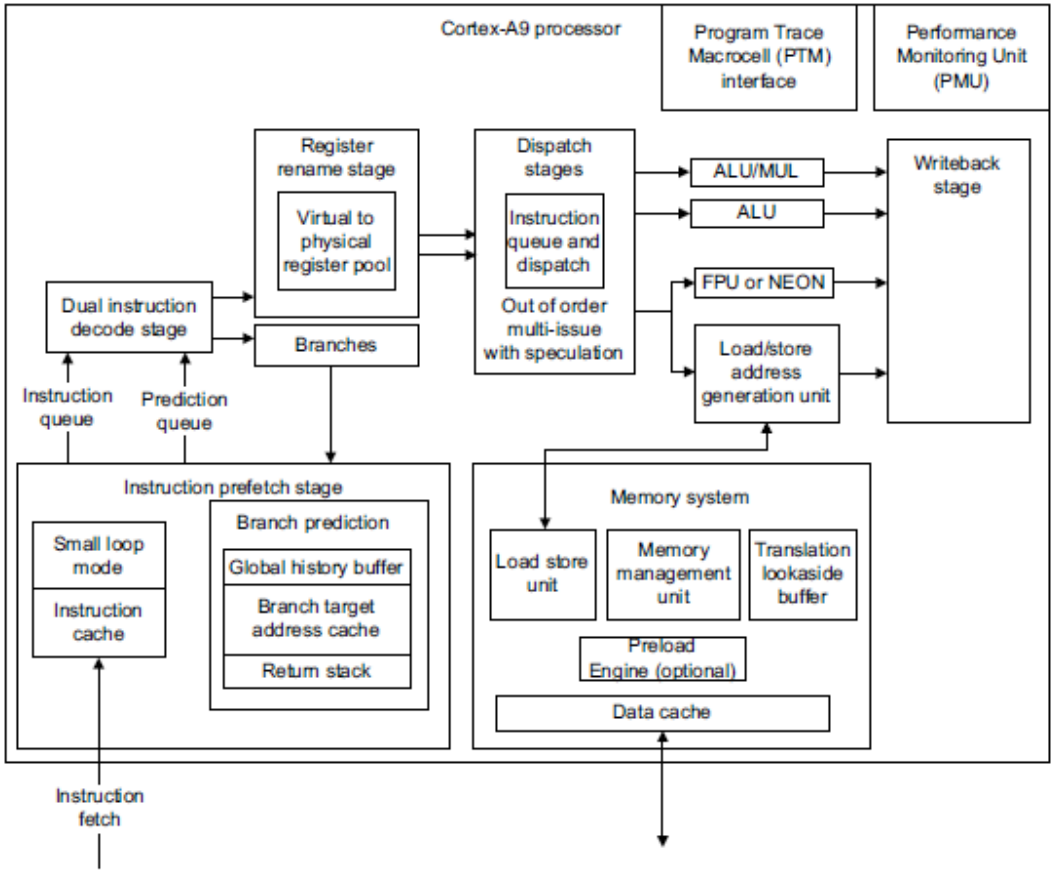
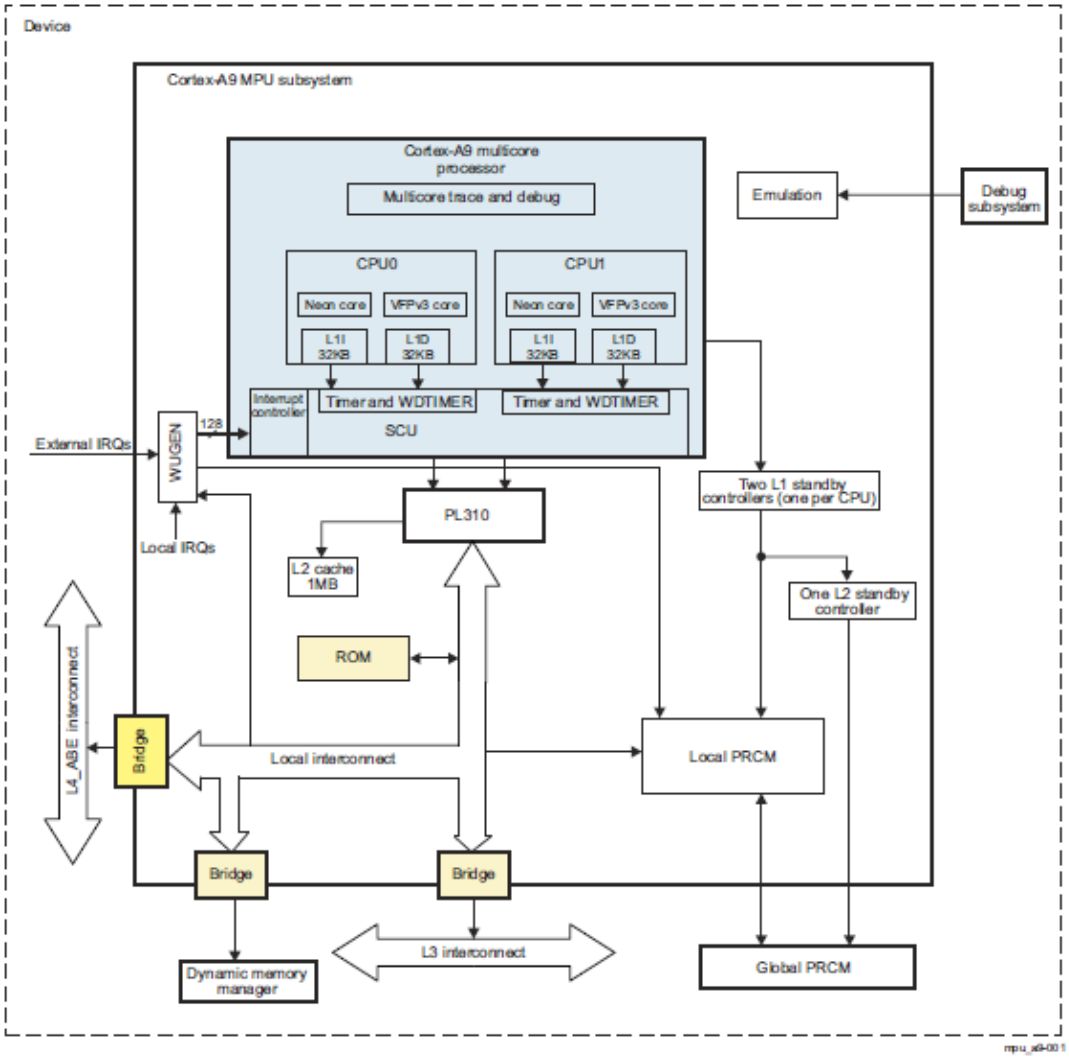


Figure 2-1 Cortex-A9 processor top-level diagram

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

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

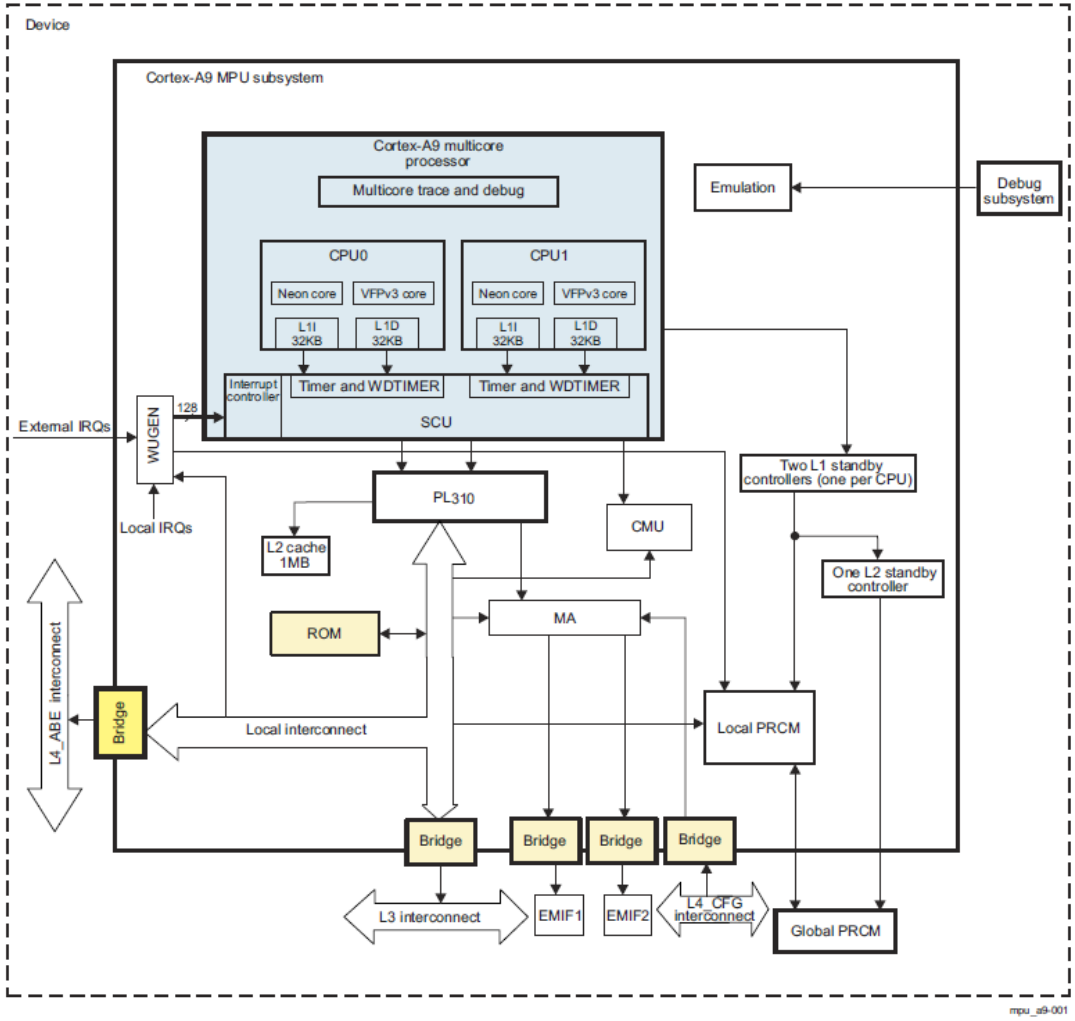
Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4430 Technical Reference Manual, p. 1077.

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

Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4470 Technical Reference Manual, p. 1091.

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

1.3 Block Diagram

Figure 1-1 illustrates the block diagram of Exynos 5250.

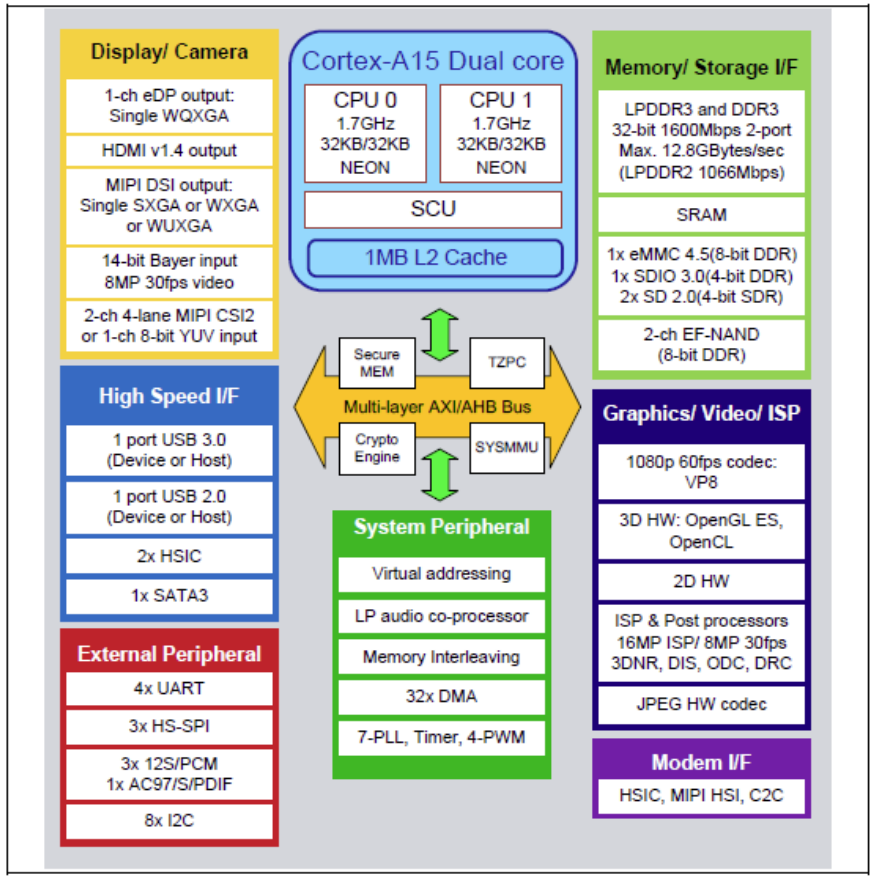
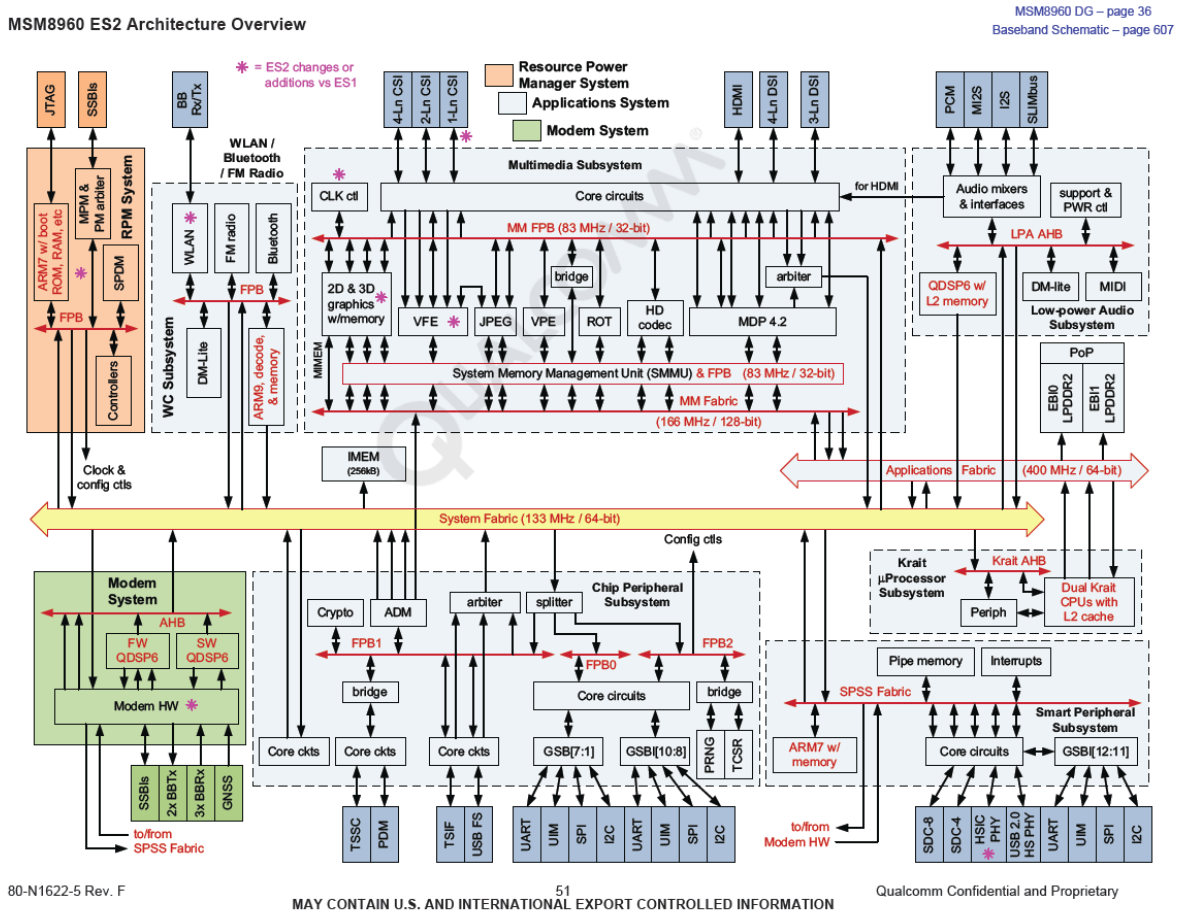


Figure 1-1 Block Diagram of Exynos 5250

Exynos 5 Dual Reference Manual, Fig. 1-1.

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



MSM8960 Chipset Schematics and Design Guidelines, p. 50.

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

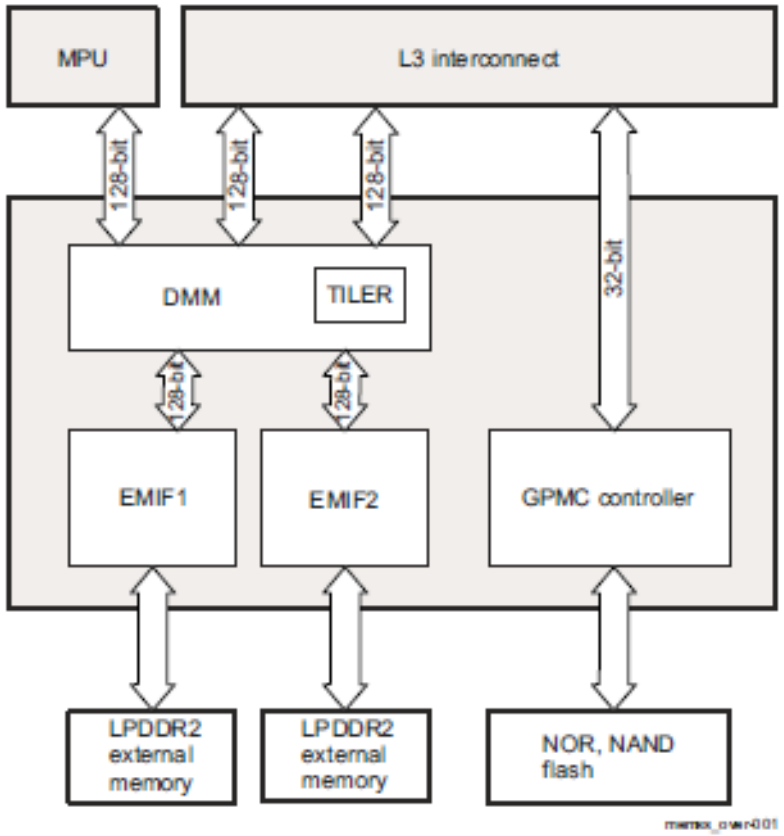
EXHIBIT E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

	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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

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

Figure 15-1. Memory Subsystem Functional Diagram

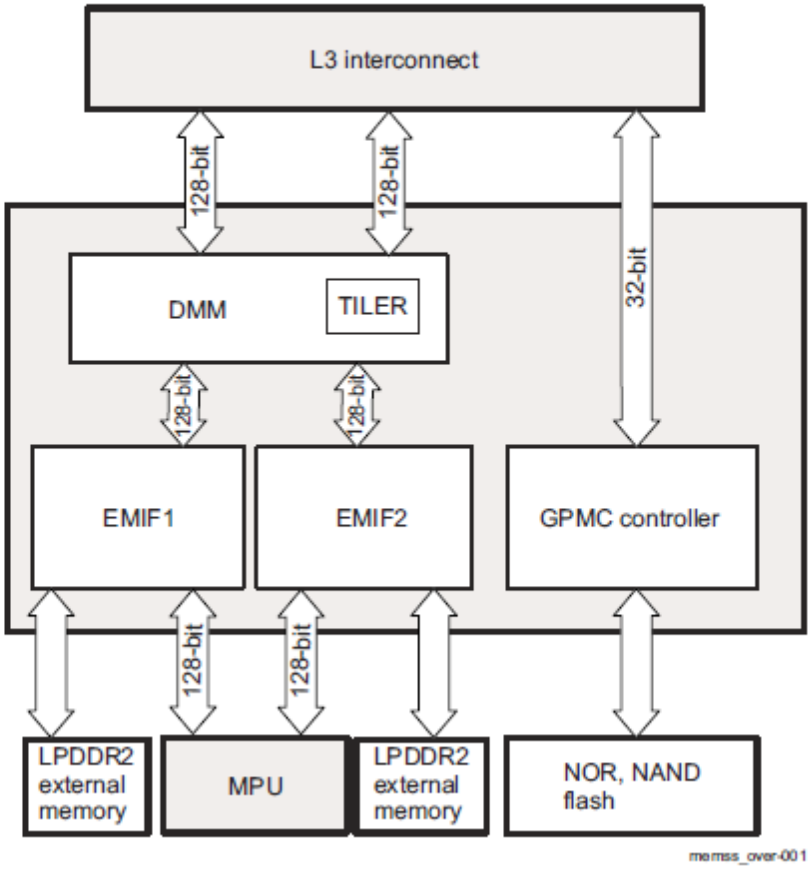


OMAP4430 Technical Reference Manual, p. 3207.

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

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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

1.3 Block Diagram

Figure 1-1 illustrates the block diagram of Exynos 5250.

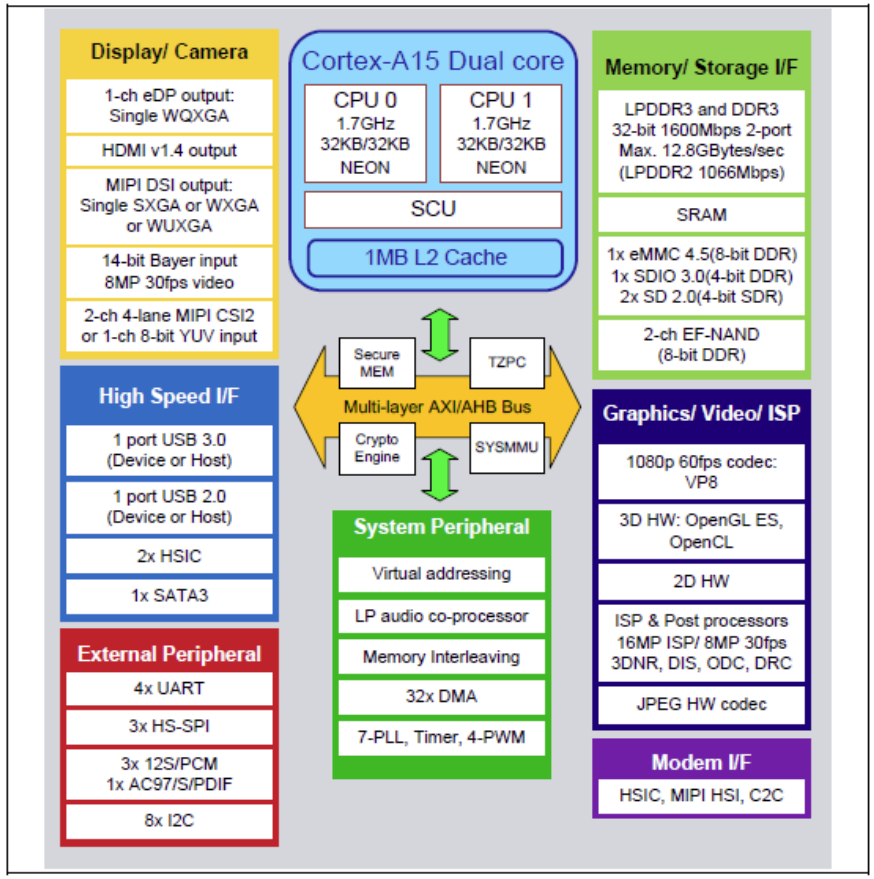
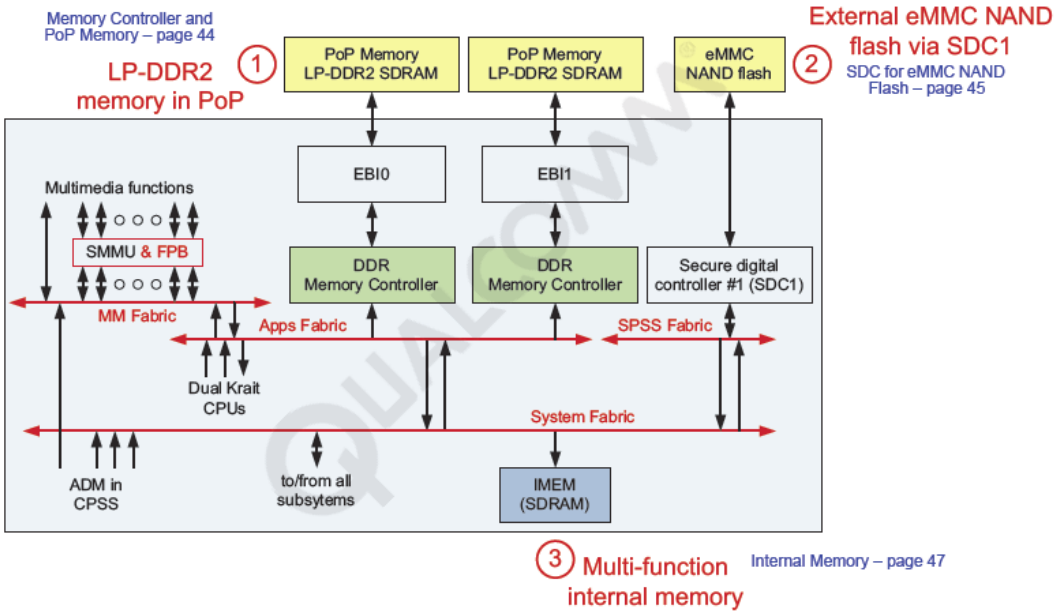


Figure 1-1 Block Diagram of Exynos 5250

Exynos 5 Dual Reference Manual, Fig. 1-1.

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



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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

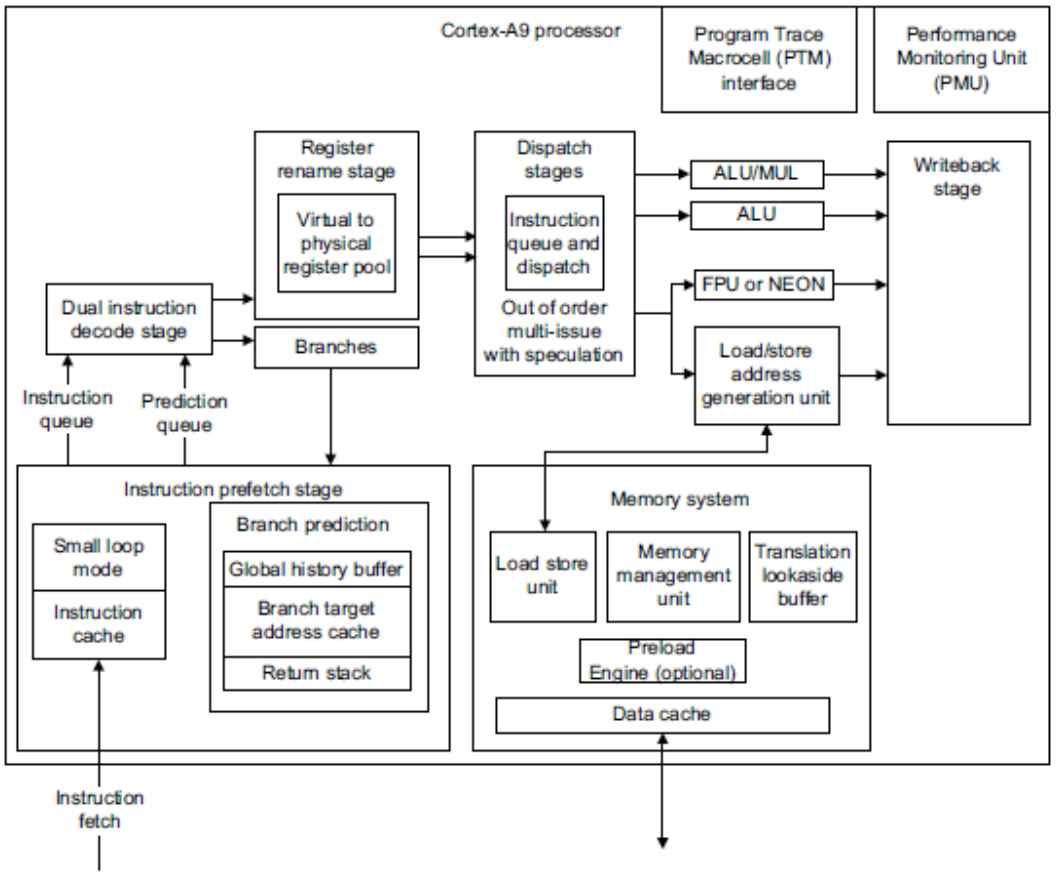


Figure 2-1 Cortex-A9 processor top-level diagram

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

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

	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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

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

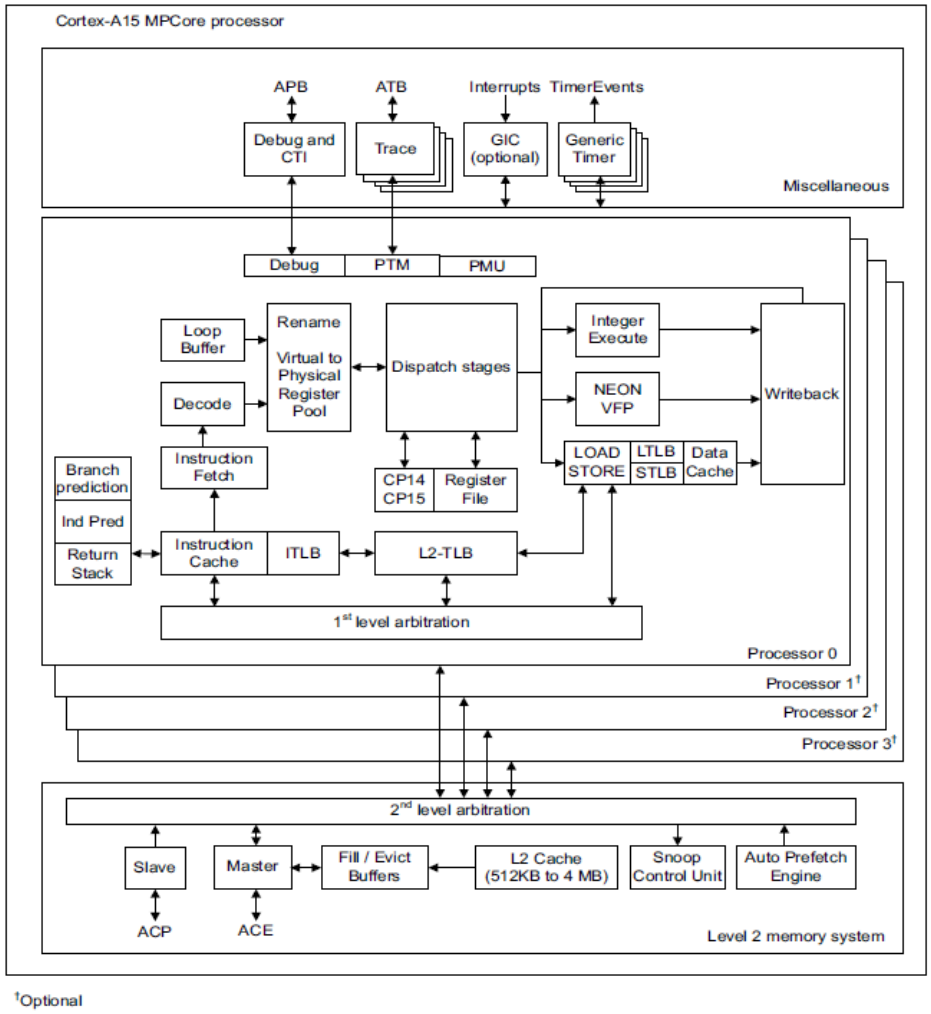
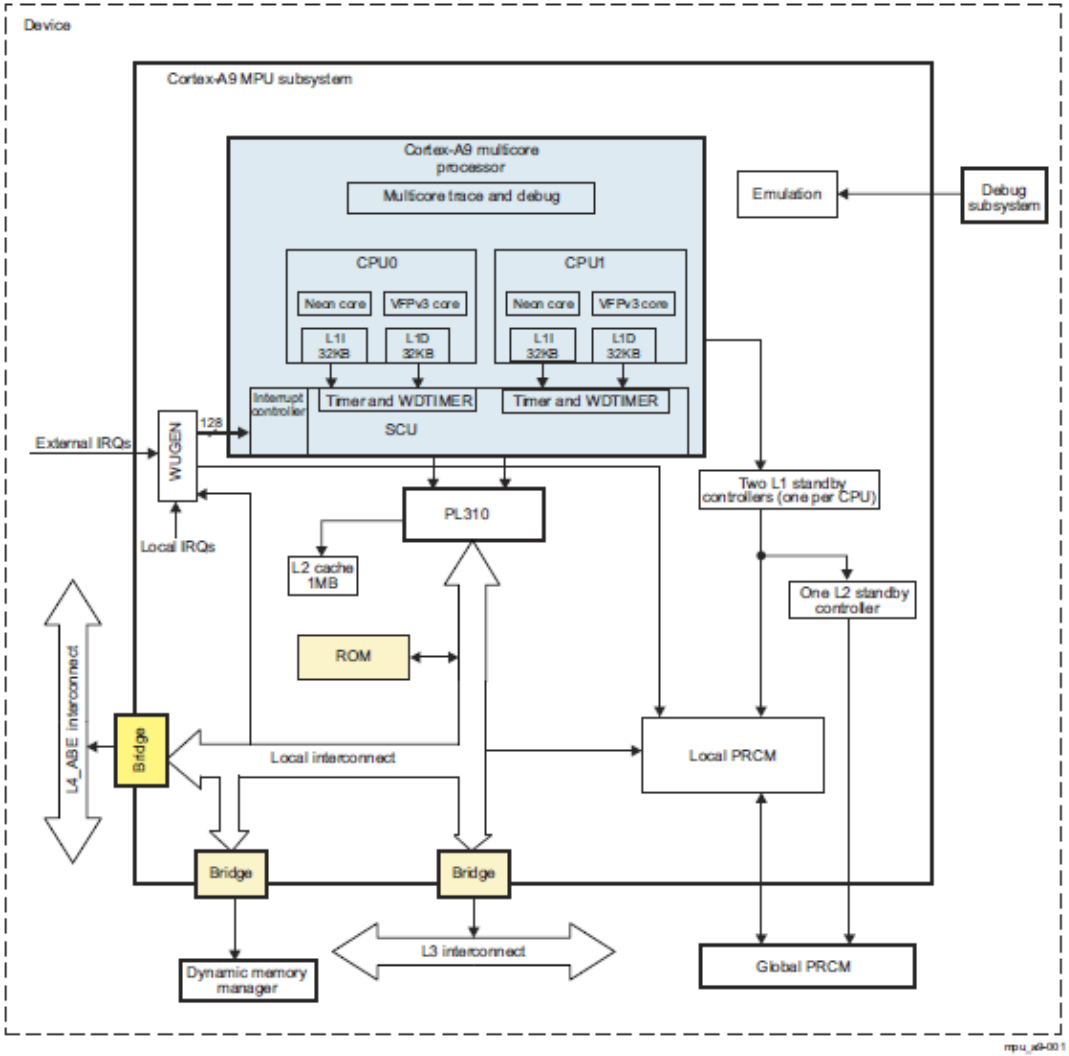


Figure 2-1 Block diagram

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

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

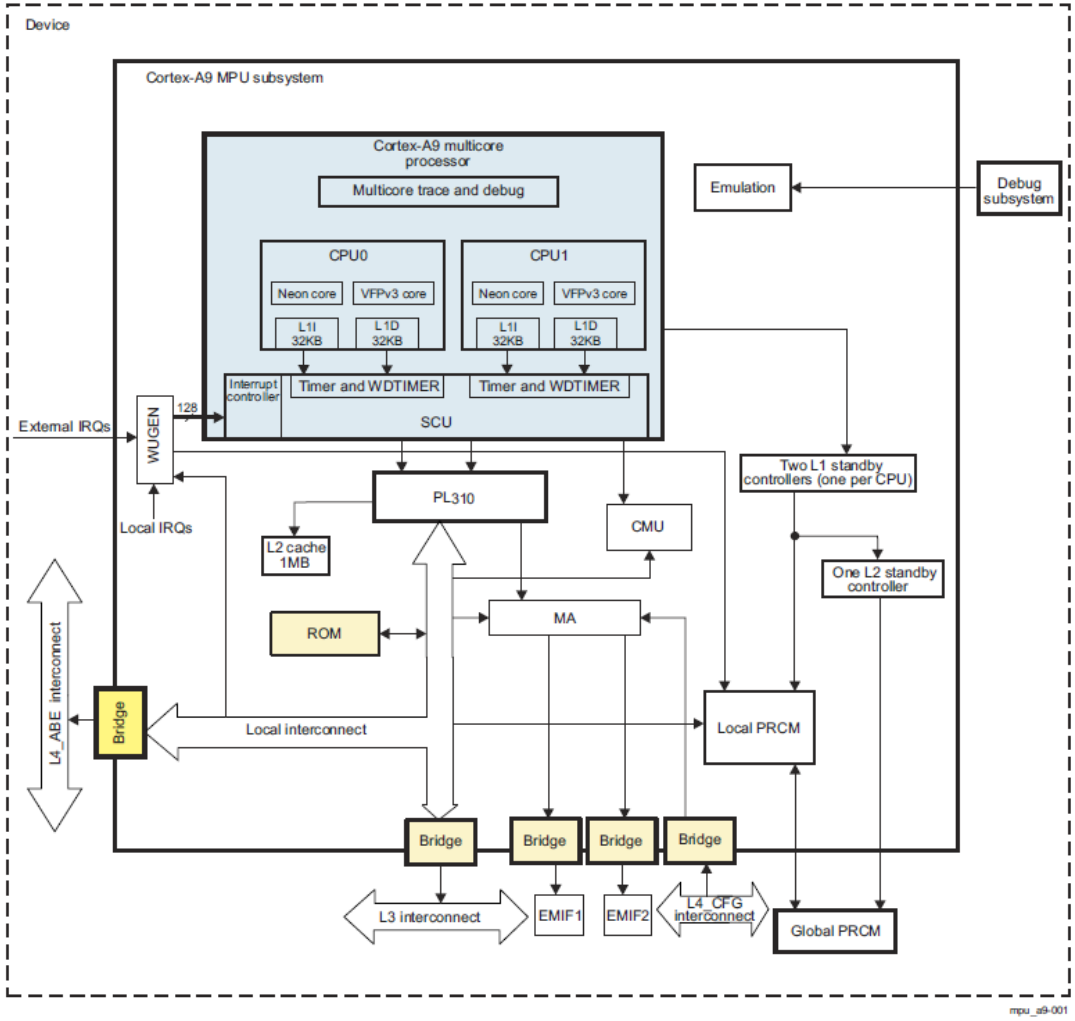
Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4430 Technical Reference Manual, p. 1077.

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

Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4470 Technical Reference Manual, p. 1091.

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

	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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

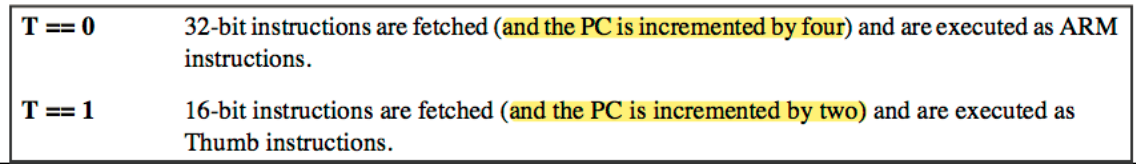
	 **T == 0** 32-bit instructions are fetched (and the PC is incremented by four) and are executed as ARM instructions. **T == 1** 16-bit instructions are fetched (and the PC is incremented by two) and are executed as Thumb instructions. See ARM Architecture Reference Manual at A6-2 . **A3.1.1 Address incrementing 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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

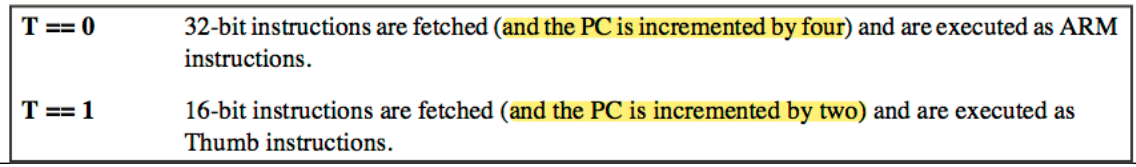
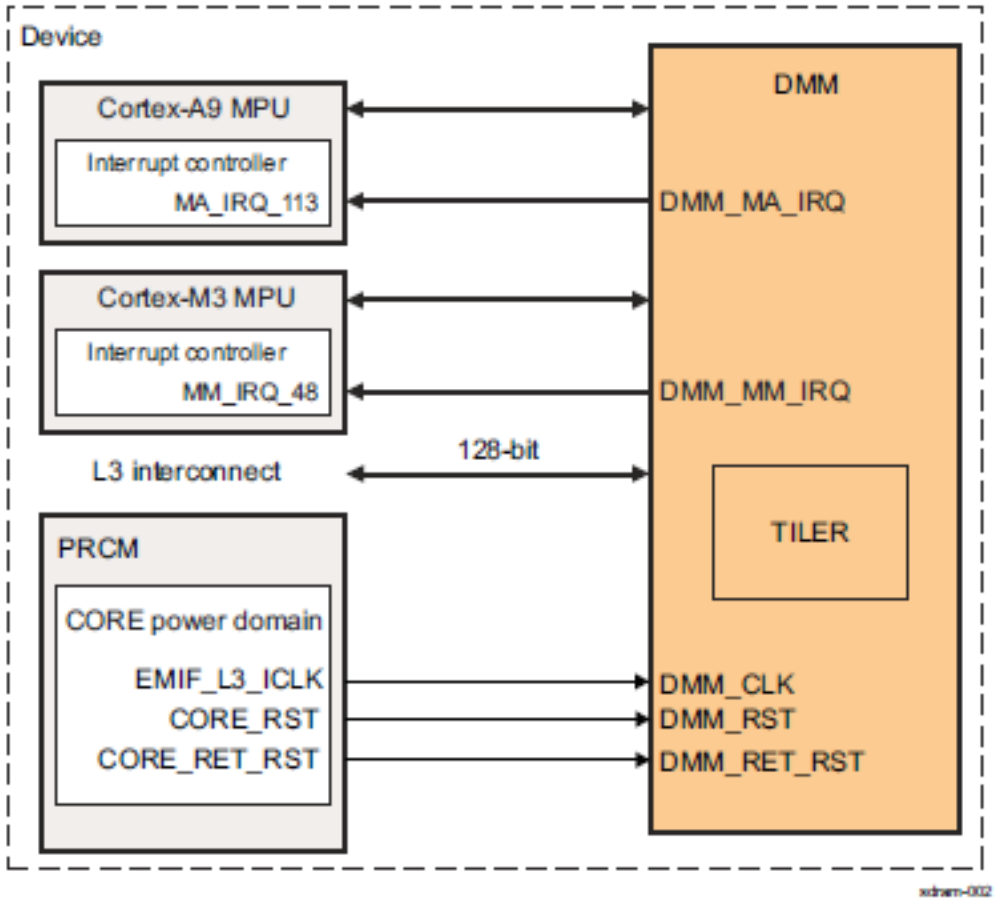
	 **T == 0** 32-bit instructions are fetched (and the PC is incremented by four) and are executed as ARM instructions. **T == 1** 16-bit instructions are fetched (and the PC is incremented by two) and are executed as Thumb instructions. See ARM Architecture Reference Manual at A6-2 . **A3.1.1 Address incrementing 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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

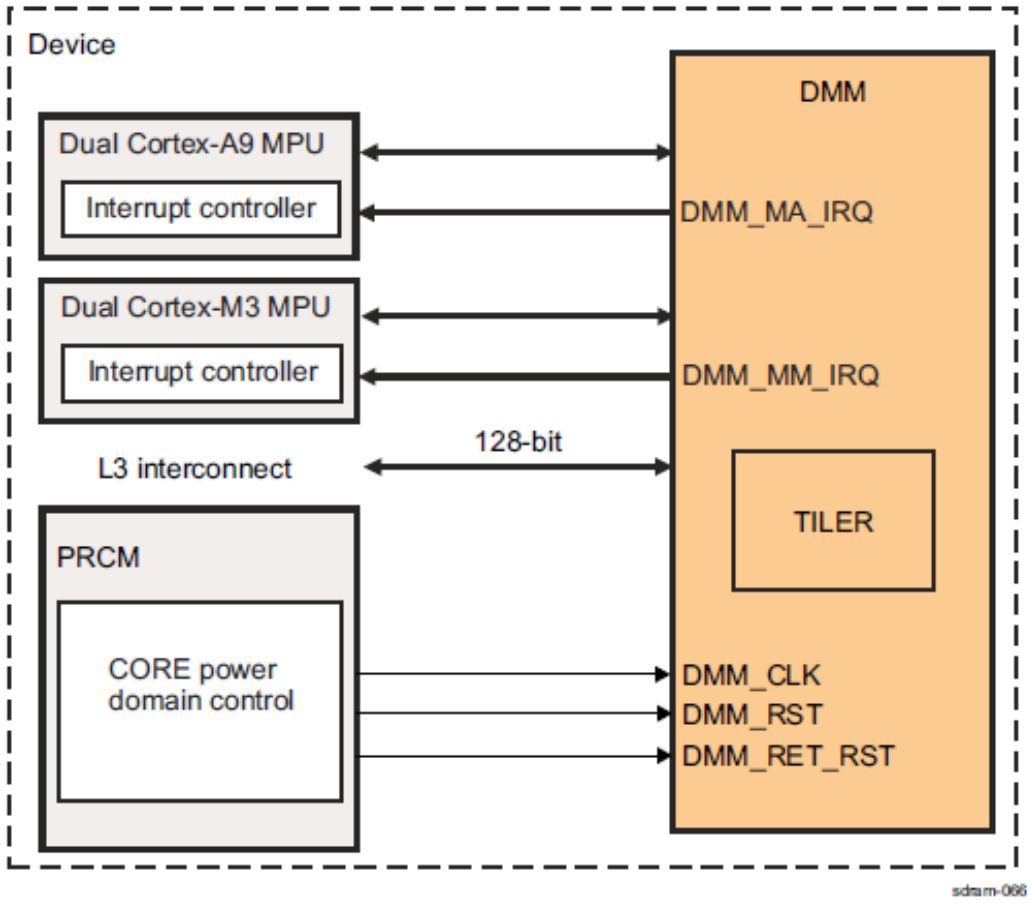
Figure 15-3. DMM Integration



OMAP4430 Technical Reference Manual, p. 3213.

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

Figure 16-2. DMM and Tiler Overview



OMAP4470 Technical Reference Manual, p. 3308.

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

1.3 Block Diagram

Figure 1-1 illustrates the block diagram of Exynos 5250.

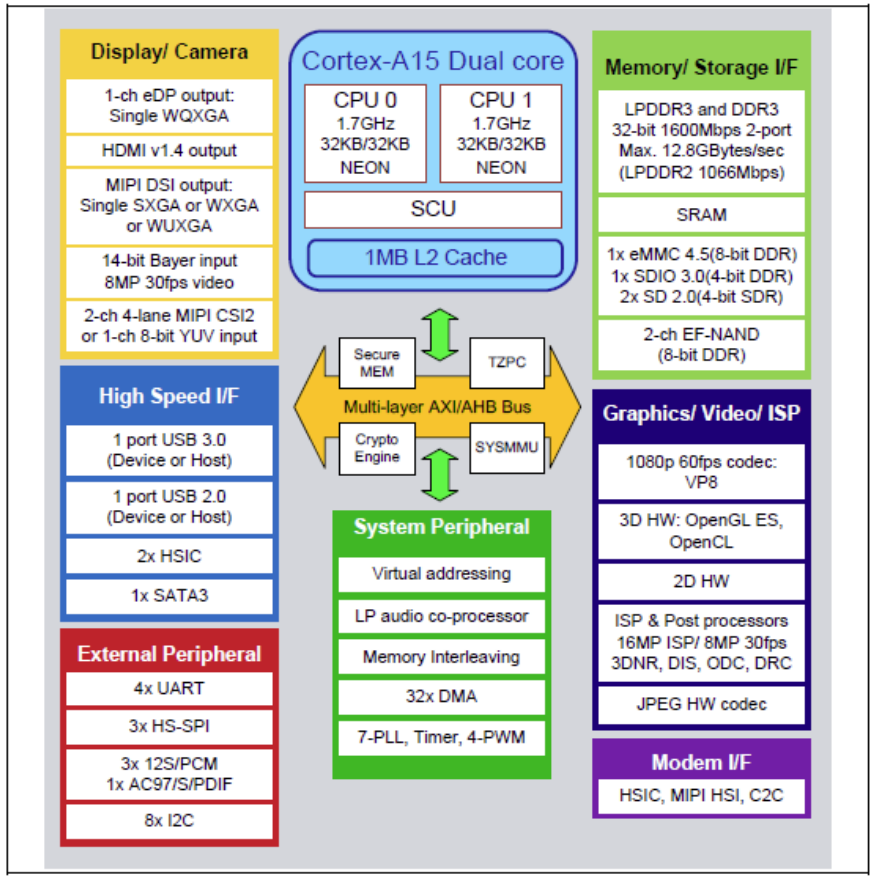
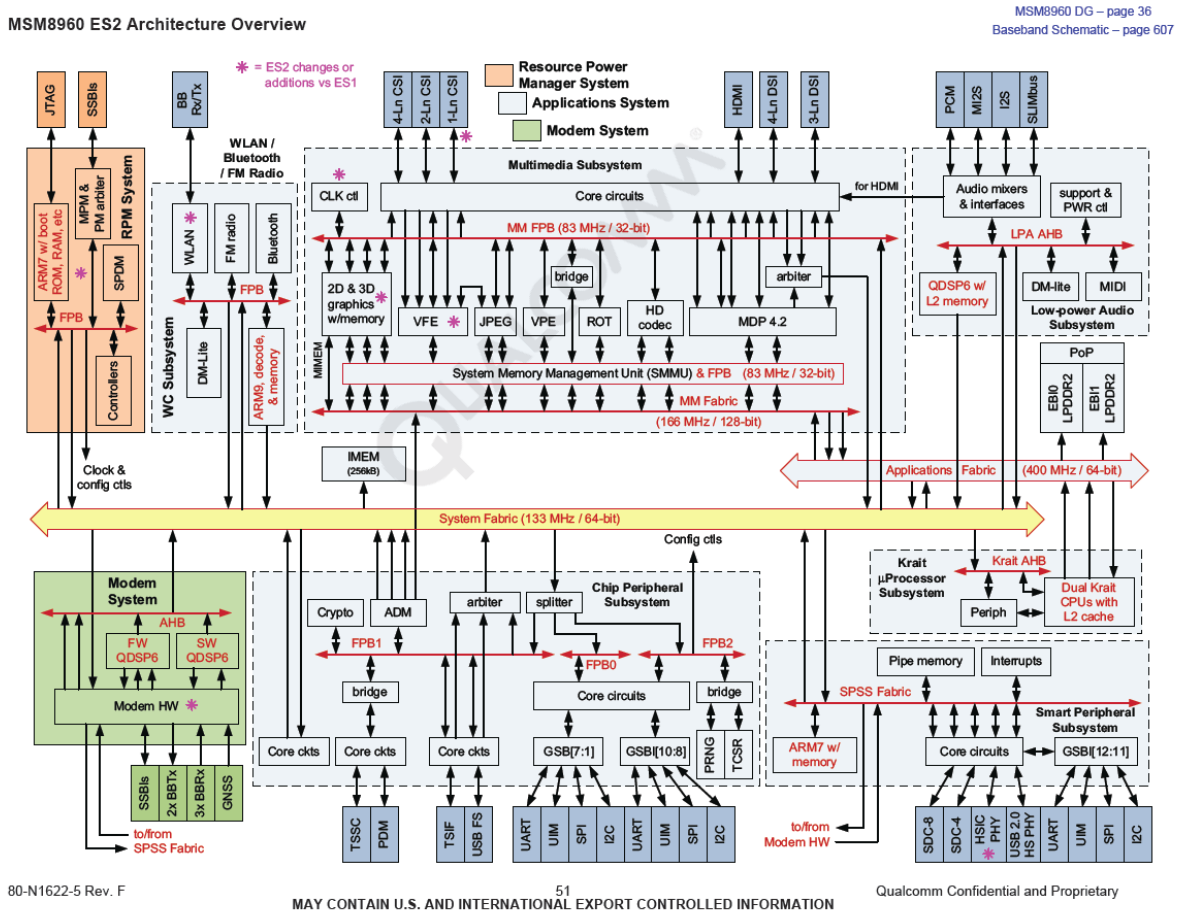


Figure 1-1 Block Diagram of Exynos 5250

Exynos 5 Dual Reference Manual, Fig. 1-1.

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



MSM8960 Chipset Schematics and Design Guidelines, p. 50.

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

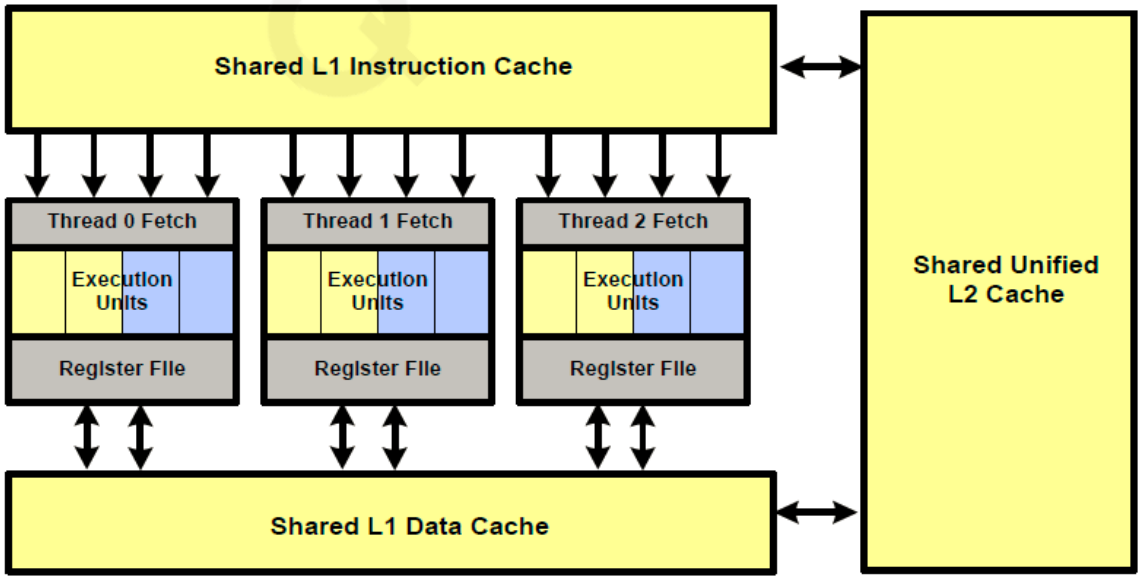
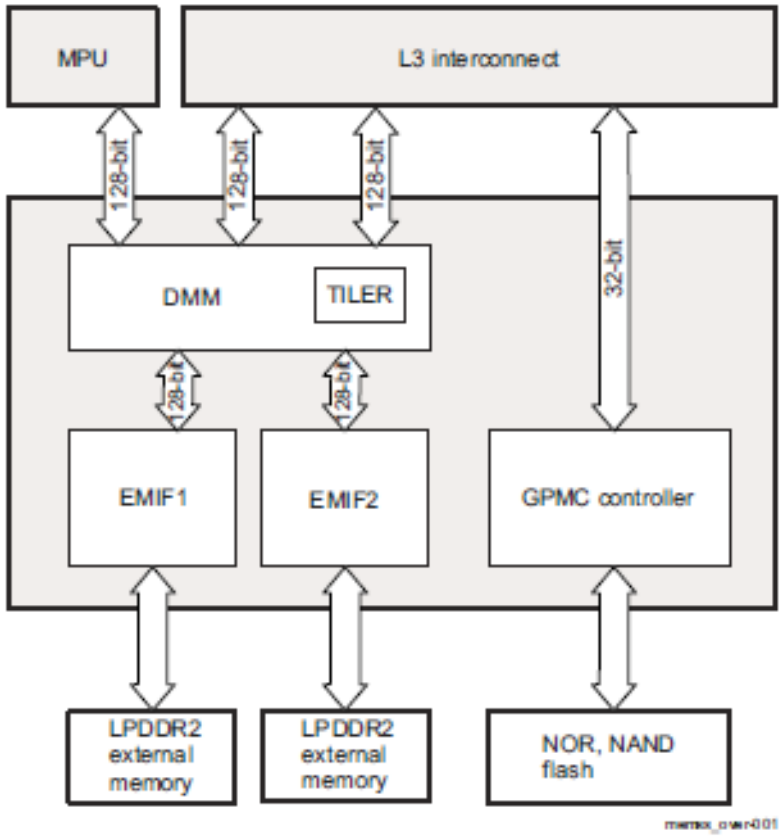
	 Hexagon Programming: Processor Overview, p.10.
said memory controller having an address/data bus and a plurality of control lines for connection to a random access memory.	On information and belief the memory controller of each Accused Microprocessor has an address/data bus and a plurality of control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

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

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

Figure 15-1. Memory Subsystem Functional Diagram

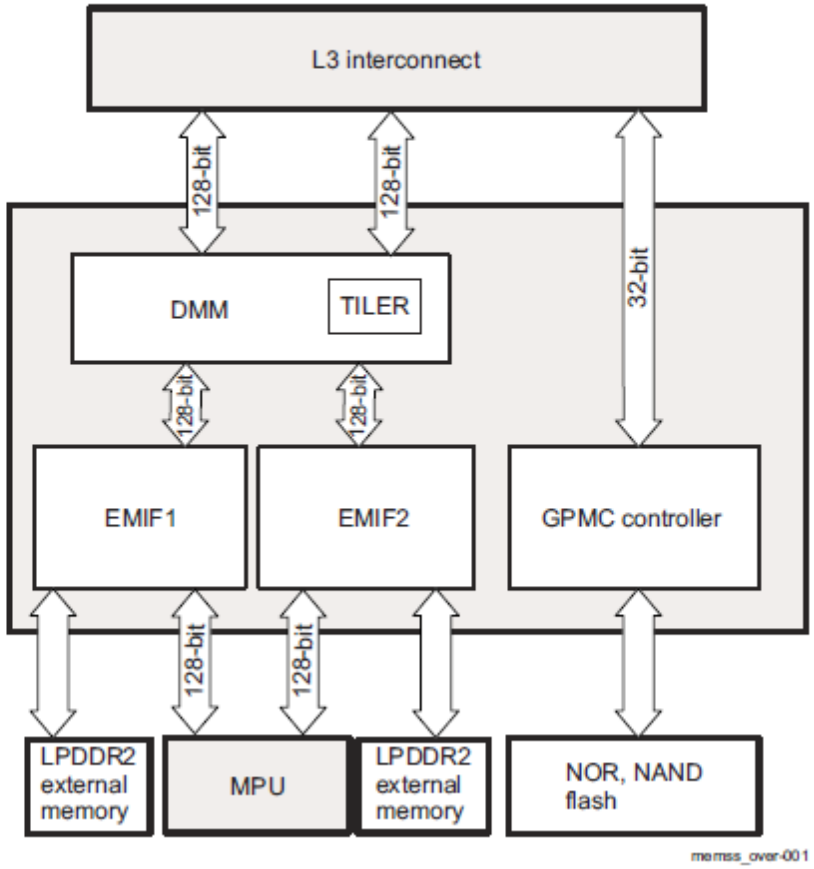


OMAP4430 Technical Reference Manual, p. 3207.

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

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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

1.4.2 Memory Subsystem

Exynos 5250 provides the leading memory bandwidth for mobile applications.

- Mobile DDR3 (LPDDR3) Interface
  - Two ports x32 data bus with 800 MHz per pin, double data rate (DDR)
  - 1.2 V interface voltage
  - Density: Maximum 4 GB by memory map limit, Recommend: Maximum 2 memory dice per port.

NOTE: The max number of column address bits is 10.

- Mobile DDR2 (LPDDR2) interface
  - Two ports x32 data bus with 533 MHz per pin and double data rate (DDR)
  - 1.2 V interface voltage
  - Density: Maximum 4 GB by memory map limit, Recommend: Maximum 2 memory dice per port.

NOTE: The max number of column address bits is 10.

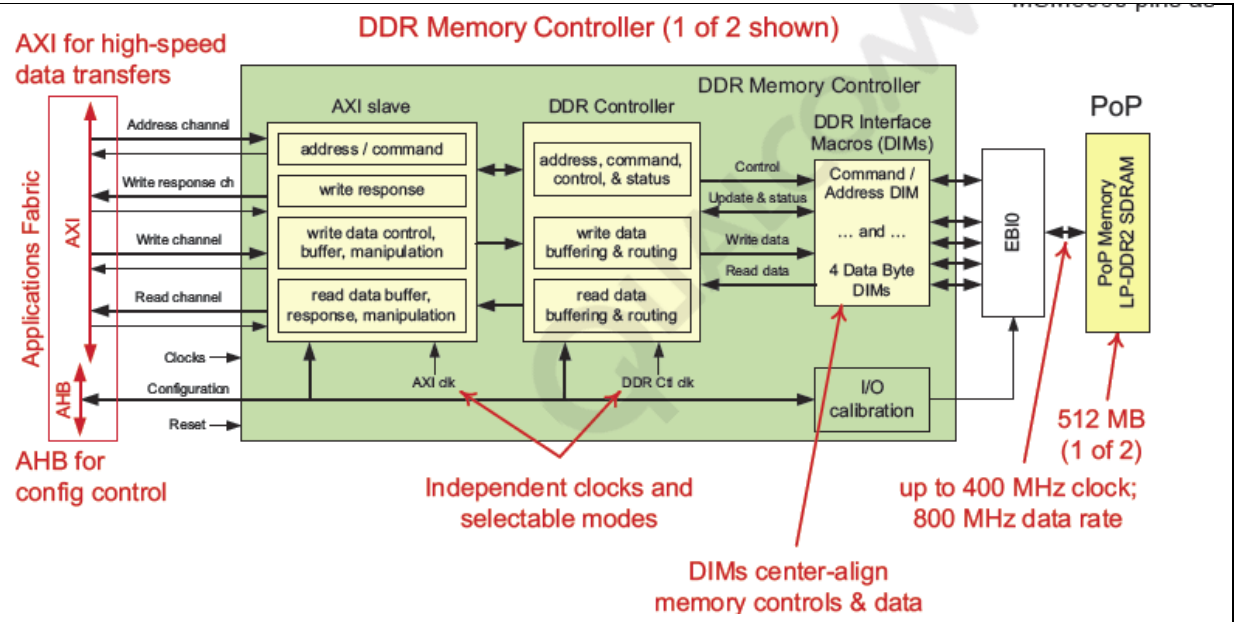
- DDR3/DDR3L Interface
  - Two ports x32 data bus with 800 MHz per pin and double data rate (DDR)
  - 1.5 V/1.35 V interface voltage
  - Density: Maximum 4 GB by memory map limit, Recommend: Maximum 2 memory dice per port.

NOTE: The max number of column address bits is 10.

- eMMC and SD card interface
  - One channel 8-bit eMMC4.5 (1.8 V only)
  - One channel 4-bit SD3.0 (1.8 V only)
  - Two channel eMMC4.3/SD2.0 3.3 V/1.8 V interface voltage
- EF-NAND3.0 interface
  - Two ports x8-bit data bus with up to 200 MHz per pin and double data rate (DDR)
- Embedded internal ROM booting: The system does not need a booting device

Exynos 5 Dual Reference Manual, p. 1-8.

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

	 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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

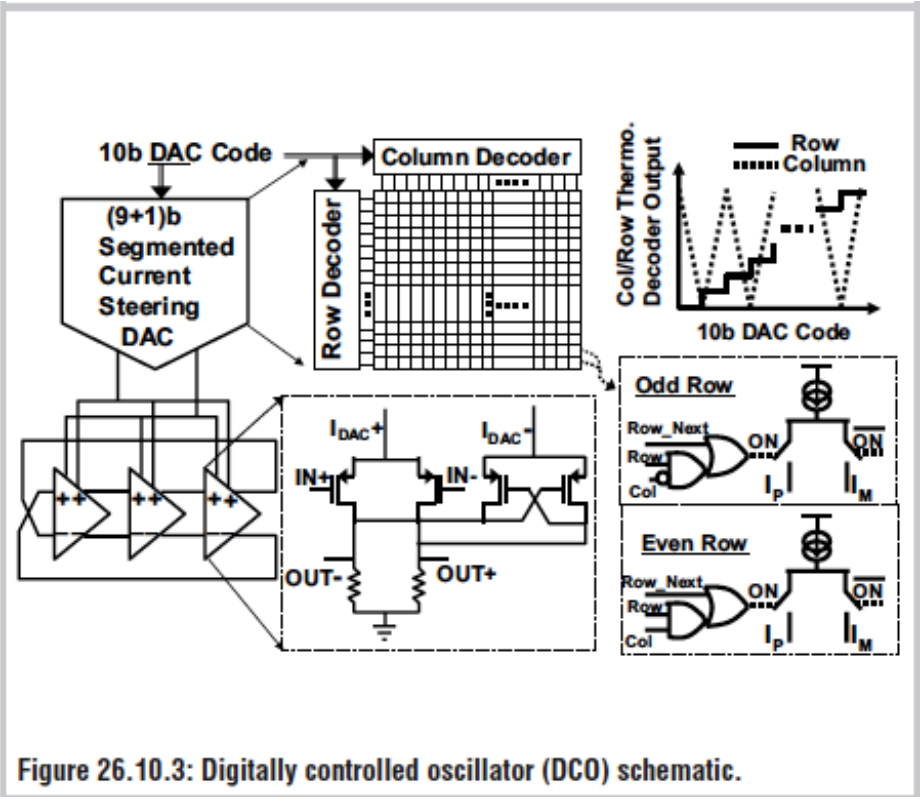
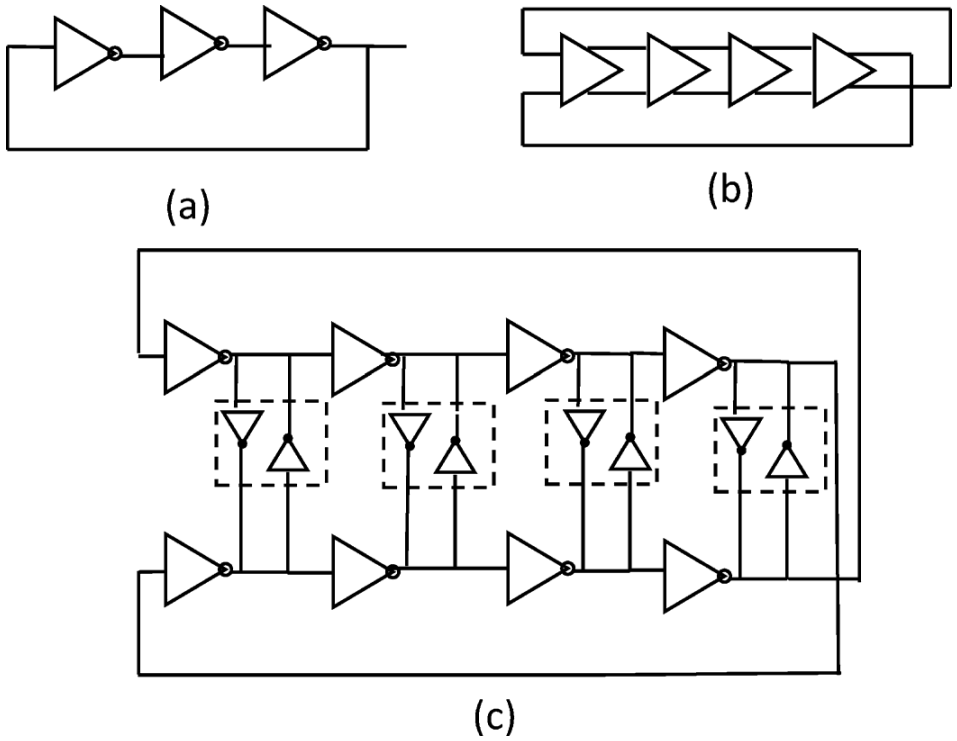
integrated circuit.	For example, generally in TI microprocessors, PLLs have voltage-controlled oscillators, which are circuits capable of maintaining an alternating output. TI has published papers regarding its DPLLs confirming the use of ring oscillators.  Figure 26.10.3: Digitally controlled oscillator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

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

(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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

	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 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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

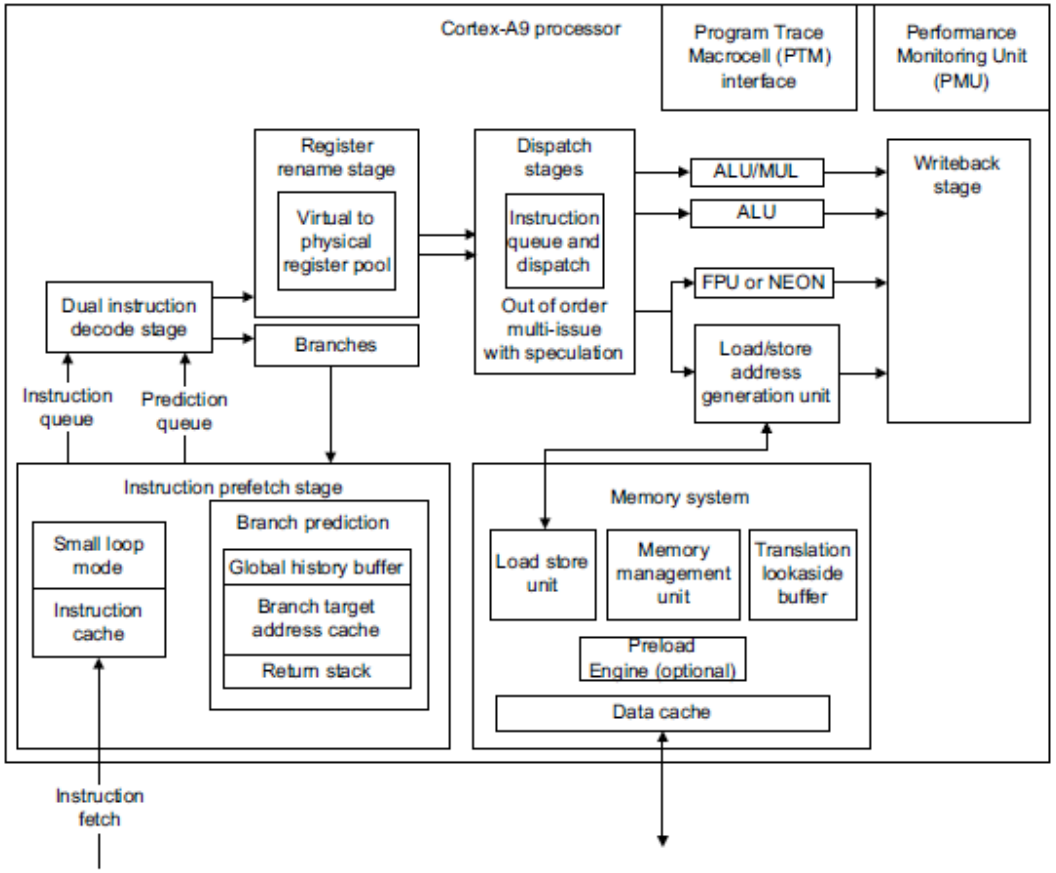
Claim 9	
The microprocessor of claim 1	
in which said first push down stack has a first plurality of stack elements configured as latches,	On information and belief, the first plurality of stack elements is configured as latches (register file or L1 data cache).  **Figure 2-1 Cortex-A9 processor top-level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

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

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

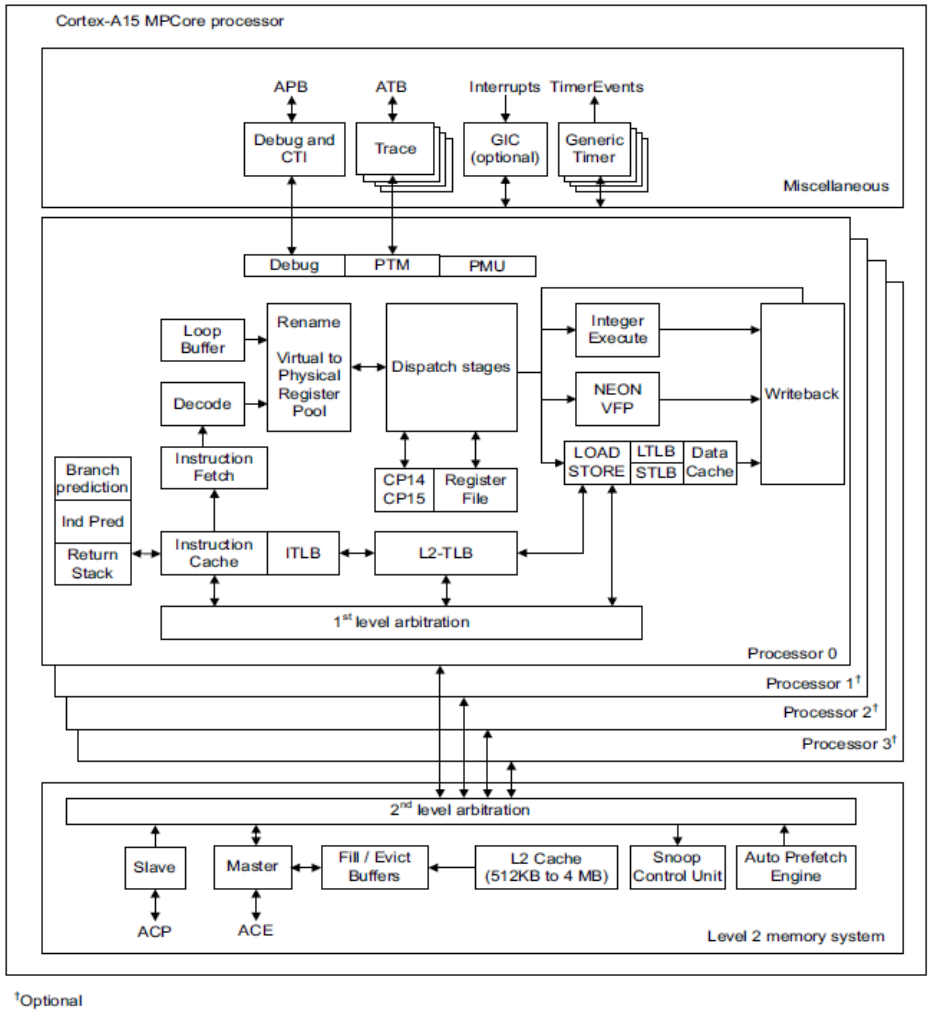


Figure 2-1 Block diagram

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

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

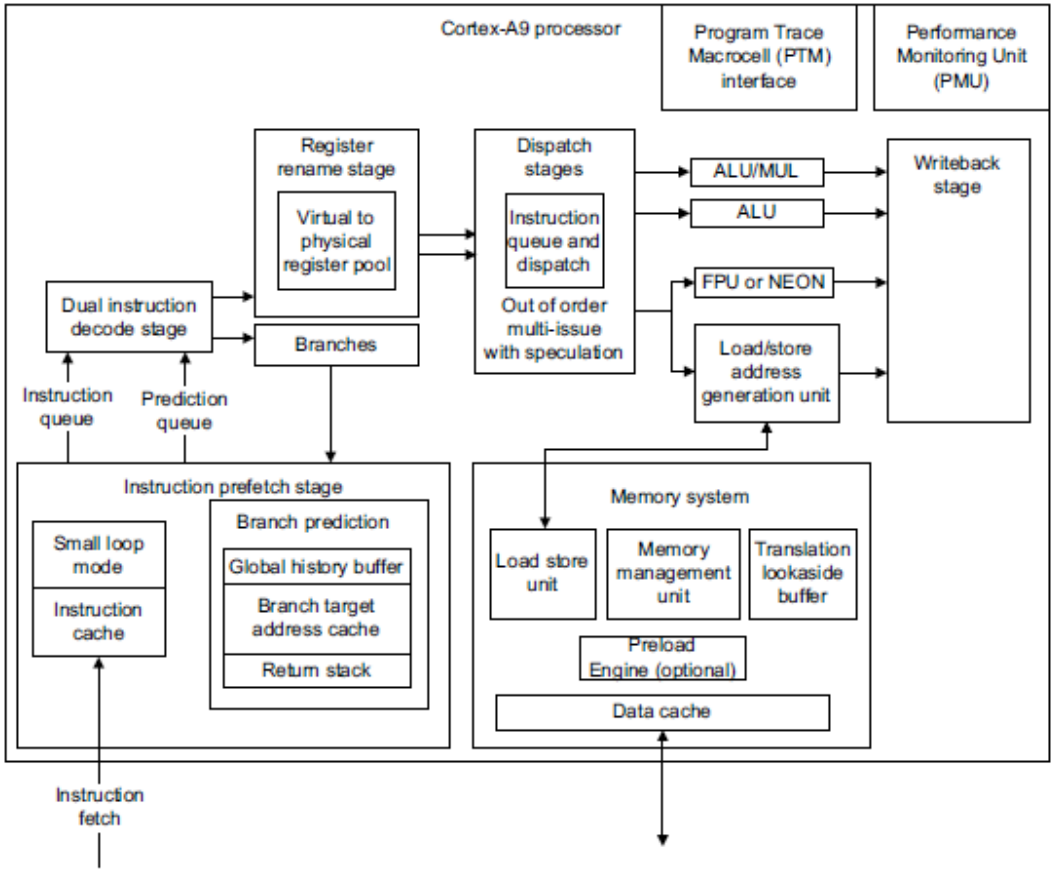
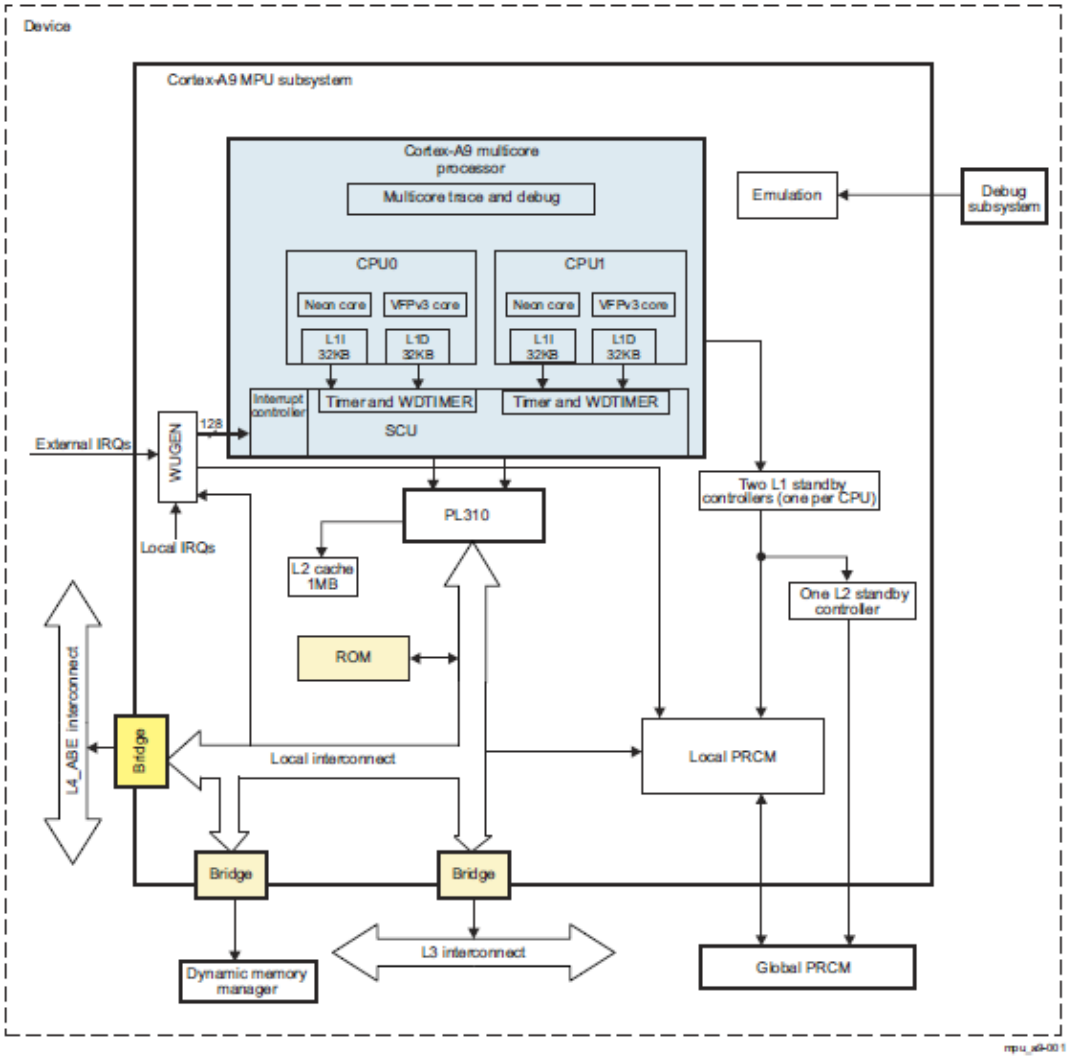
a second plurality of stack elements configured as a random access memory,	On information and belief, the second plurality of stack elements is configured as random access memory (L1 cache or L2 cache).  **Figure 2-1 Cortex-A9 processor top-level diagram** Cortex-A9 Technical Reference Manual, p.2-2.

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

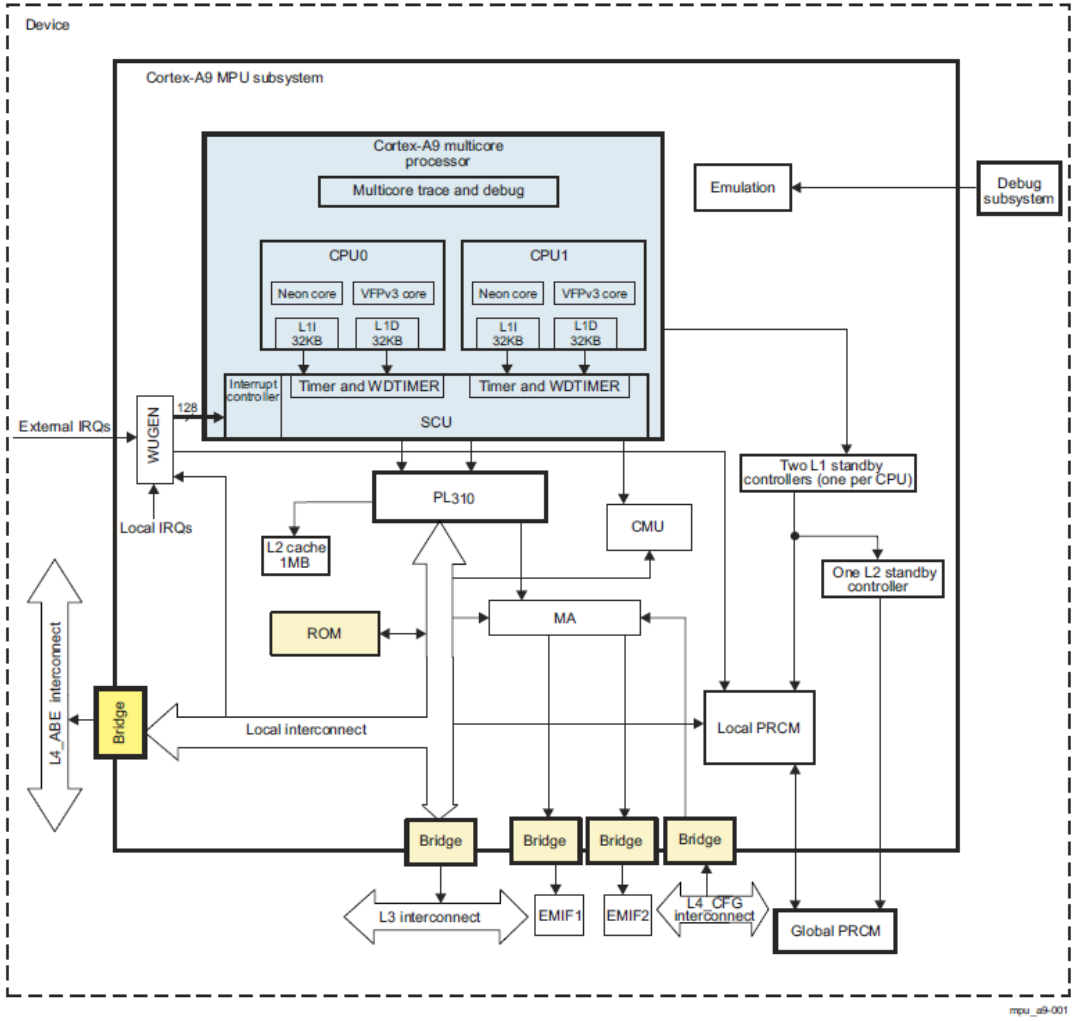
Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4430 Technical Reference Manual, p. 1077.

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

Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4470 Technical Reference Manual, p. 1091.

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

1.3 Block Diagram

Figure 1-1 illustrates the block diagram of Exynos 5250.

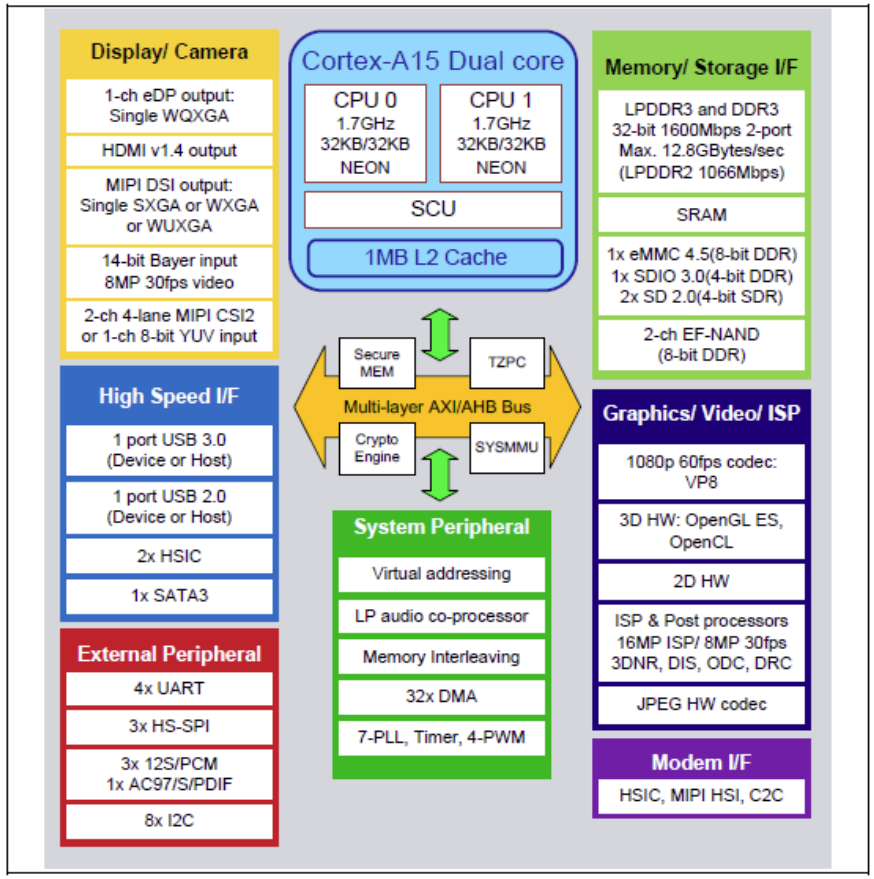


Figure 1-1 Block Diagram of Exynos 5250

Exynos 5 Dual Reference Manual, Fig. 1-1.

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

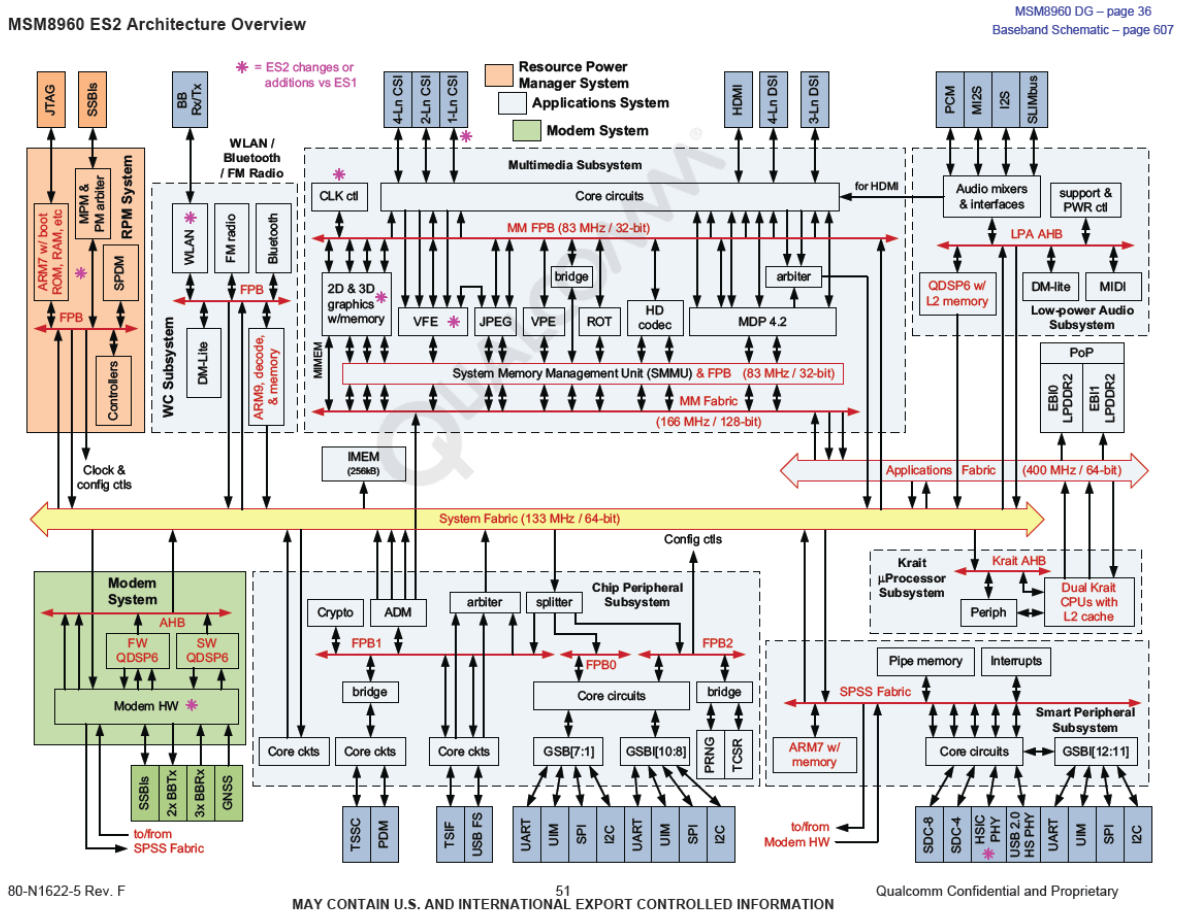
	 MSM8960 Chipset Schematics and Design Guidelines, p. 50.
said first and second plurality of stack elements and said central processing unit being provided in a single integrated circuit,	On information and belief, both the first and the second plurality of stack elements are the central processing unit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

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

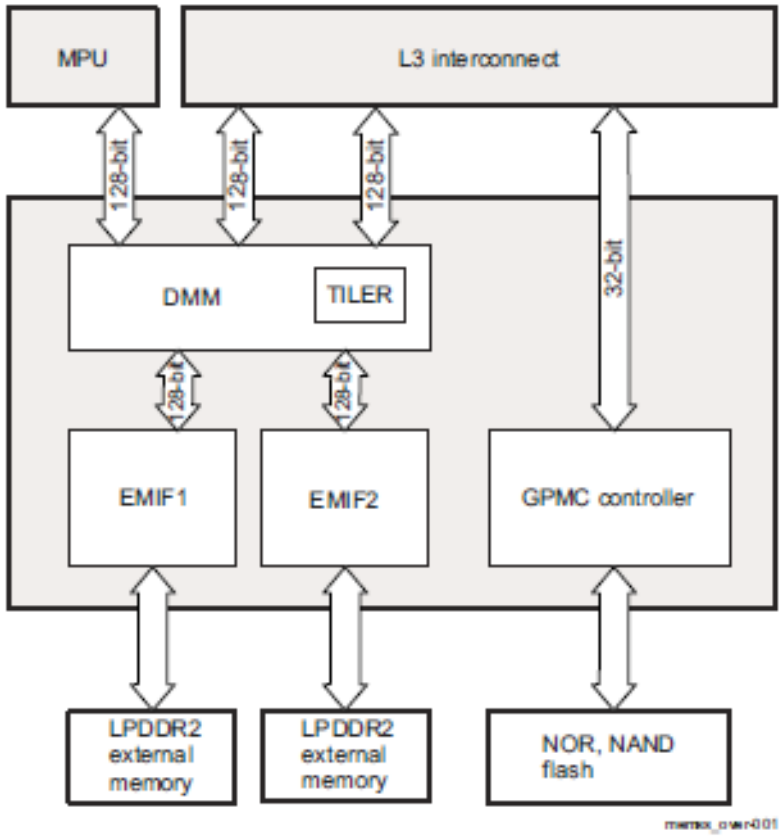
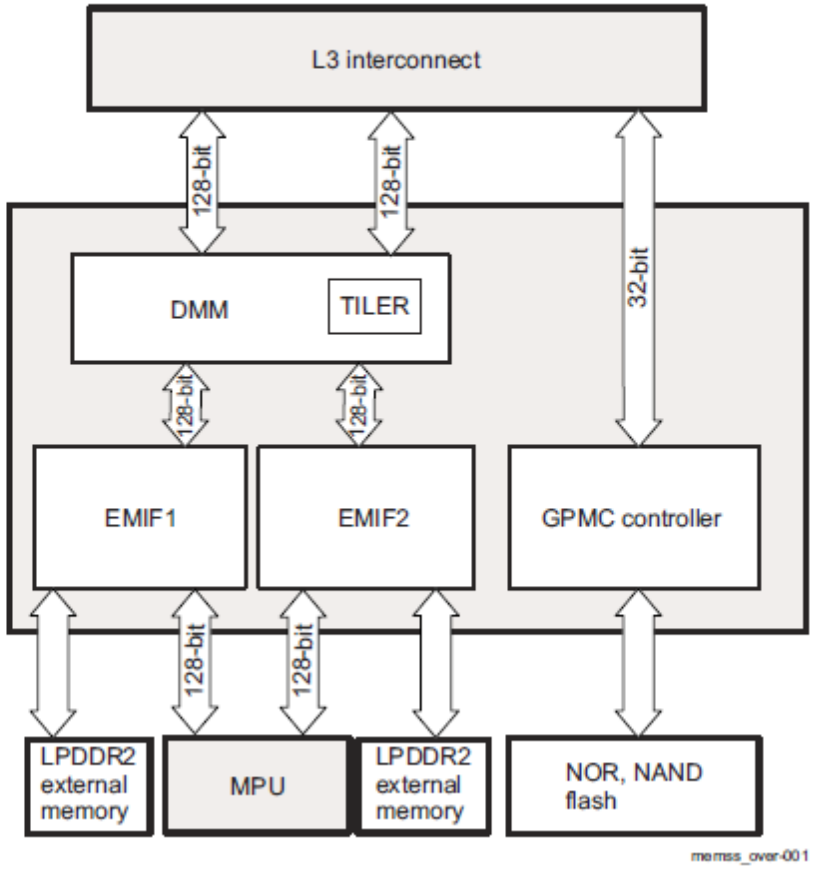
and a third plurality of stack elements configured as a random access memory external to said single integrated circuit.	On information and belief, a third plurality of stack elements is configured as external RAM (e.g, SDRAM). Figure 15-1 shows a functional diagram of all memory subsystems in the device.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

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

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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

1.3 Block Diagram

Figure 1-1 illustrates the block diagram of Exynos 5250.

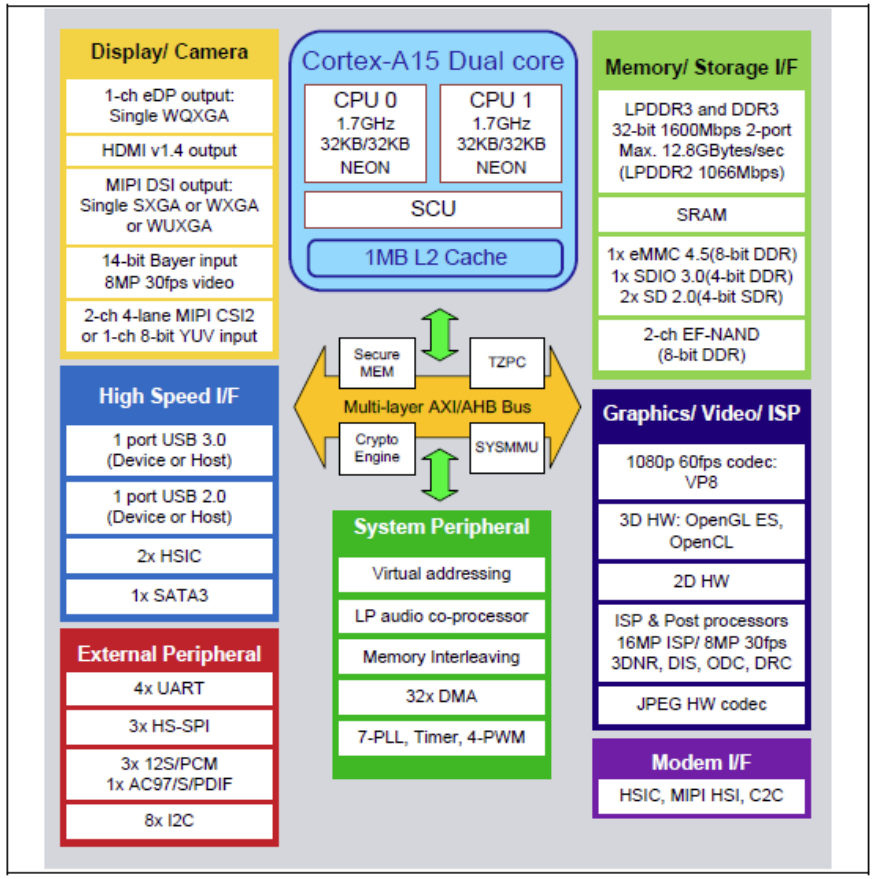
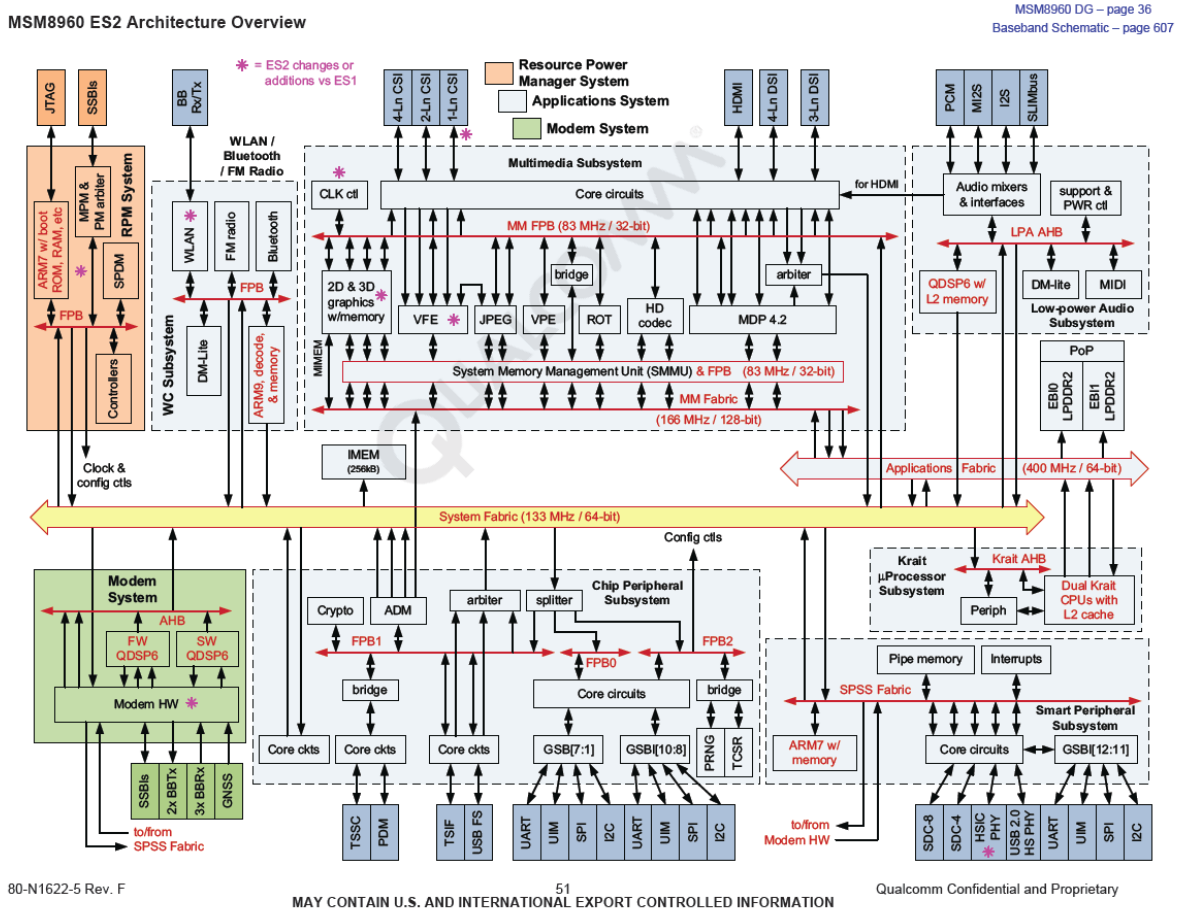


Figure 1-1 Block Diagram of Exynos 5250

Exynos 5 Dual Reference Manual, Fig. 1-1.

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



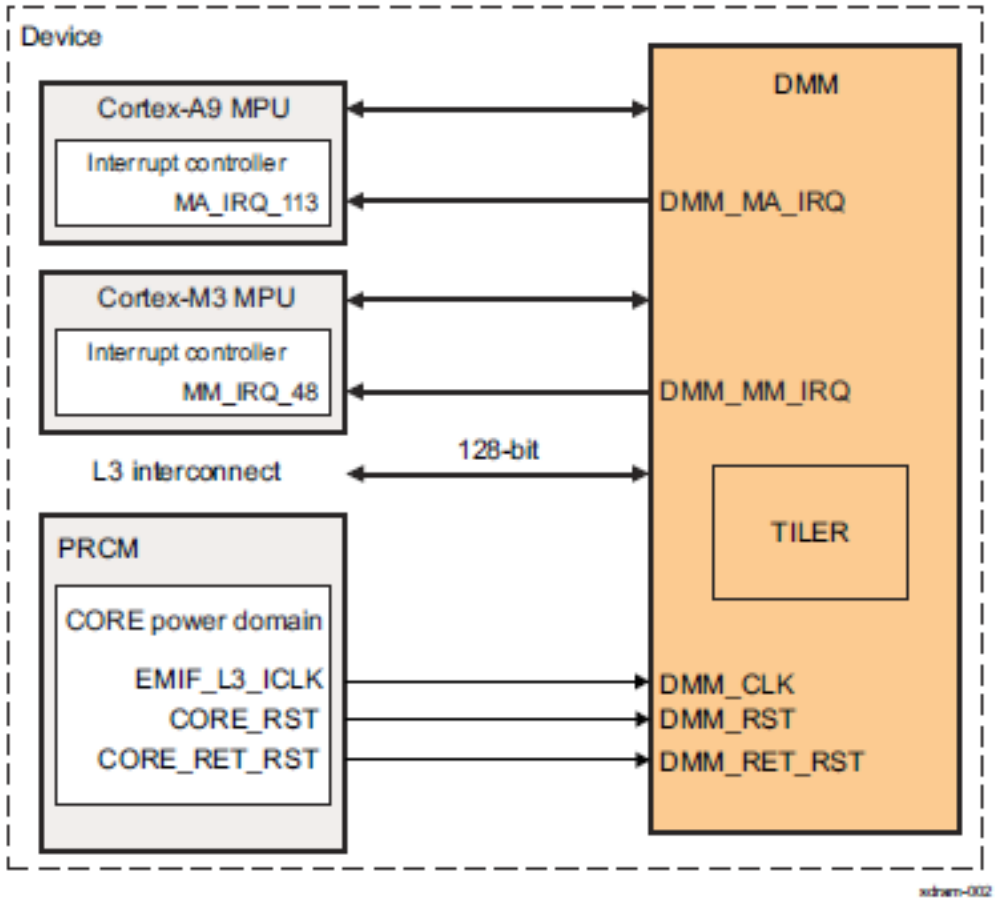
MSM8960 Chipset Schematics and Design Guidelines, p. 50.

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

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/Exynos; 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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

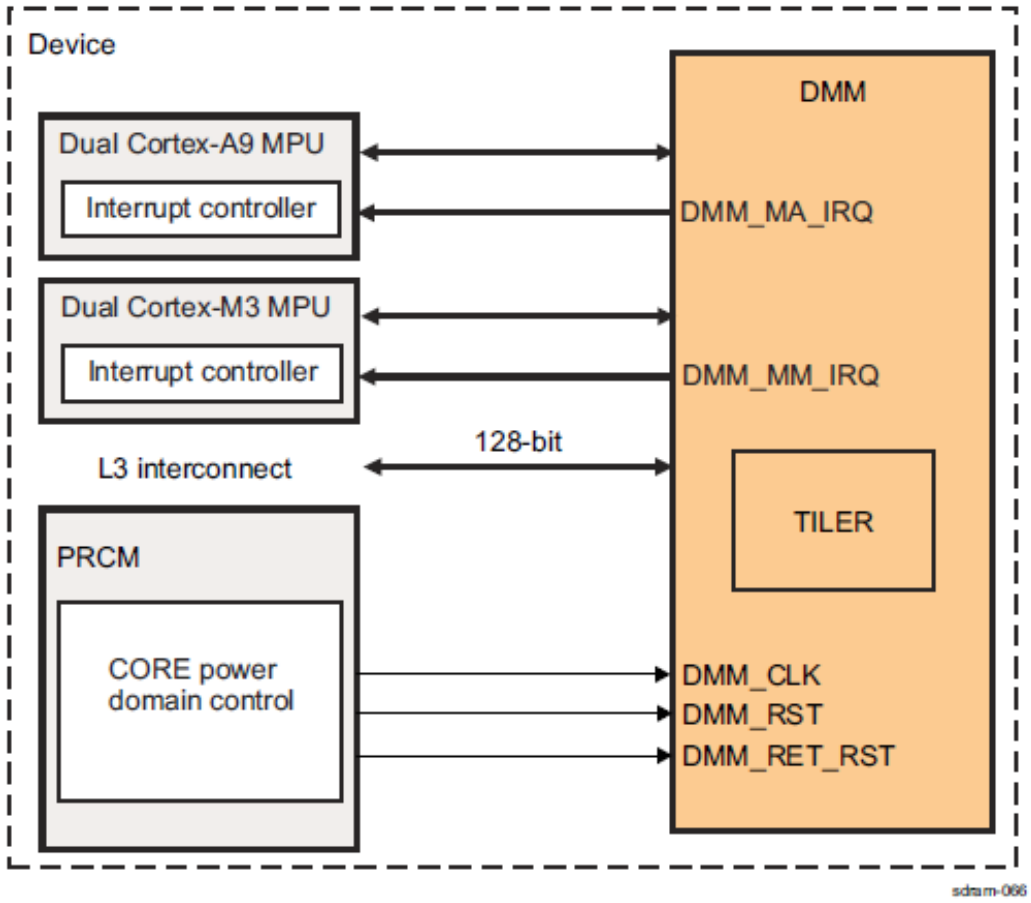
Figure 15-3. DMM Integration



OMAP4430 Technical Reference Manual, p. 3213.

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

Figure 16-2. DMM and Tiler Overview



OMAP4470 Technical Reference Manual, p. 3308.

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

1.3 Block Diagram

Figure 1-1 illustrates the block diagram of Exynos 5250.

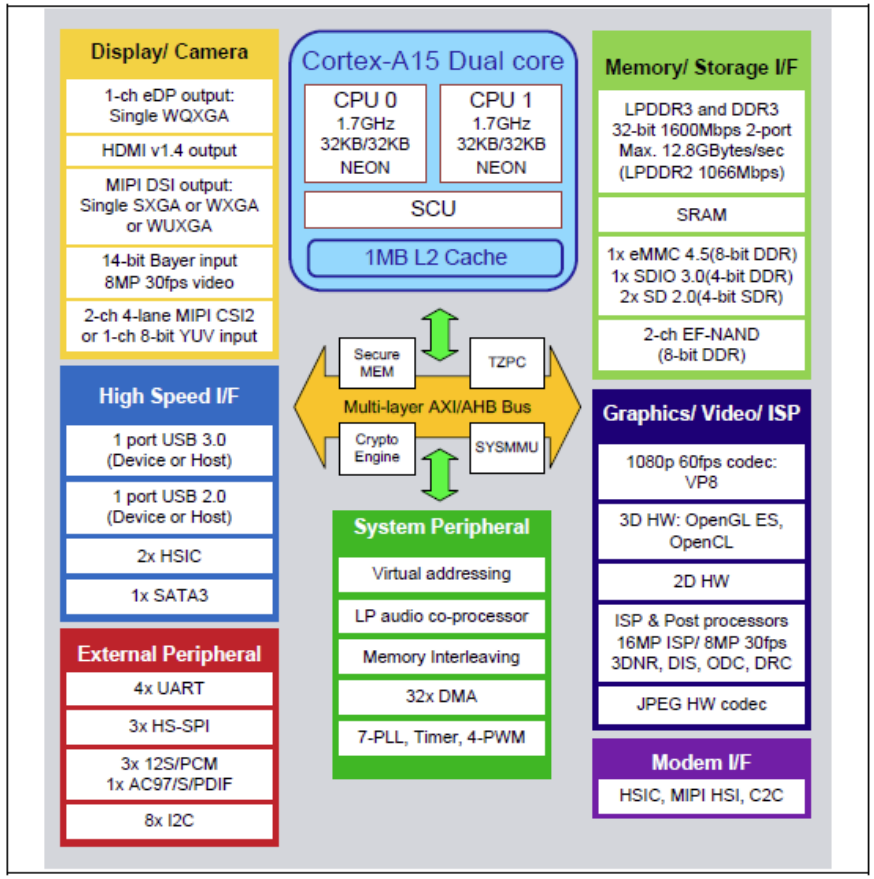


Figure 1-1 Block Diagram of Exynos 5250

Exynos 5 Dual Reference Manual, Fig. 1-1.

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

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

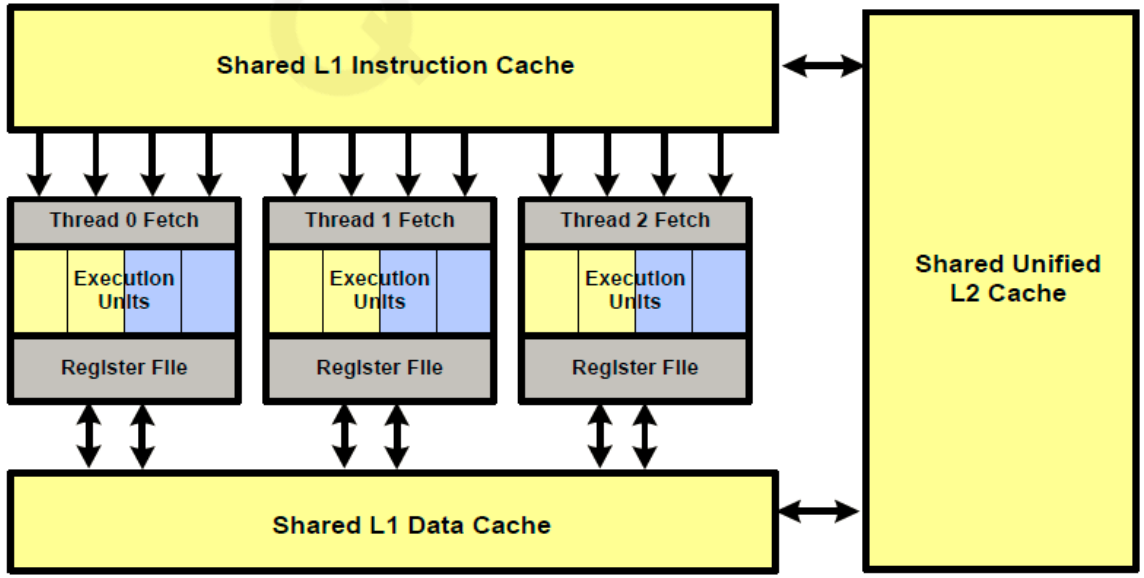
	central processing unit.  Hexagon Programming: Processor Overview, p. 10.

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

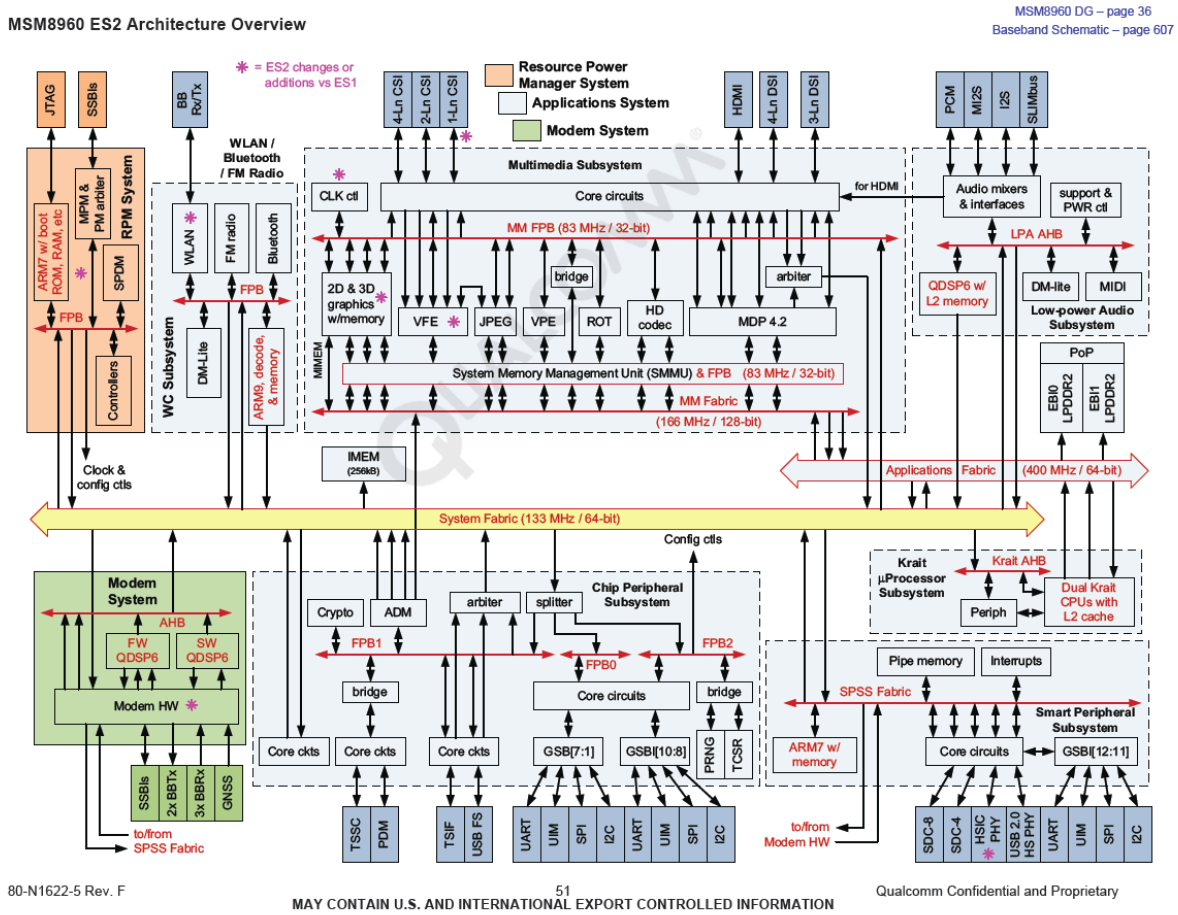
	 MSM8960 Chipset Schematics and Design Guidelines, p. 50. *See also.*, http://en.wikipedia.org/wiki/Qualcomm_Snapdragon; http://en.wikipedia.org/wiki/Exynos; http://en.wikipedia.org/wiki/OMAP.
said main central processing unit having an arithmetic logic unit,	ARM cores, such as those contained within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

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

	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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

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.

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

	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, the Accused Microprocessors also include 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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

	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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

	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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

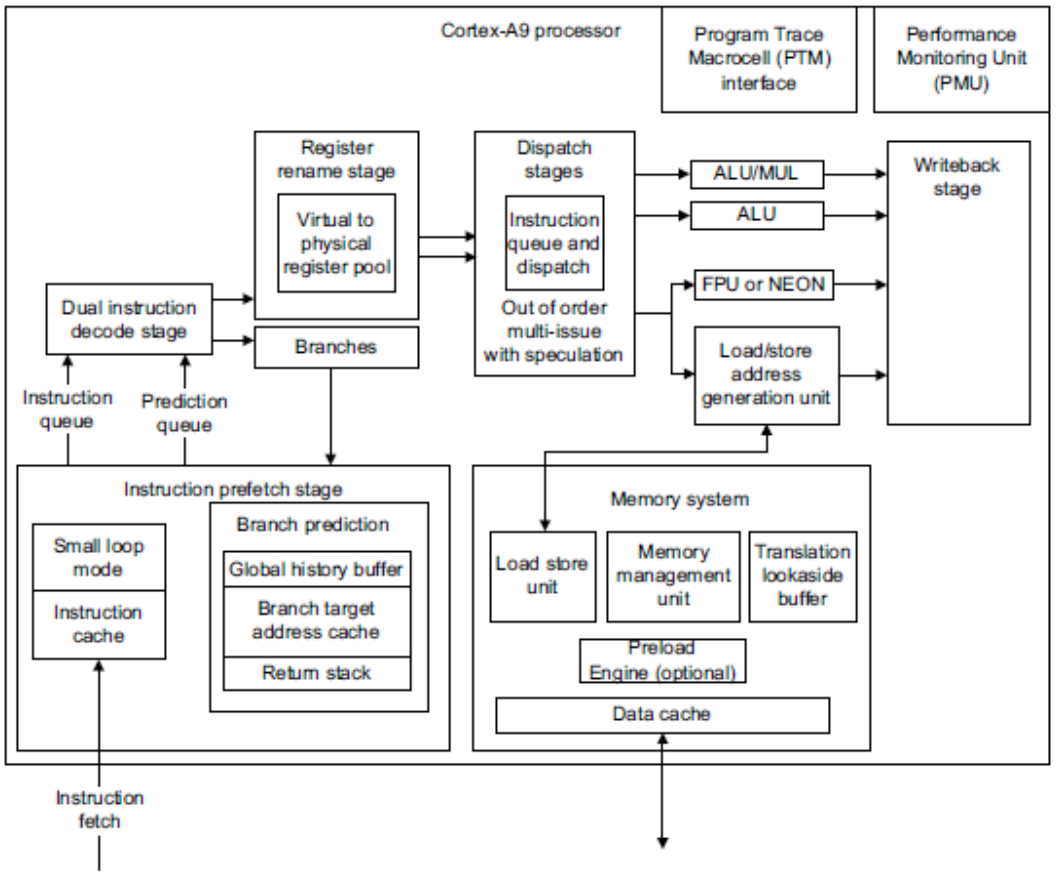
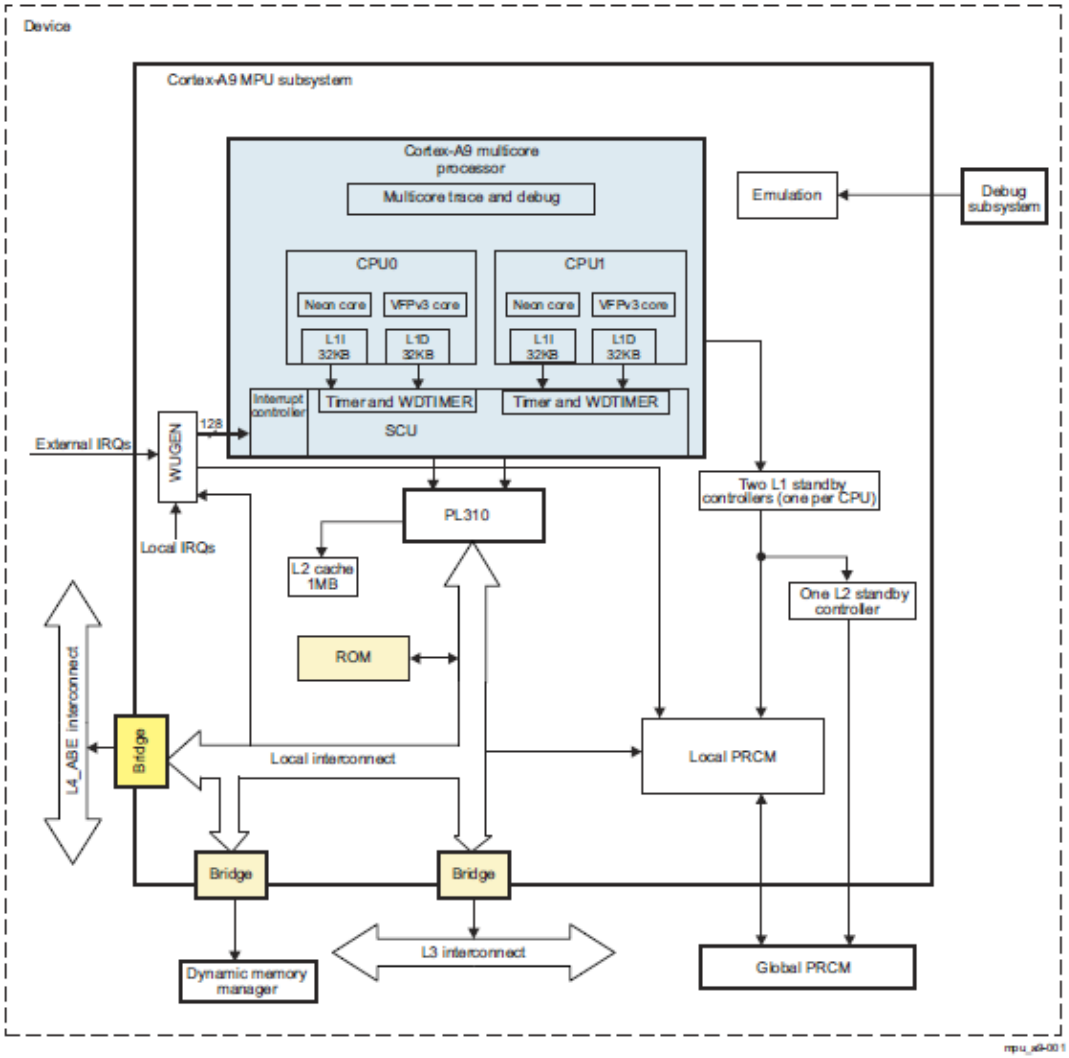


Figure 2-1 Cortex-A9 processor top-level diagram

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

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

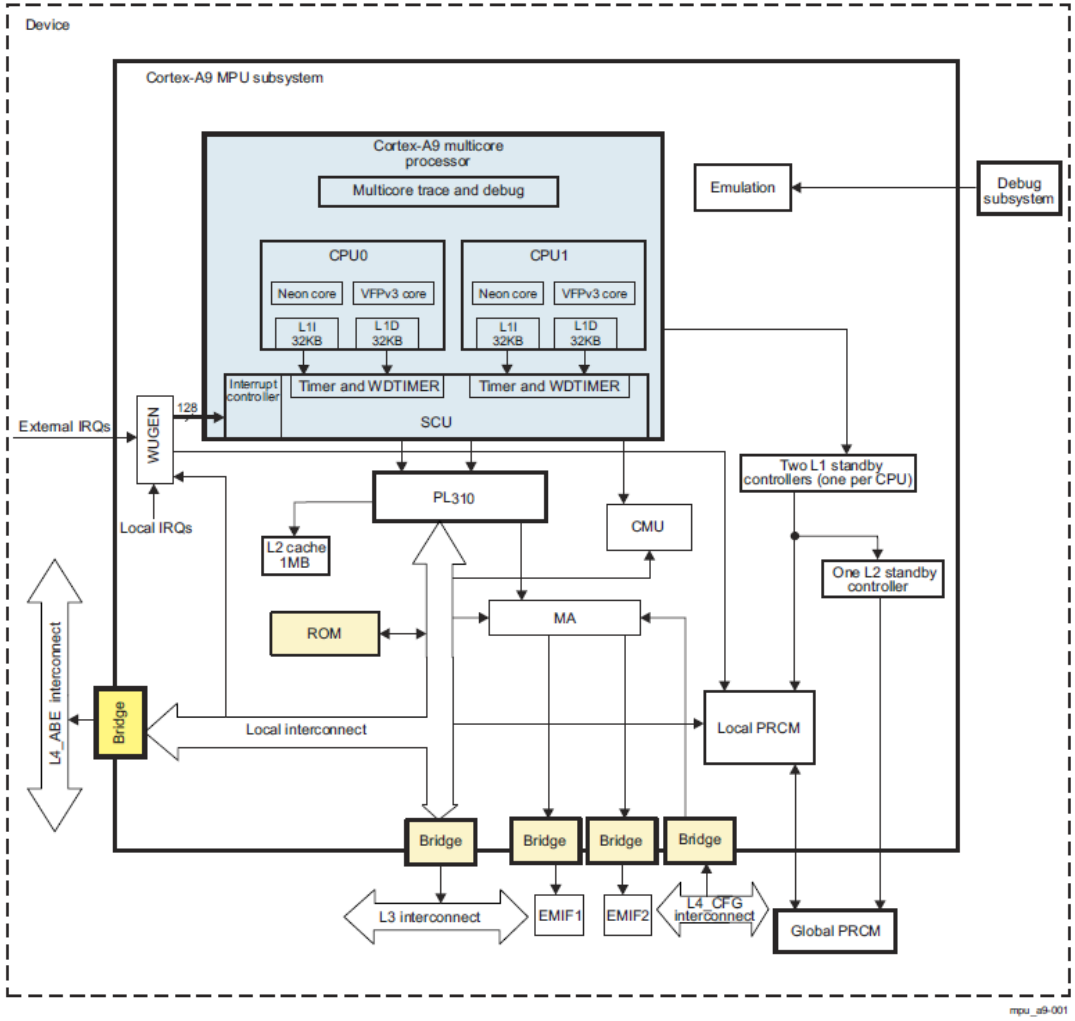
Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4430 Technical Reference Manual, p. 1077.

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

Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4470 Technical Reference Manual, p. 1091.

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

1.3 Block Diagram

Figure 1-1 illustrates the block diagram of Exynos 5250.

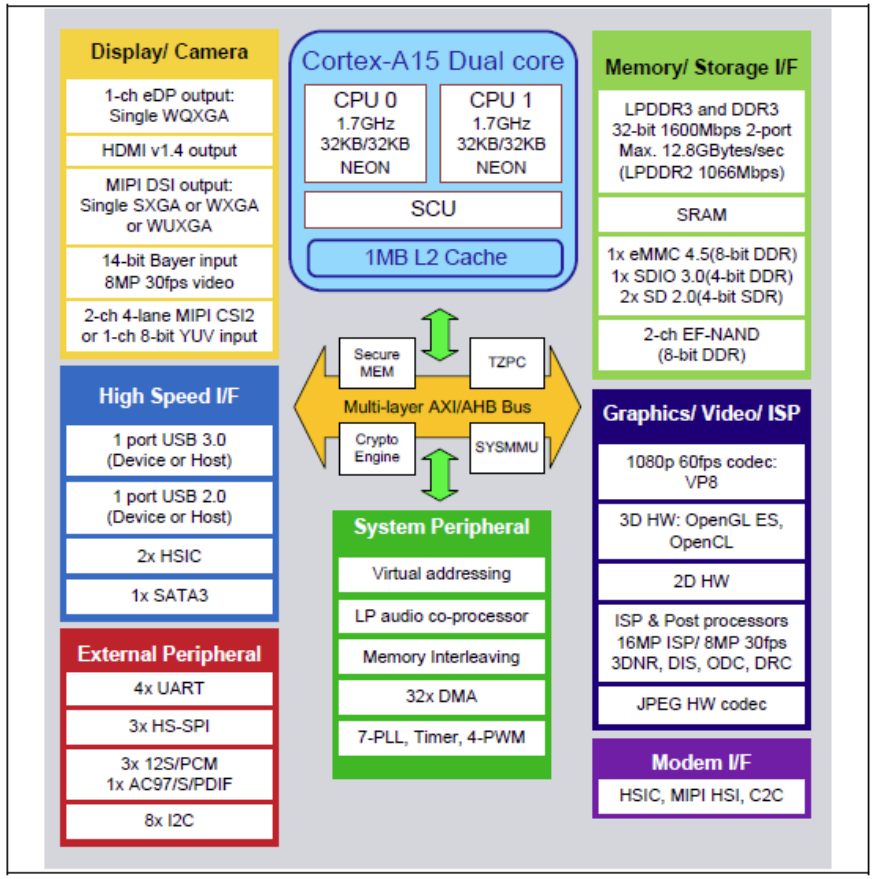
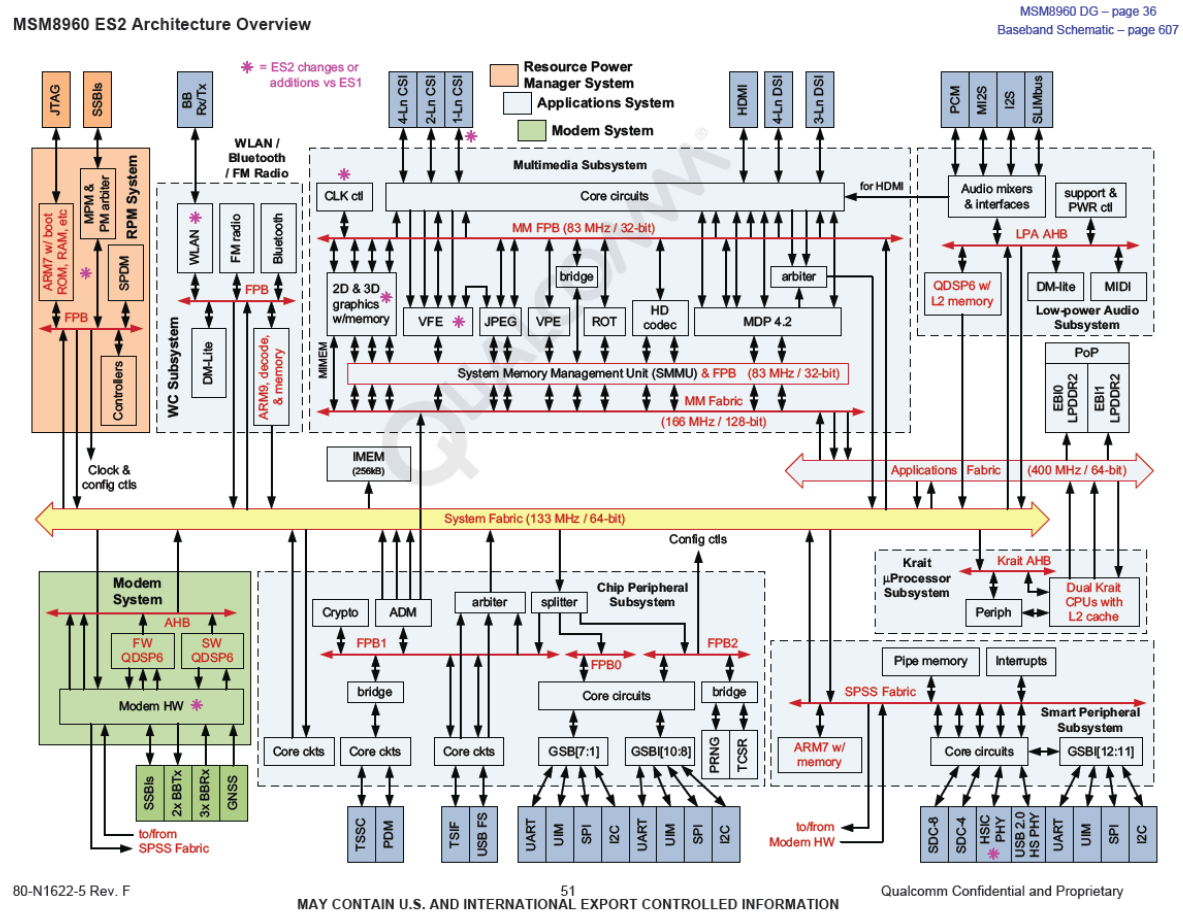


Figure 1-1 Block Diagram of Exynos 5250

Exynos 5 Dual Reference Manual, Fig. 1-1.

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



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 OMAP4430's or OMAP4470's L3 interconnect, the MPU local interconnect, and other internal buses.

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

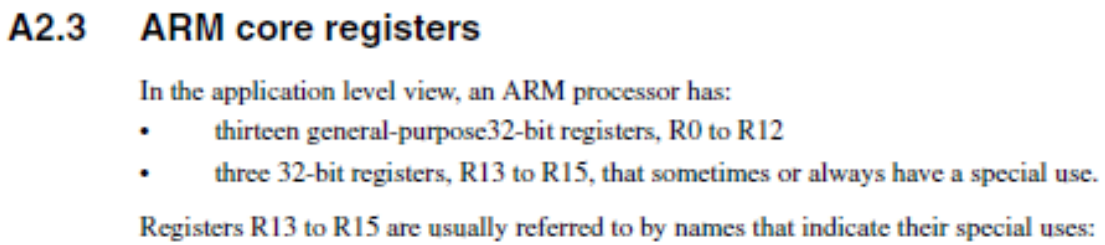
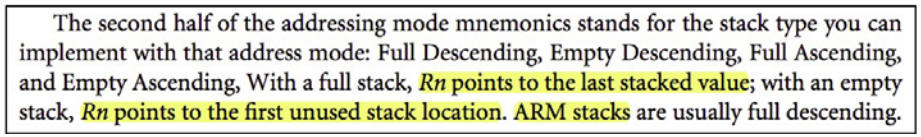
	Plaintiffs contend that this claim element is literally present as described above. In the event that this claim element is not found to be literally present, Plaintiffs contend that the above identified buses, either alone or in combination, are equivalent to "an internal data bus" and any differences 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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

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

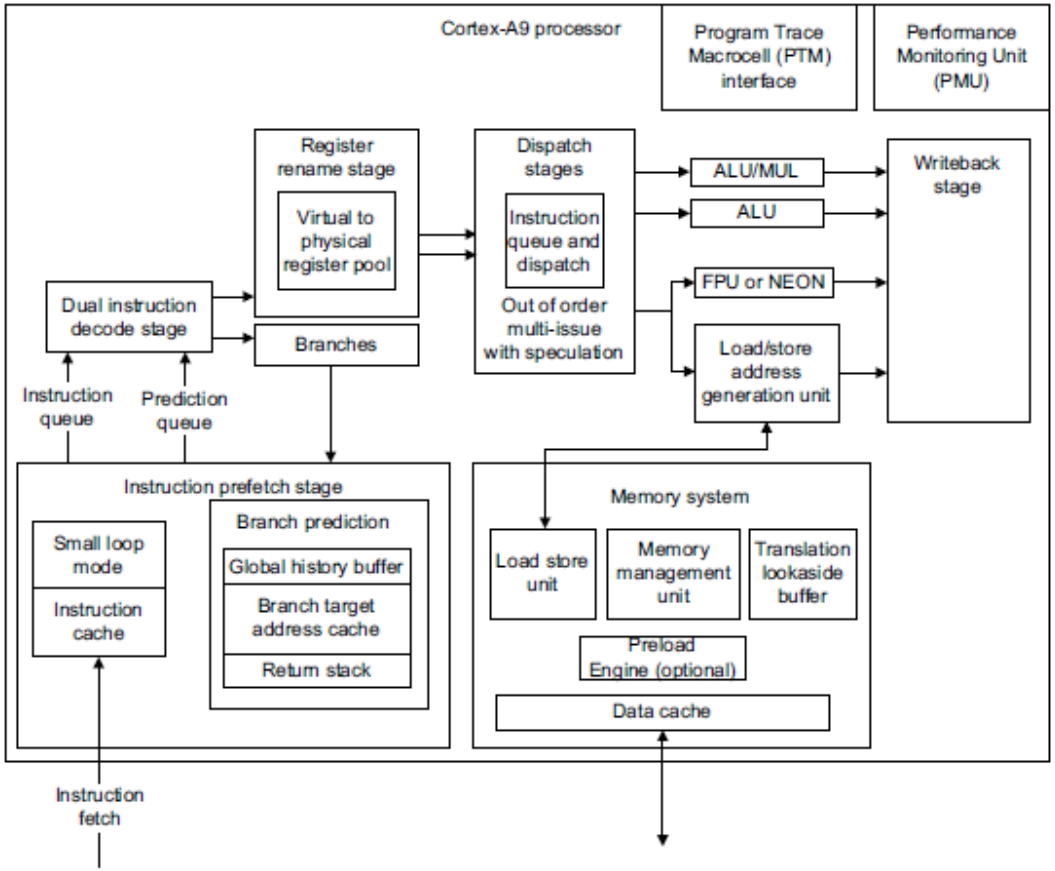
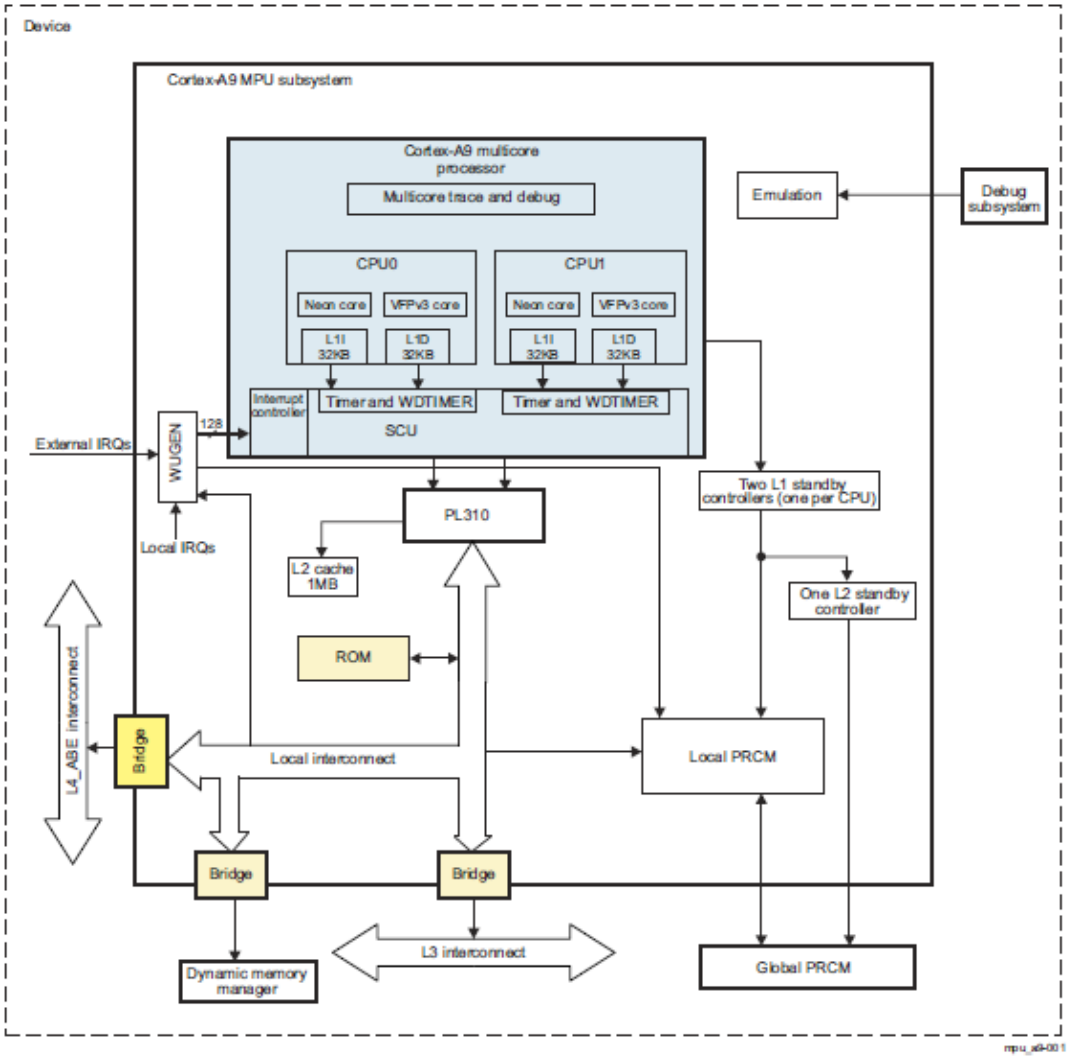


Figure 2-1 Cortex-A9 processor top-level diagram

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

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

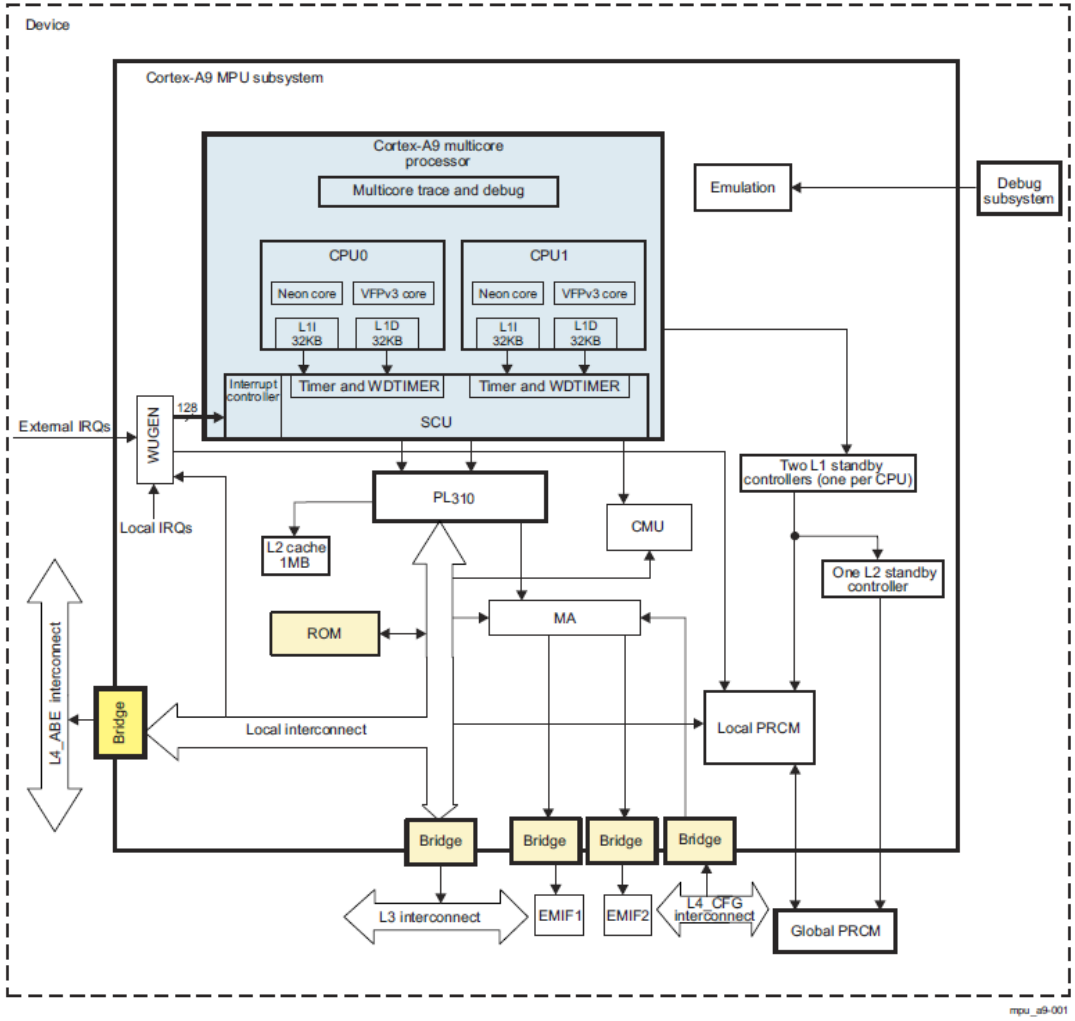
Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4430 Technical Reference Manual, p. 1077.

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

Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4470 Technical Reference Manual, p. 1091.

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

1.3 Block Diagram

Figure 1-1 illustrates the block diagram of Exynos 5250.

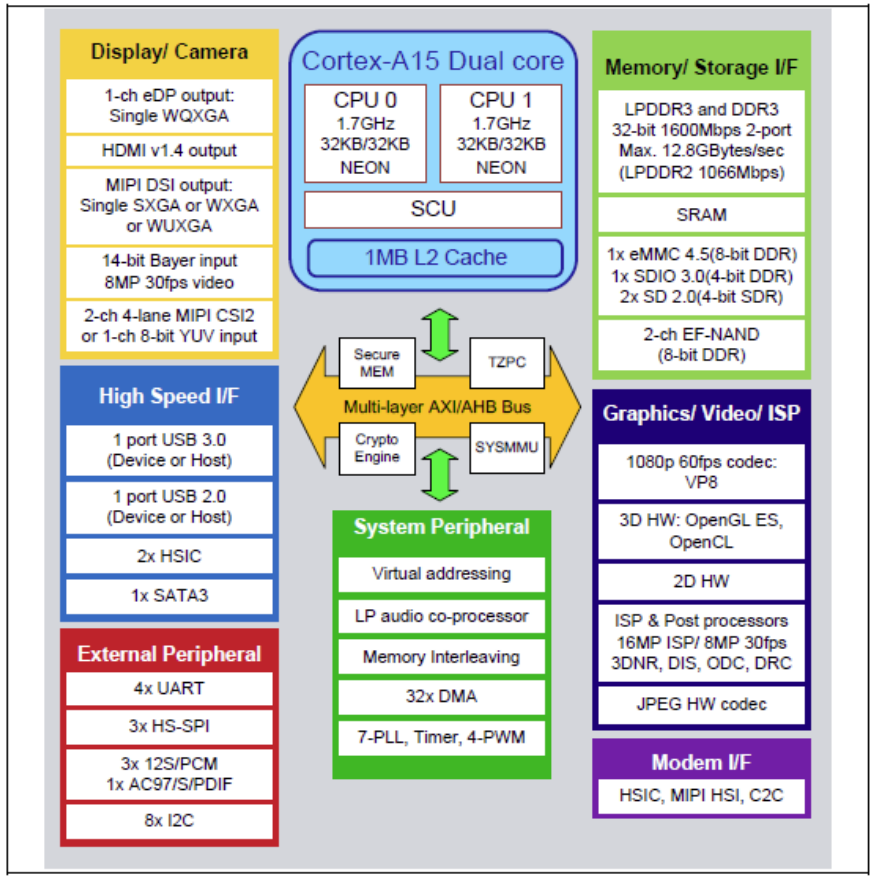


Figure 1-1 Block Diagram of Exynos 5250

Exynos 5 Dual Reference Manual, Fig. 1-1.

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

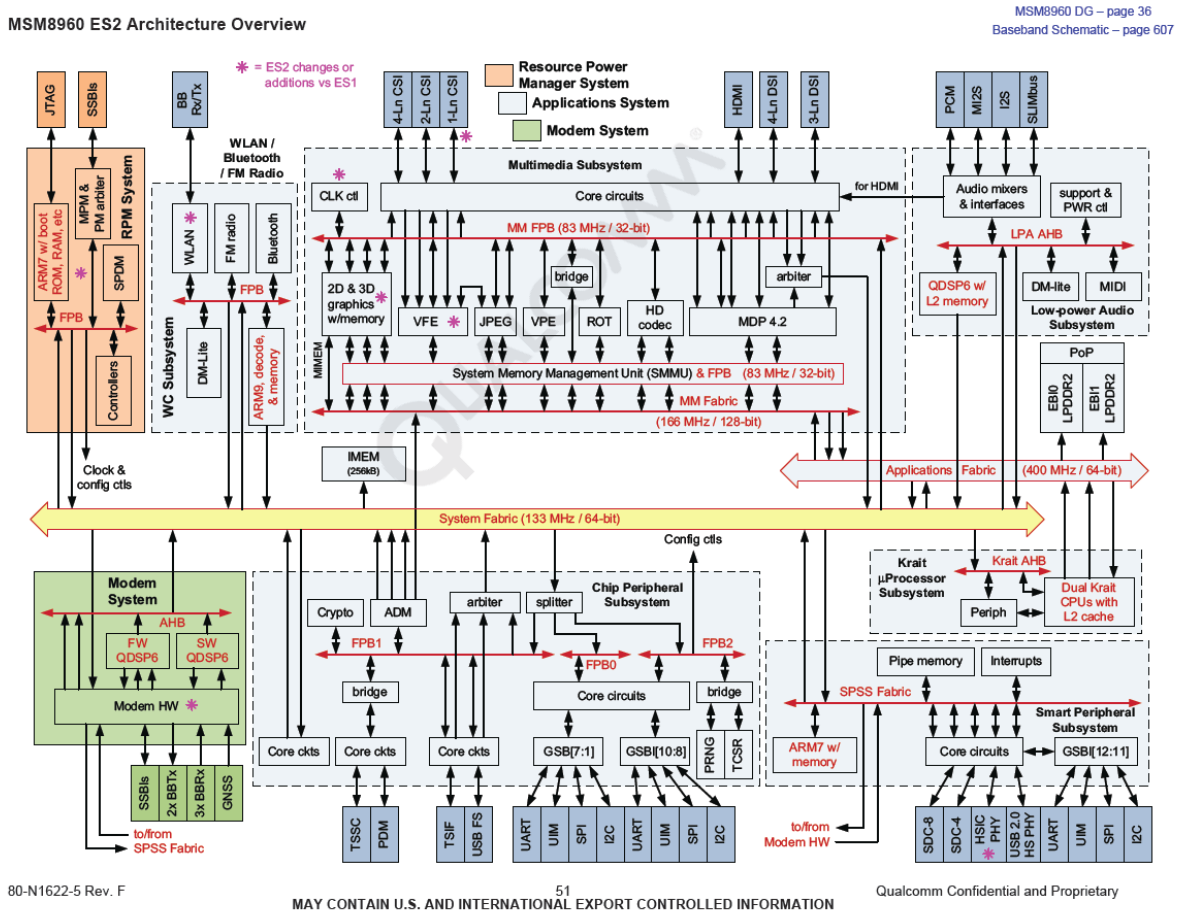
	 MSM8960 Chipset Schematics and Design Guidelines, p. 50. On information and belief, when executing Jazelle instructions within a loop, the processor maintains a loop counter that is bidirectionally connected to the Accused Microprocessor's internal buses.
said loop counter being connected to	Each ARM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

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

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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

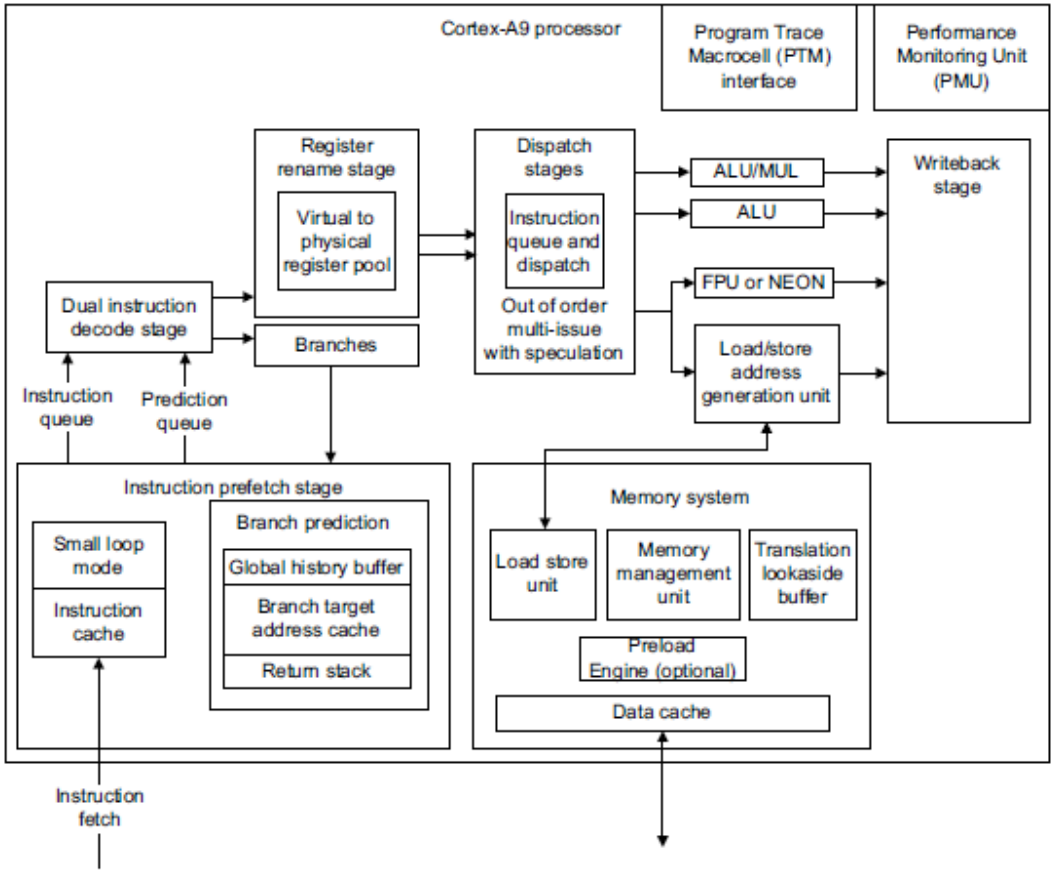
	 **Figure 2-1 Cortex-A9 processor top-level diagram** Cortex-A9 Technical Reference Manual, p.2-2.
said internal data bus being bidirectionally connected to a stack pointer, return stack pointer, mode register and instruction register,	Each ARM core has a stack pointer.

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

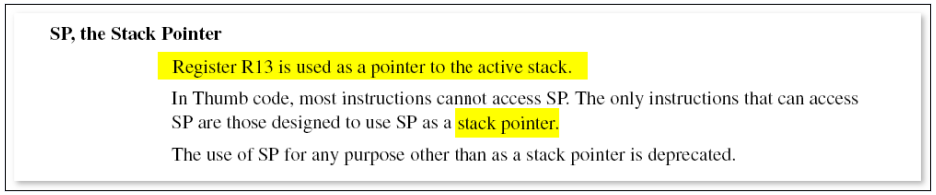
	 **SP, the Stack Pointer** Register R13 is used as a pointer to the active stack. In Thumb code, most instructions cannot access SP. The only instructions that can access SP are those designed to use SP as a stack pointer. The use of SP for any purpose other than as a stack pointer is deprecated. ARM 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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

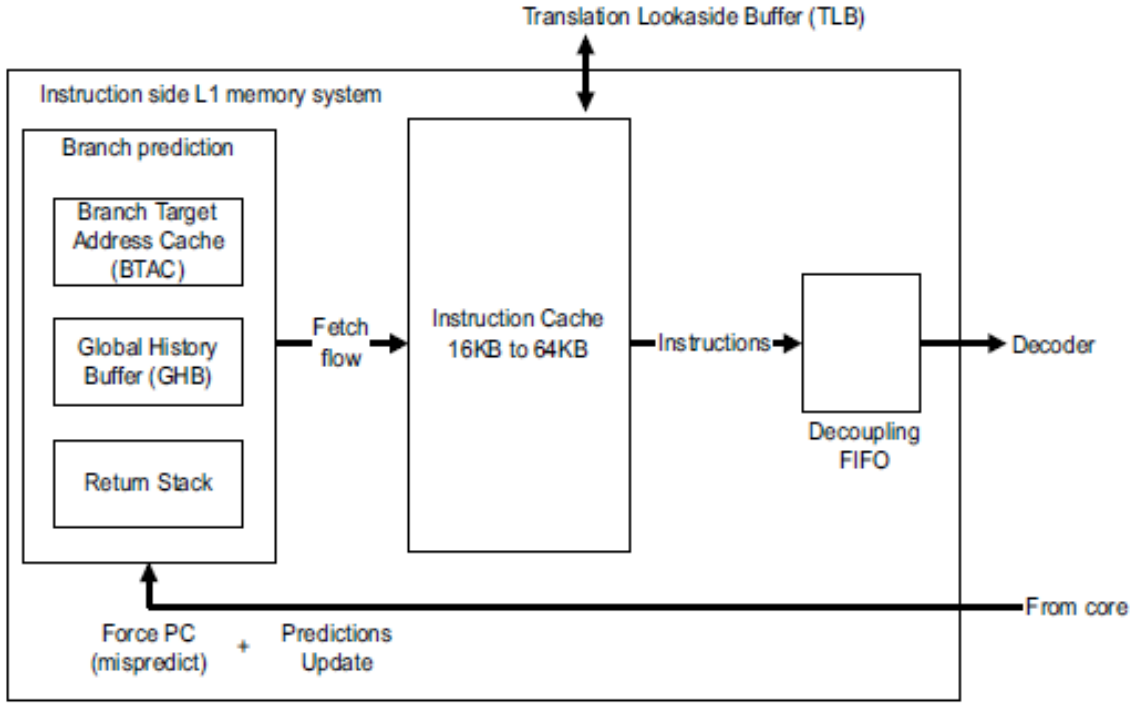


Figure 6-1 Branch prediction and instruction cache

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

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

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

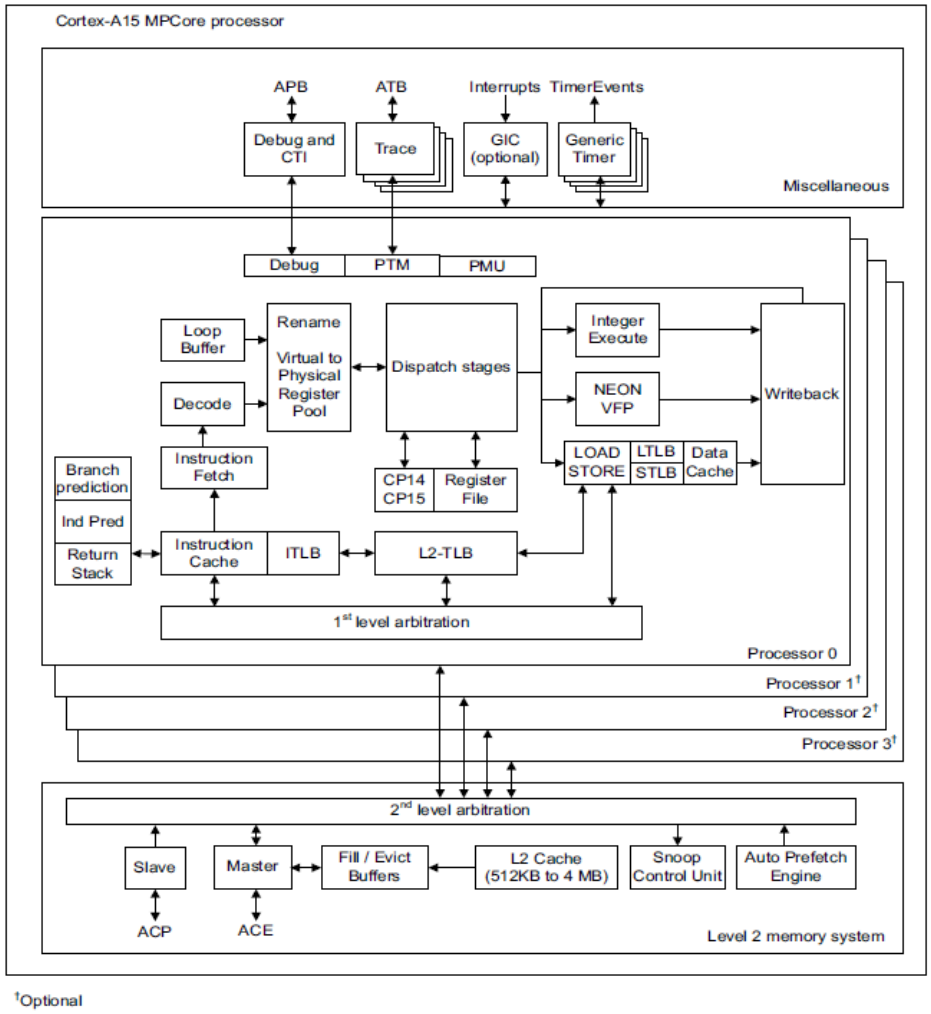


Figure 2-1 Block diagram

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

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

	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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

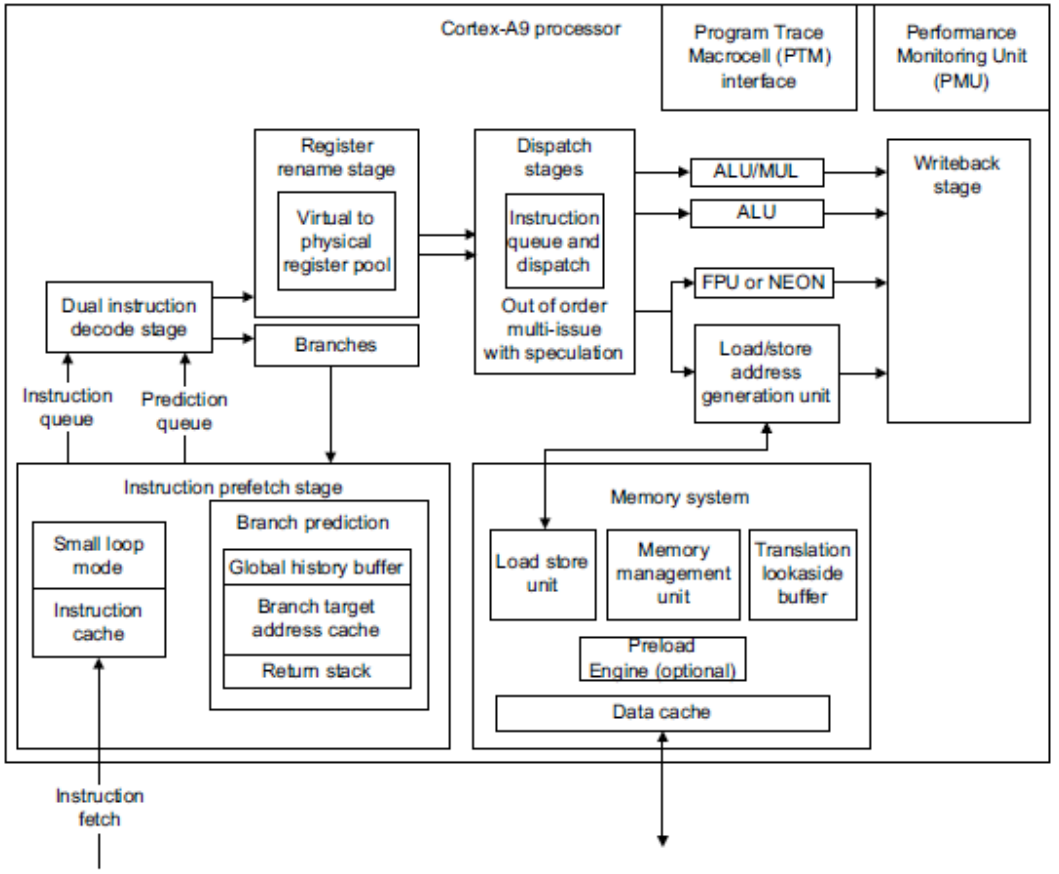
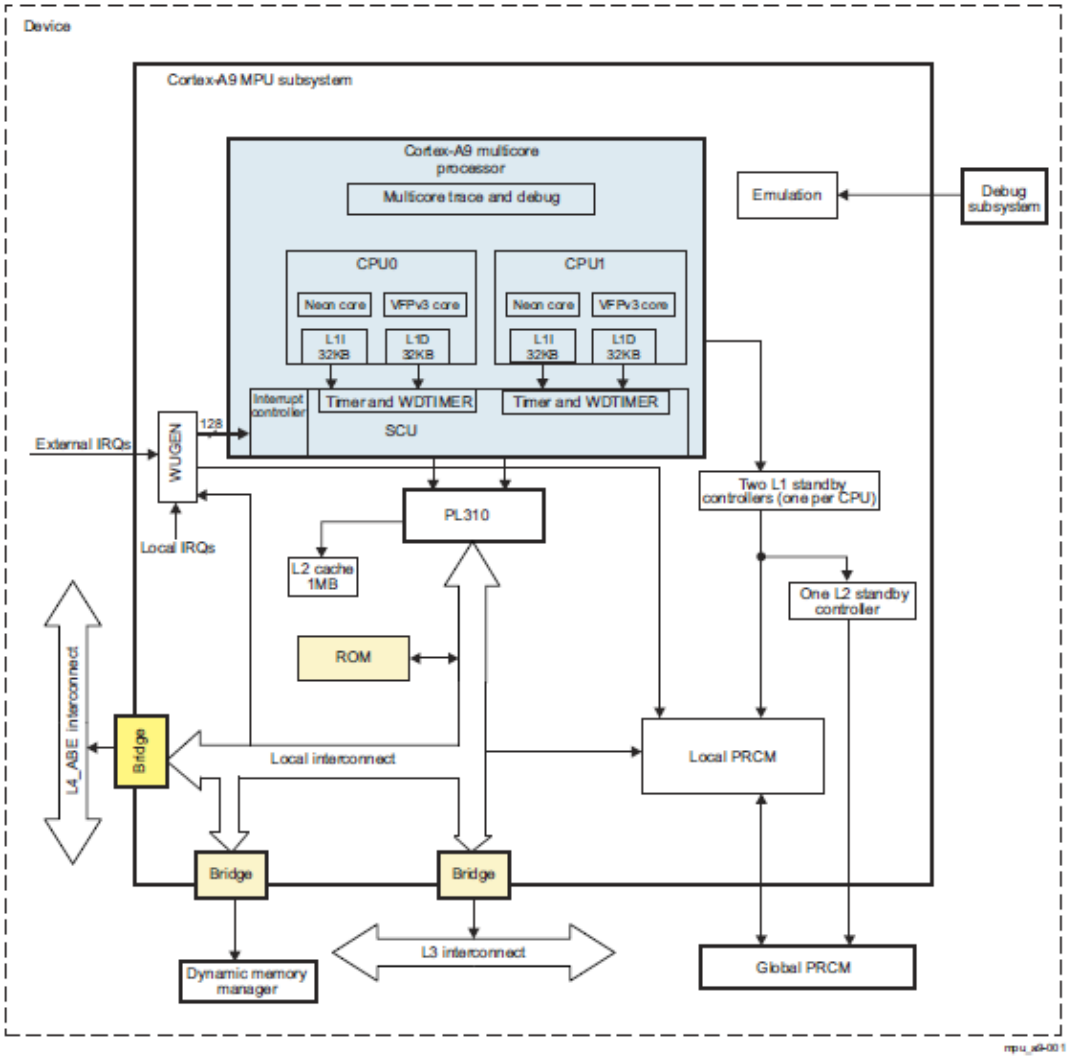


Figure 2-1 Cortex-A9 processor top-level diagram

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

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

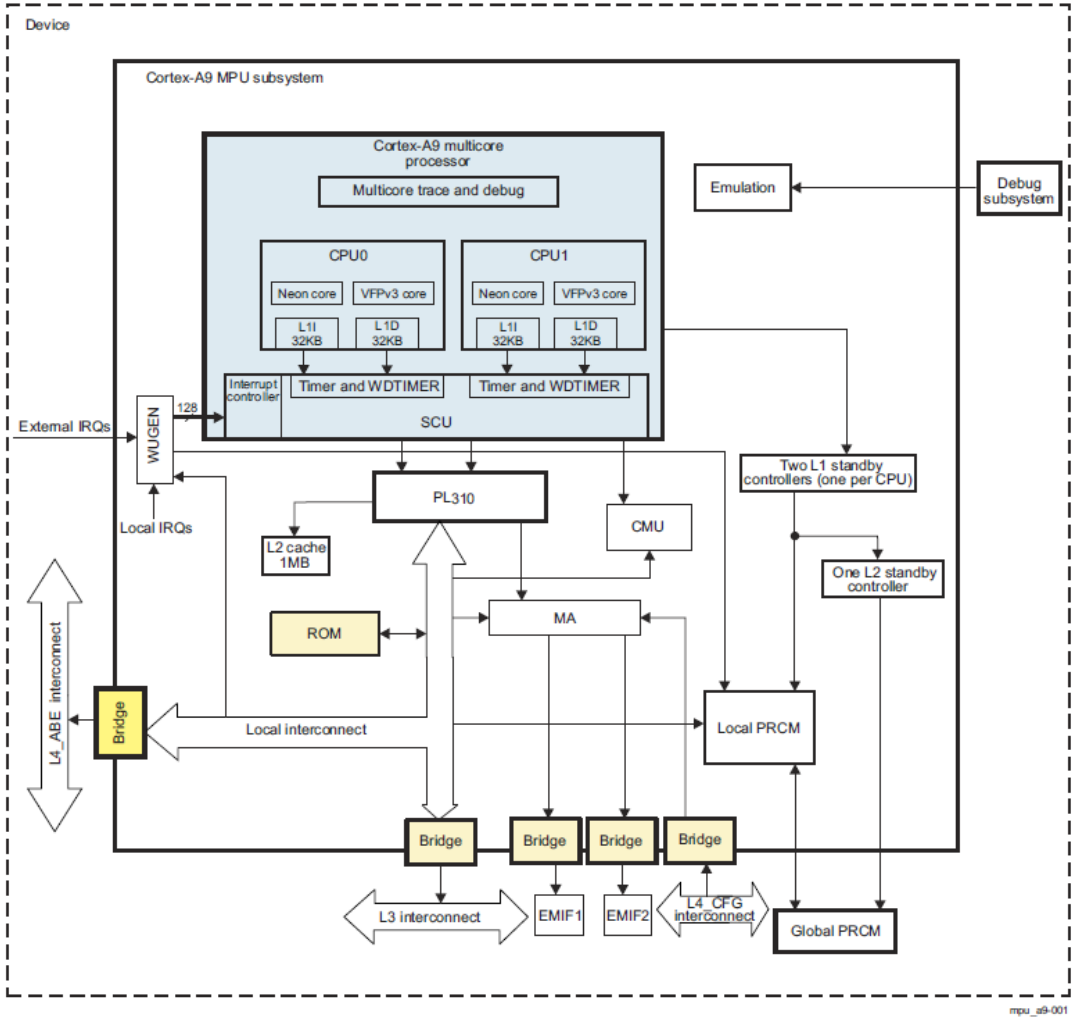
Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4430 Technical Reference Manual, p. 1077.

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

Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4470 Technical Reference Manual, p. 1091.

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

1.3 Block Diagram

Figure 1-1 illustrates the block diagram of Exynos 5250.

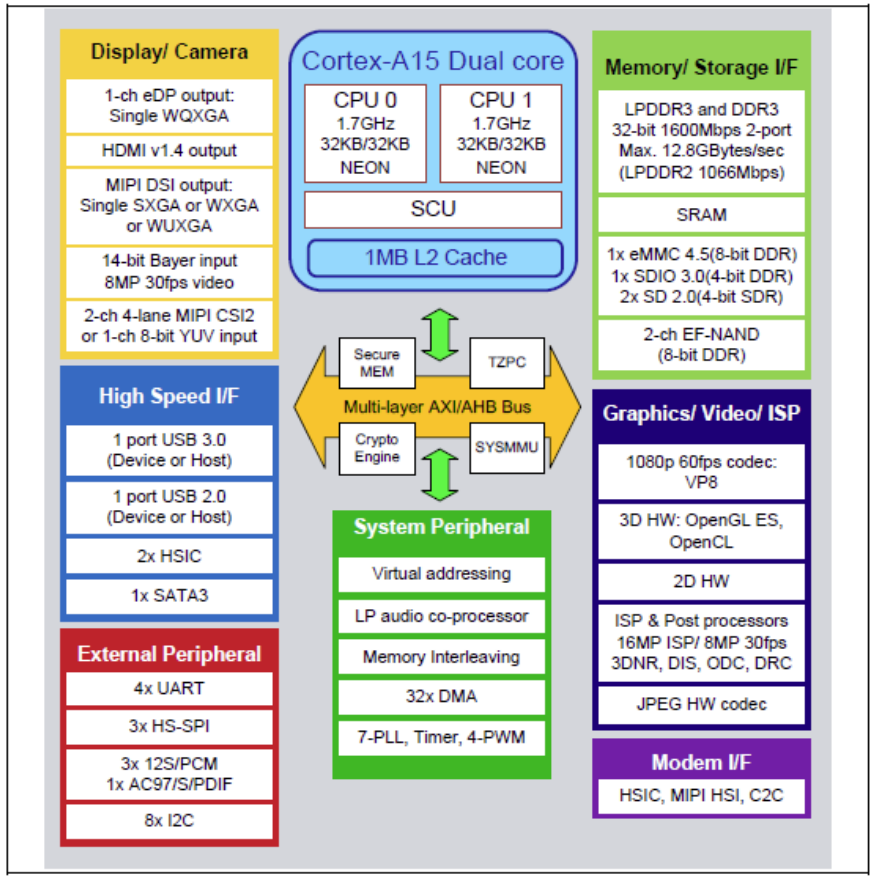
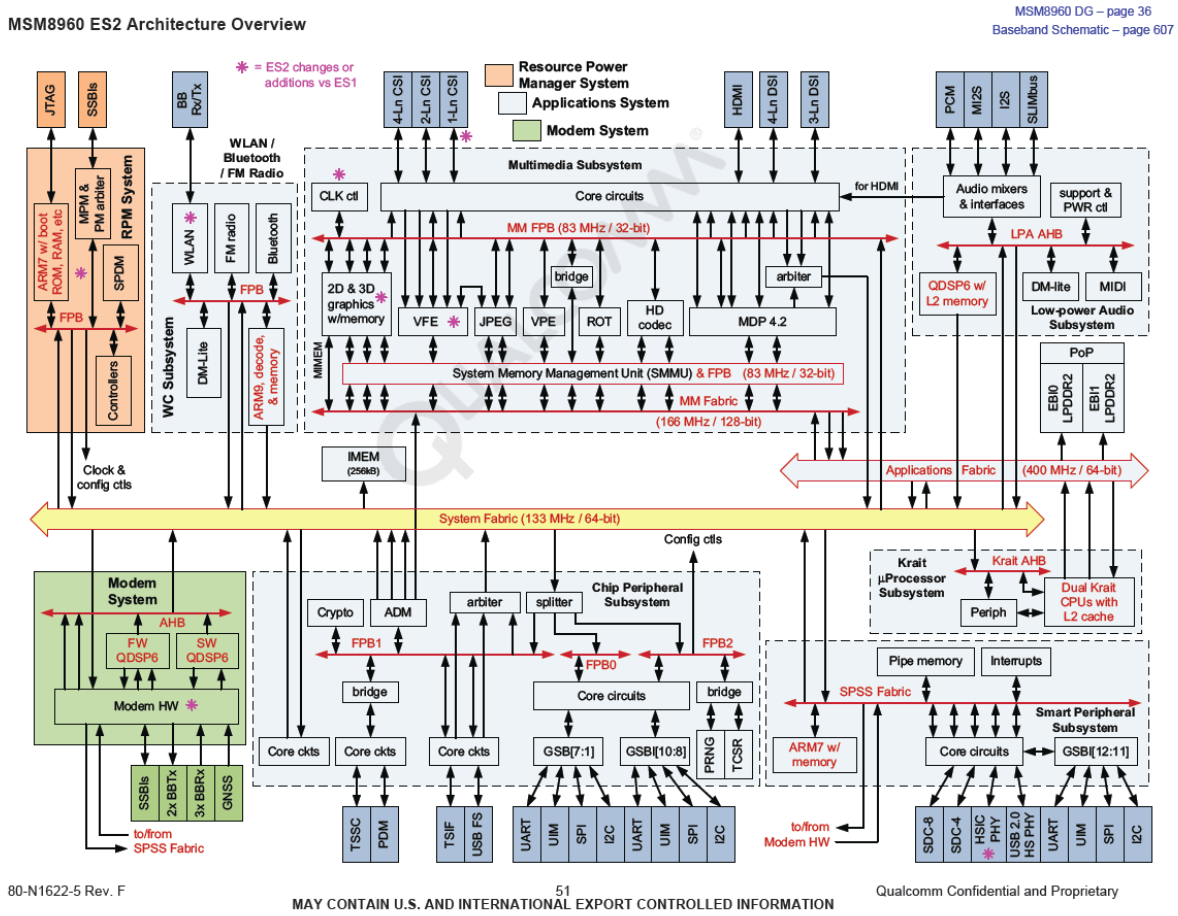


Figure 1-1 Block Diagram of Exynos 5250

Exynos 5 Dual Reference Manual, Fig. 1-1.

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



MSM8960 Chipset Schematics and Design Guidelines, p. 50.

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

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

	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.

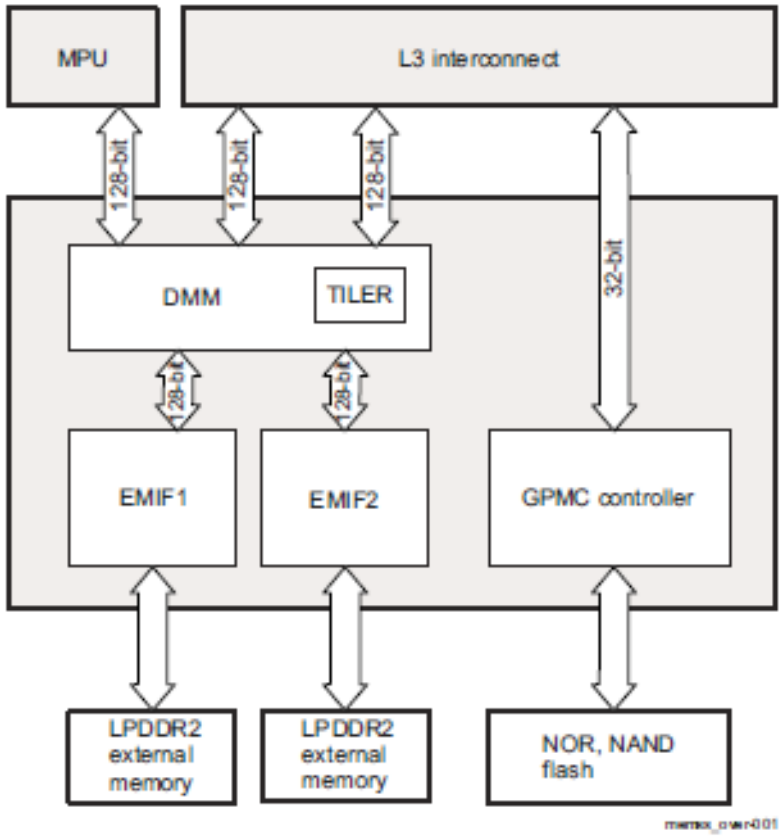
EXHIBIT E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

	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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

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

Figure 15-1. Memory Subsystem Functional Diagram

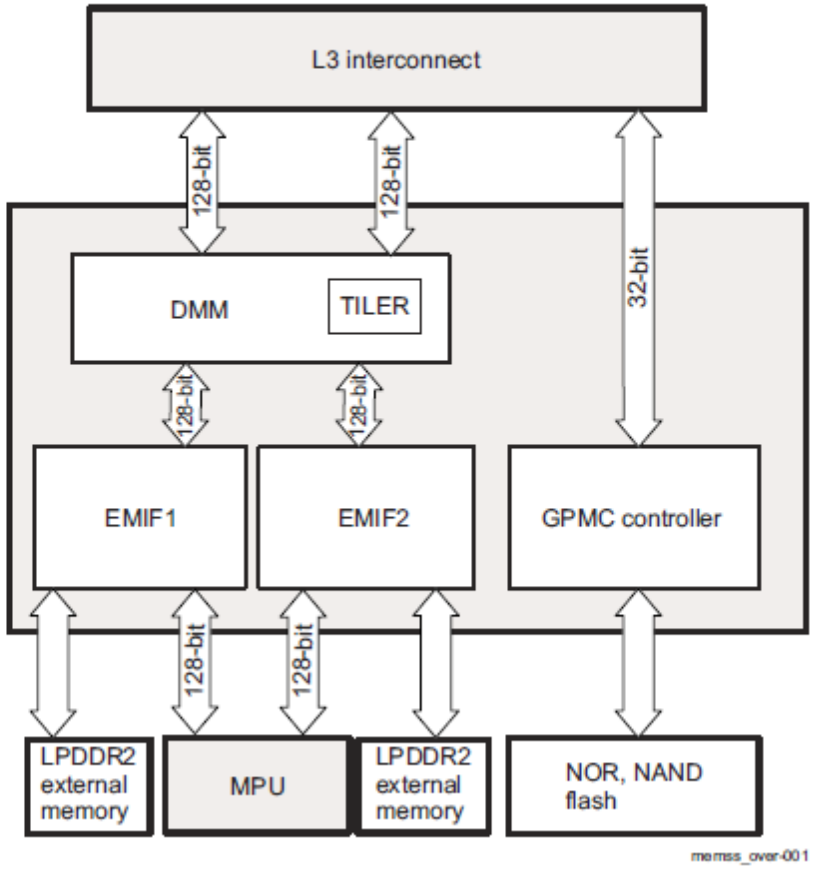


OMAP4430 Technical Reference Manual, p. 3207.

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

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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

1.3 Block Diagram

Figure 1-1 illustrates the block diagram of Exynos 5250.

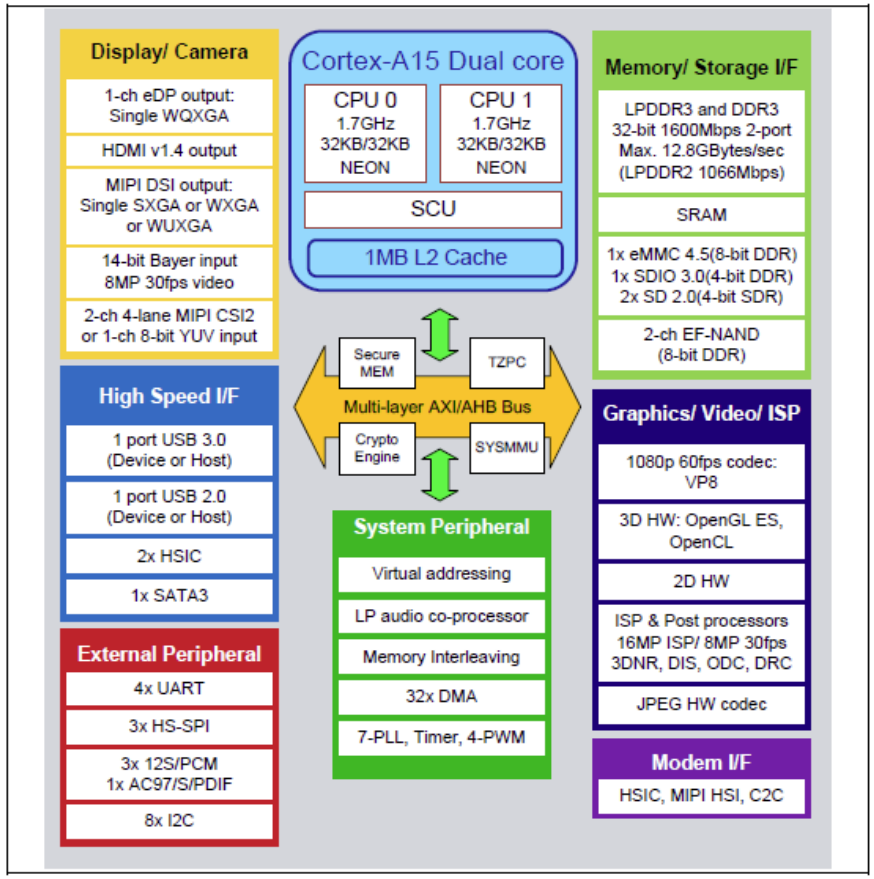
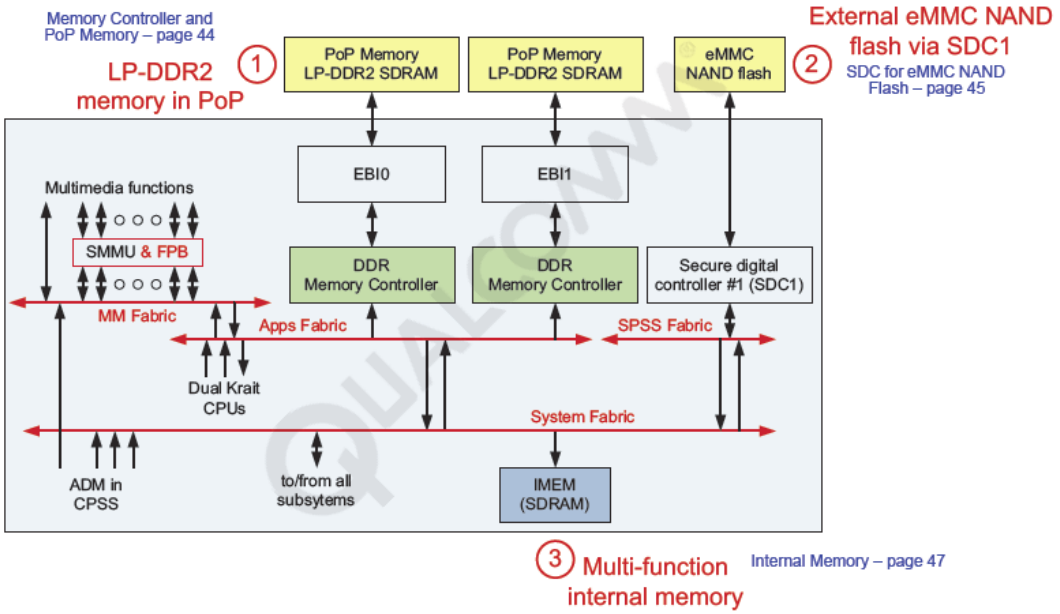


Figure 1-1 Block Diagram of Exynos 5250

Exynos 5 Dual Reference Manual, Fig. 1-1.

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



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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

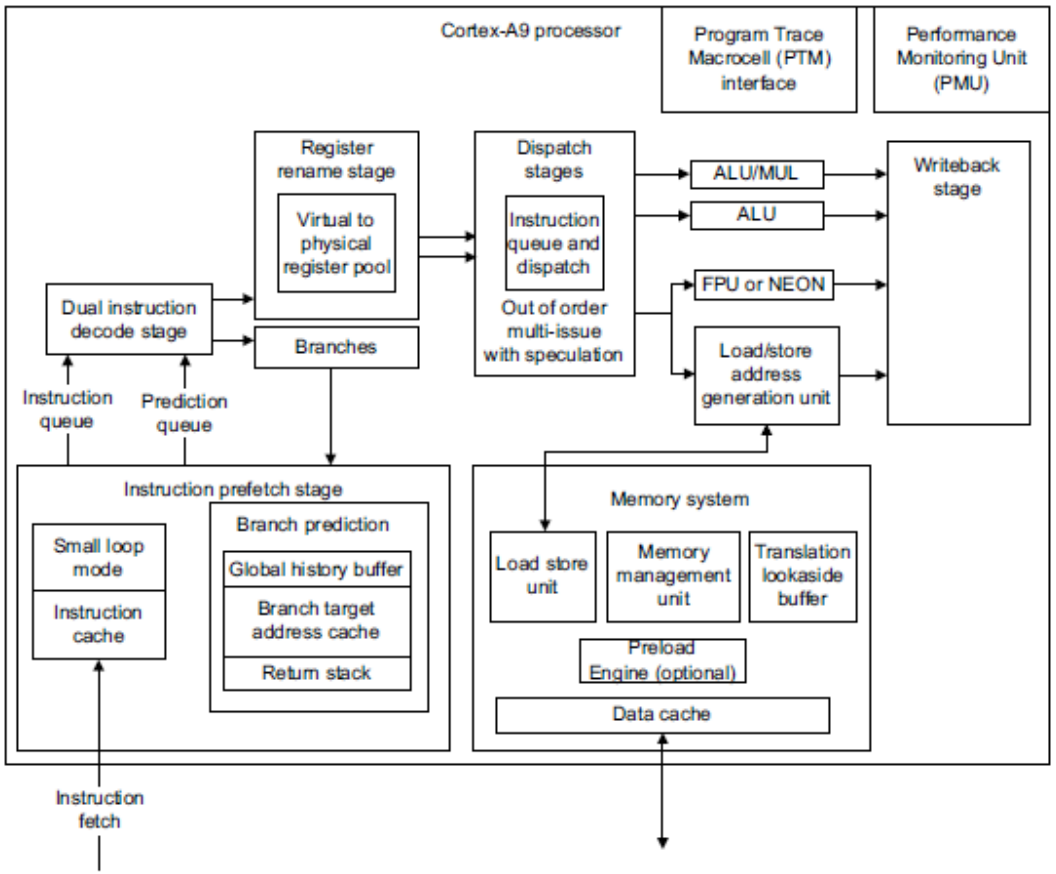


Figure 2-1 Cortex-A9 processor top-level diagram

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

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

	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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

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

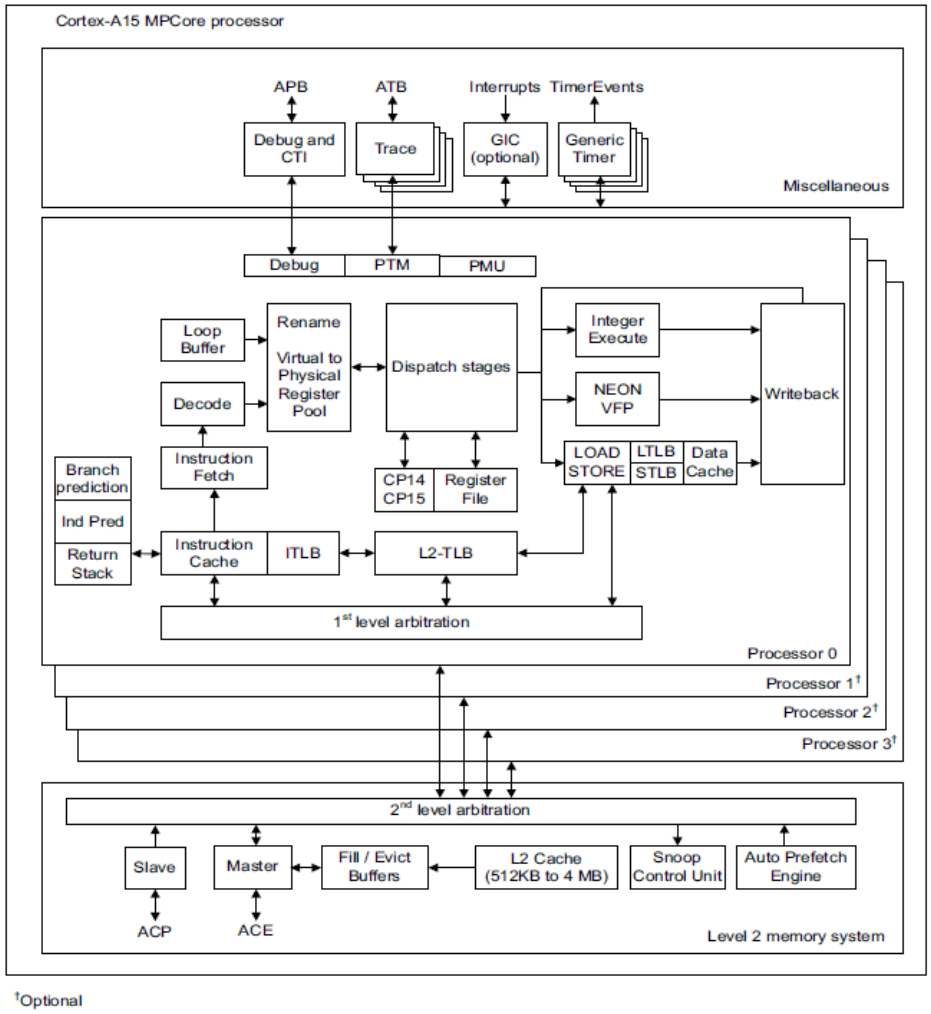
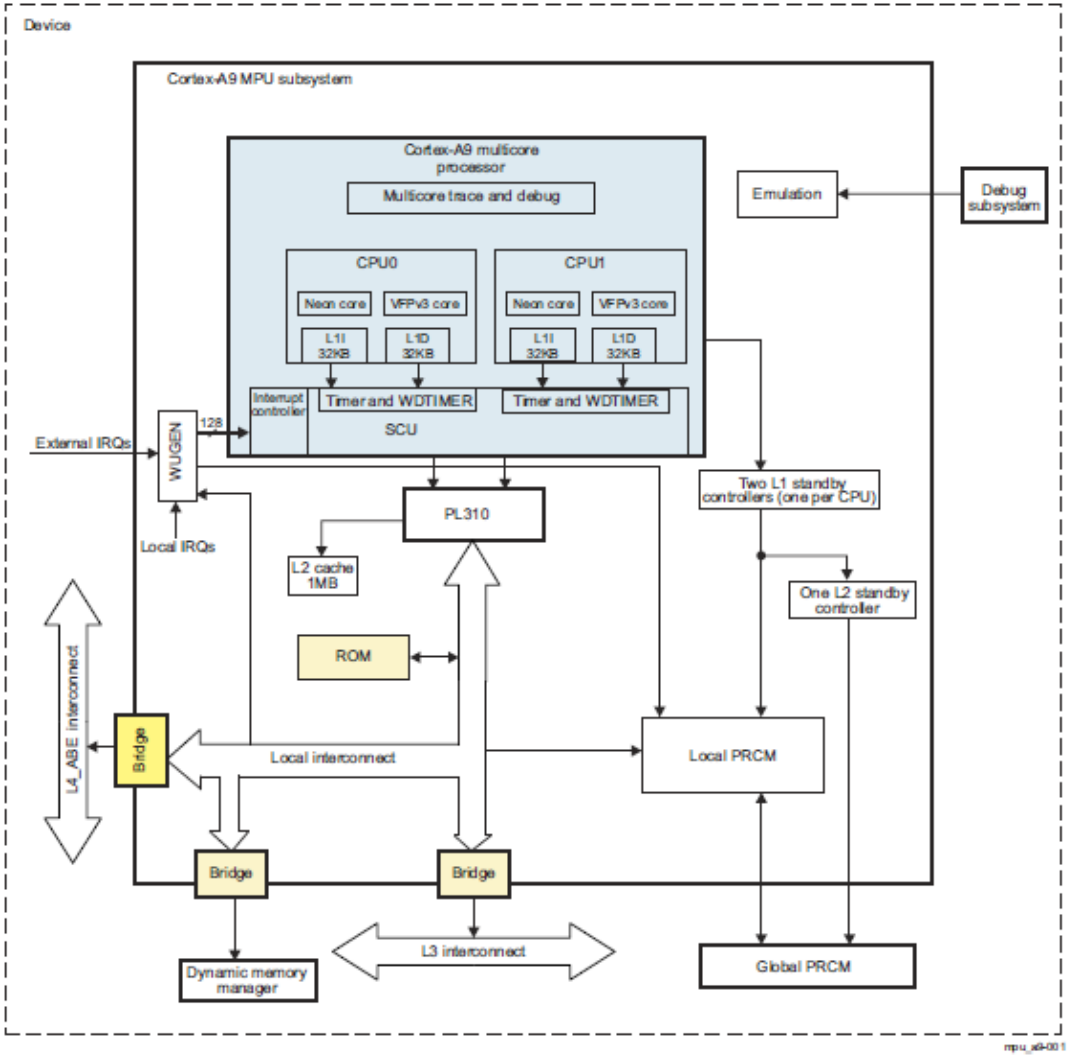


Figure 2-1 Block diagram

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

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

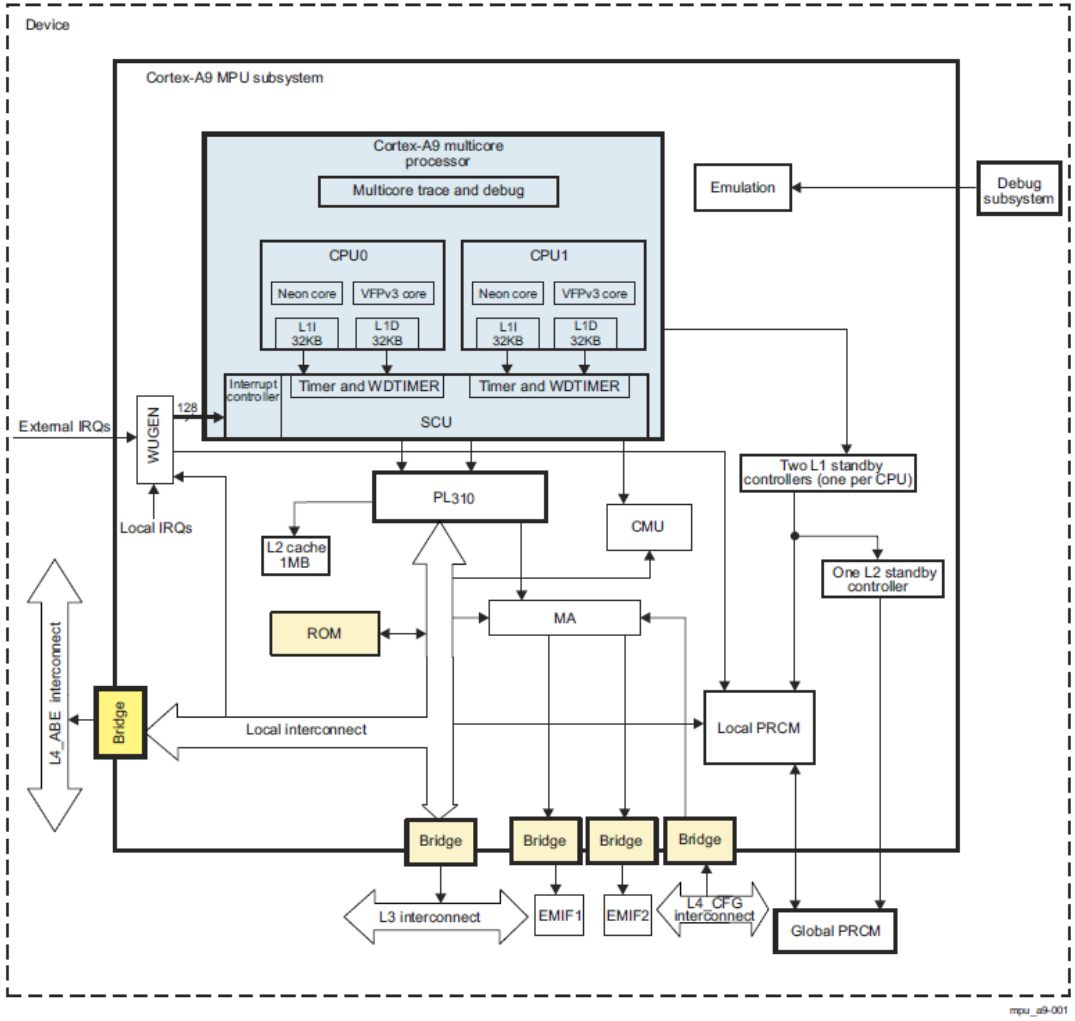
Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4430 Technical Reference Manual, p. 1077.

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

Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4470 Technical Reference Manual, p. 1091.

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

	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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

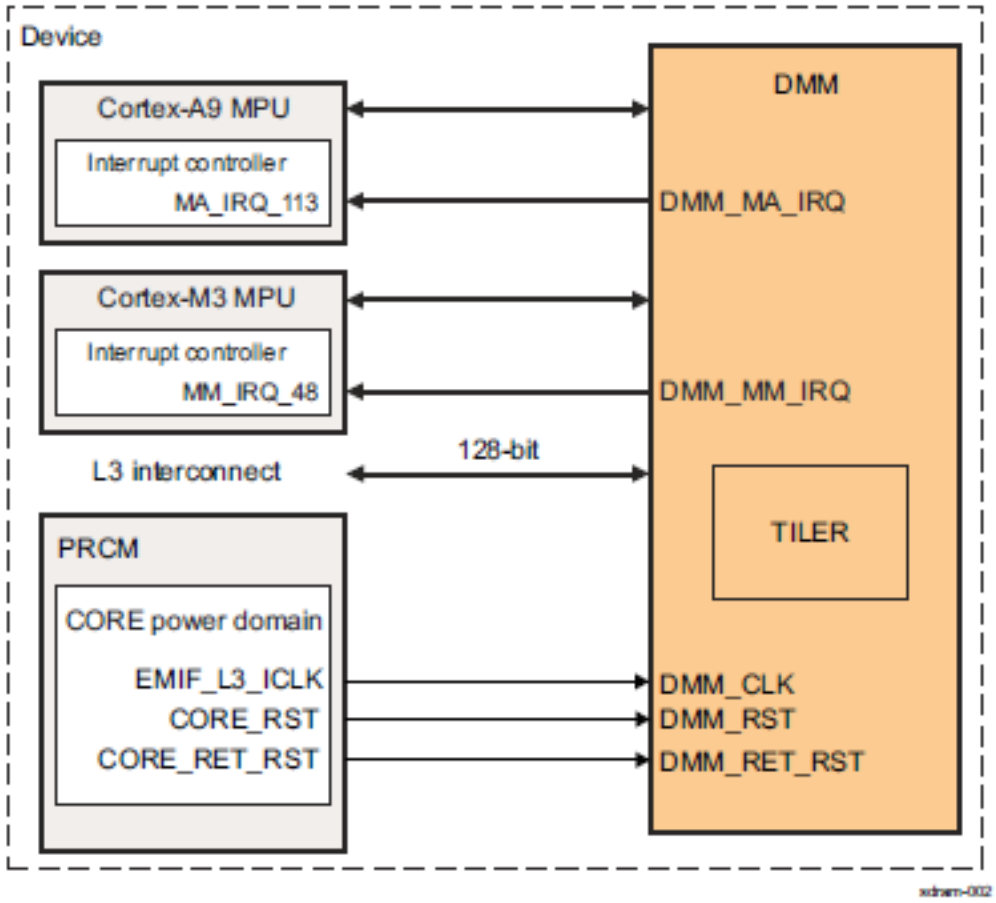
	T == 0 32-bit instructions are fetched (and the PC is incremented by four) and are executed as ARM instructions. **T == 1** 16-bit instructions are fetched (and the PC is incremented by two) and are executed as Thumb instructions. See ARM Architecture Reference Manual at A6-2 . **A3.1.1 Address incrementing 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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

	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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

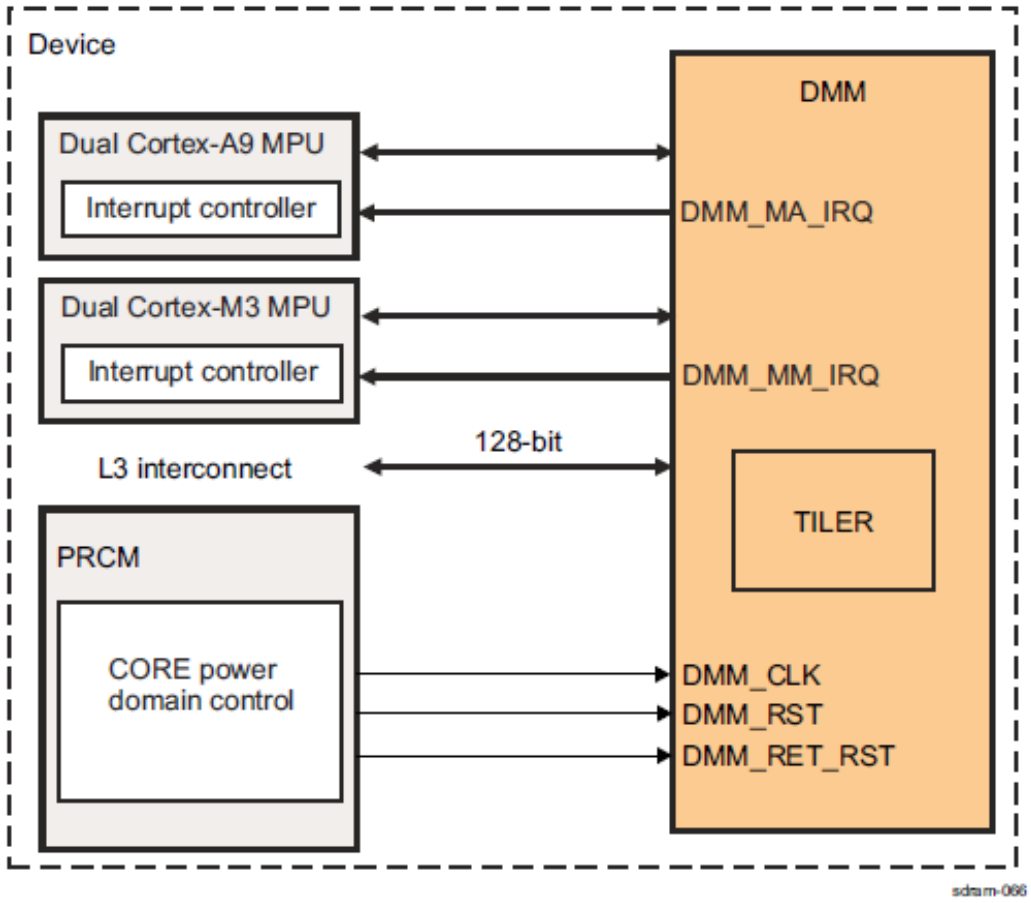
Figure 15-3. DMM Integration



OMAP4430 Technical Reference Manual, p. 3213.

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

Figure 16-2. DMM and Tiler Overview



OMAP4470 Technical Reference Manual, p. 3308.

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

1.3 Block Diagram

Figure 1-1 illustrates the block diagram of Exynos 5250.

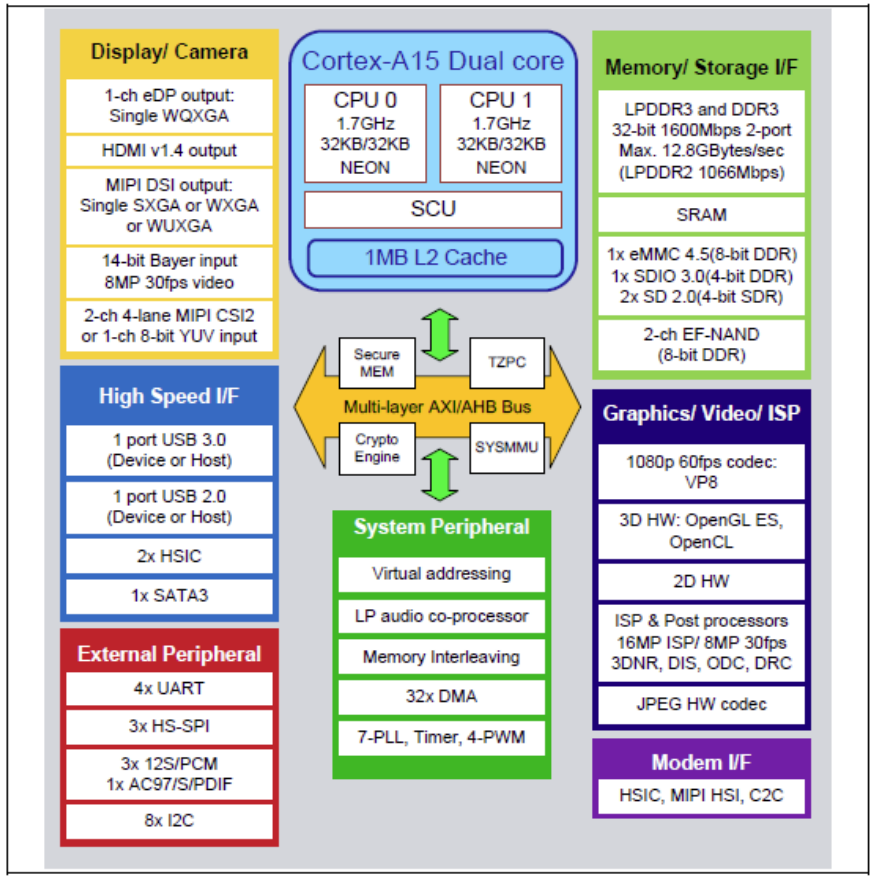


Figure 1-1 Block Diagram of Exynos 5250

Exynos 5 Dual Reference Manual, Fig. 1-1.

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

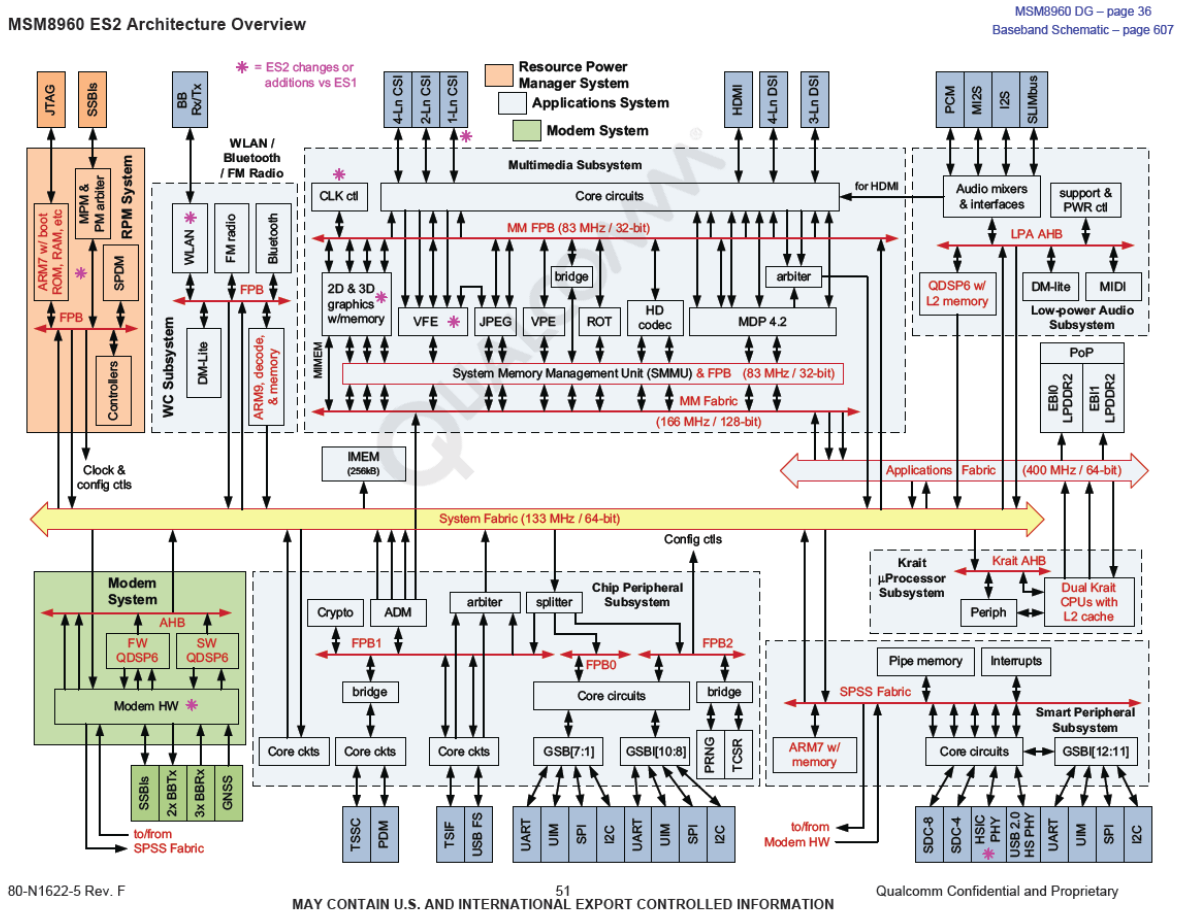
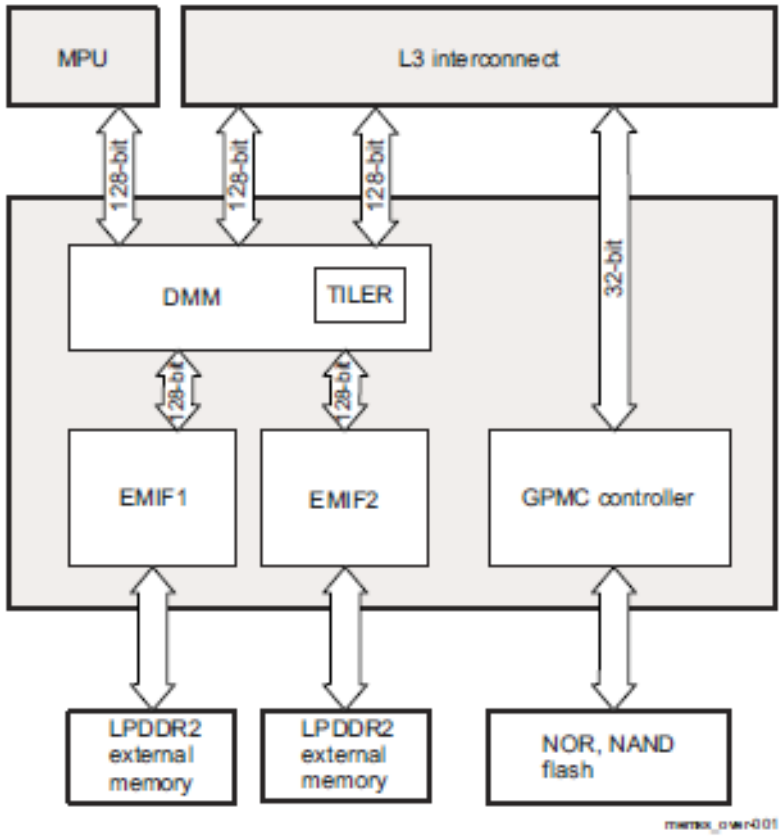
	 MSM8960 Chipset Schematics and Design Guidelines, p. 50.
said memory controller having an address/data bus and a plurality of control lines for connection to a random access memory.	On information and belief the memory controller of each Accused Microprocessor has an address/data bus and a plurality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

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

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

Figure 15-1. Memory Subsystem Functional Diagram

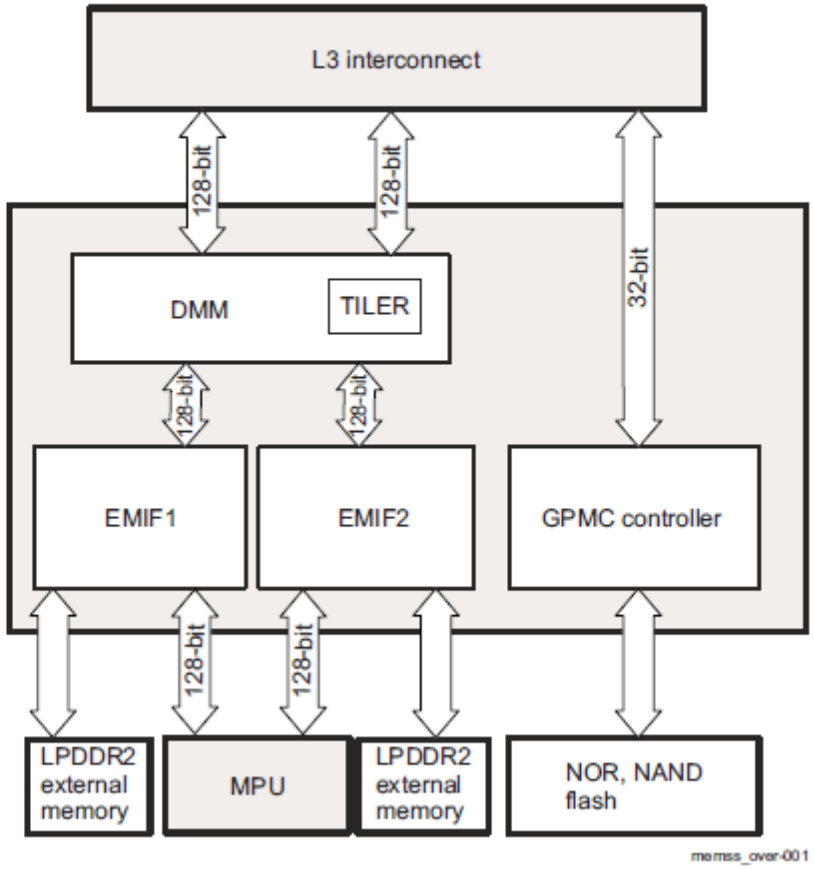


OMAP4430 Technical Reference Manual, p. 3207.

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

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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

1.4.2 Memory Subsystem

Exynos 5250 provides the leading memory bandwidth for mobile applications.

- Mobile DDR3 (LPDDR3) Interface
  - Two ports x32 data bus with 800 MHz per pin, double data rate (DDR)
  - 1.2 V interface voltage
  - Density: Maximum 4 GB by memory map limit, Recommend: Maximum 2 memory dice per port.

NOTE: The max number of column address bits is 10.

- Mobile DDR2 (LPDDR2) interface
  - Two ports x32 data bus with 533 MHz per pin and double data rate (DDR)
  - 1.2 V interface voltage
  - Density: Maximum 4 GB by memory map limit, Recommend: Maximum 2 memory dice per port.

NOTE: The max number of column address bits is 10.

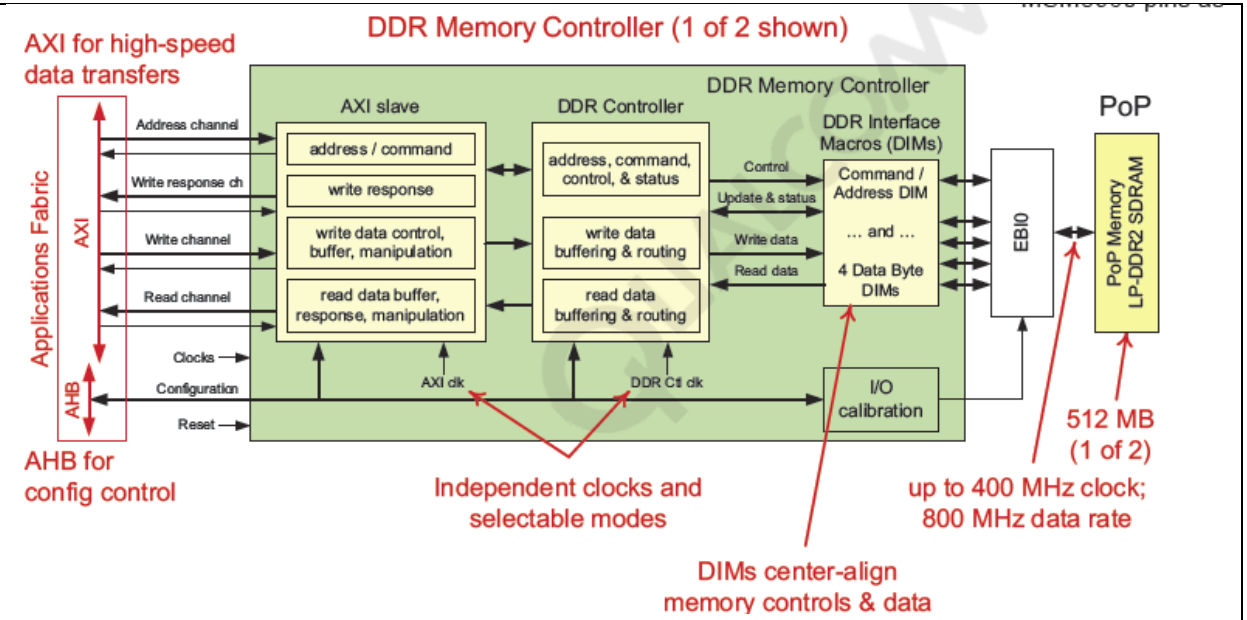
- DDR3/DDR3L Interface
  - Two ports x32 data bus with 800 MHz per pin and double data rate (DDR)
  - 1.5 V/1.35 V interface voltage
  - Density: Maximum 4 GB by memory map limit, Recommend: Maximum 2 memory dice per port.

NOTE: The max number of column address bits is 10.

- eMMC and SD card interface
  - One channel 8-bit eMMC4.5 (1.8 V only)
  - One channel 4-bit SD3.0 (1.8 V only)
  - Two channel eMMC4.3/SD2.0 3.3 V/1.8 V interface voltage
- EF-NAND3.0 interface
  - Two ports x8-bit data bus with up to 200 MHz per pin and double data rate (DDR)
- Embedded internal ROM booting: The system does not need a booting device

Exynos 5 Dual Reference Manual, p. 1-8.

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

	 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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

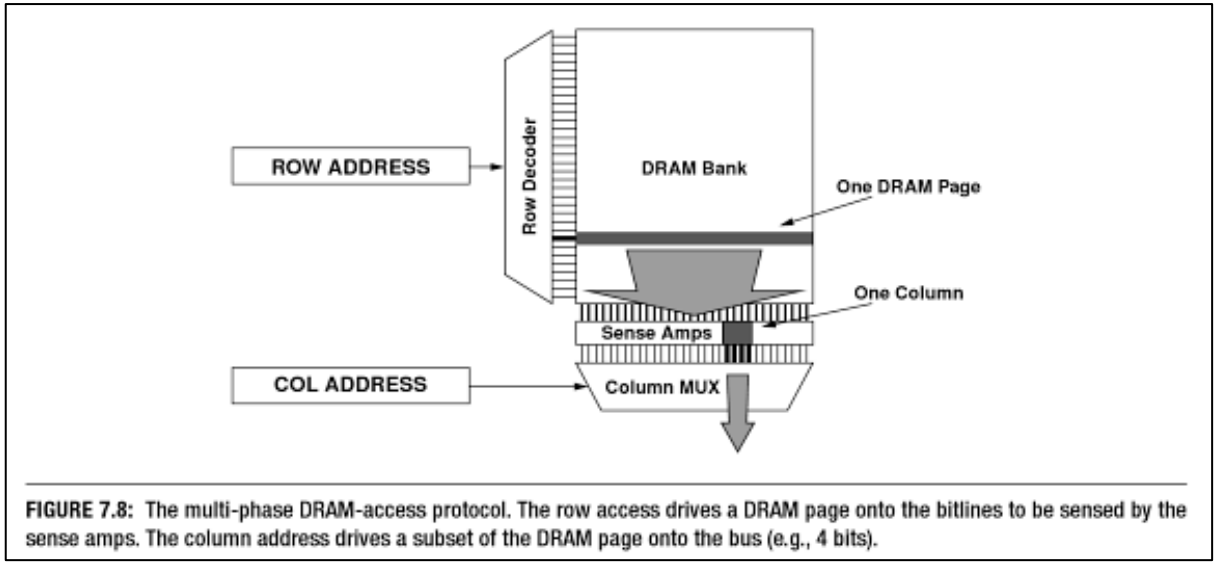


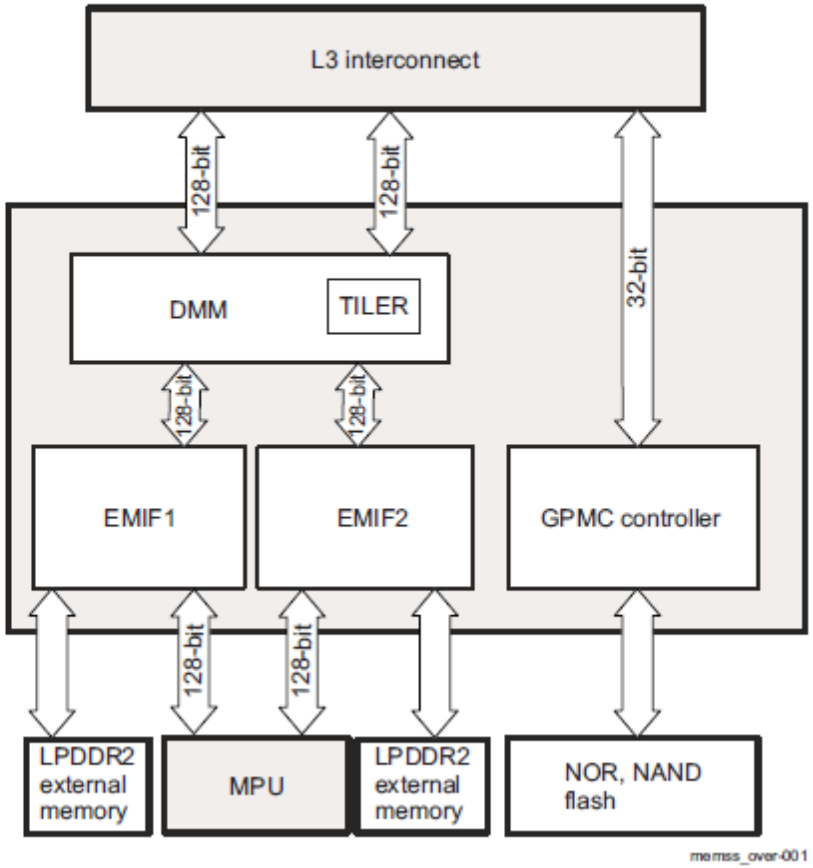
FIGURE 7.8: The multi-phase DRAM-access protocol. The row access drives a DRAM page onto the bitlines to be sensed by the sense amps. The column address drives a subset of the DRAM page onto the bus (e.g., 4 bits).

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

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

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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

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

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, the Accused Microprocessors contain a ring oscillator variable speed system clock connected to the processors on the chips. 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. Generally, 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. *See* BN853-0146739-40.

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

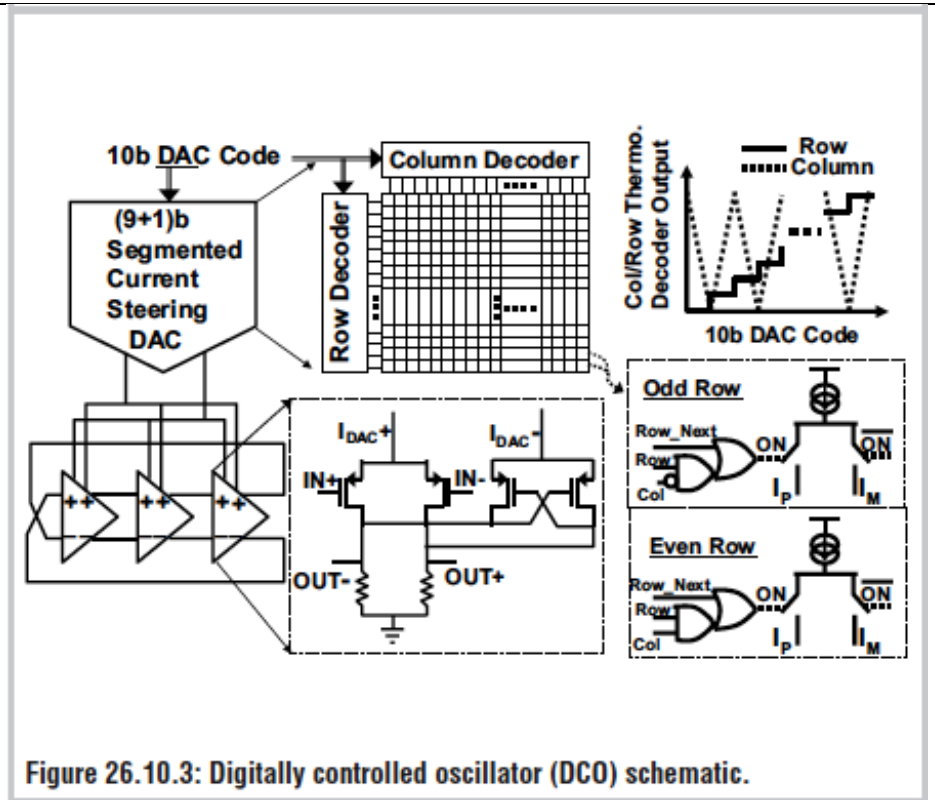
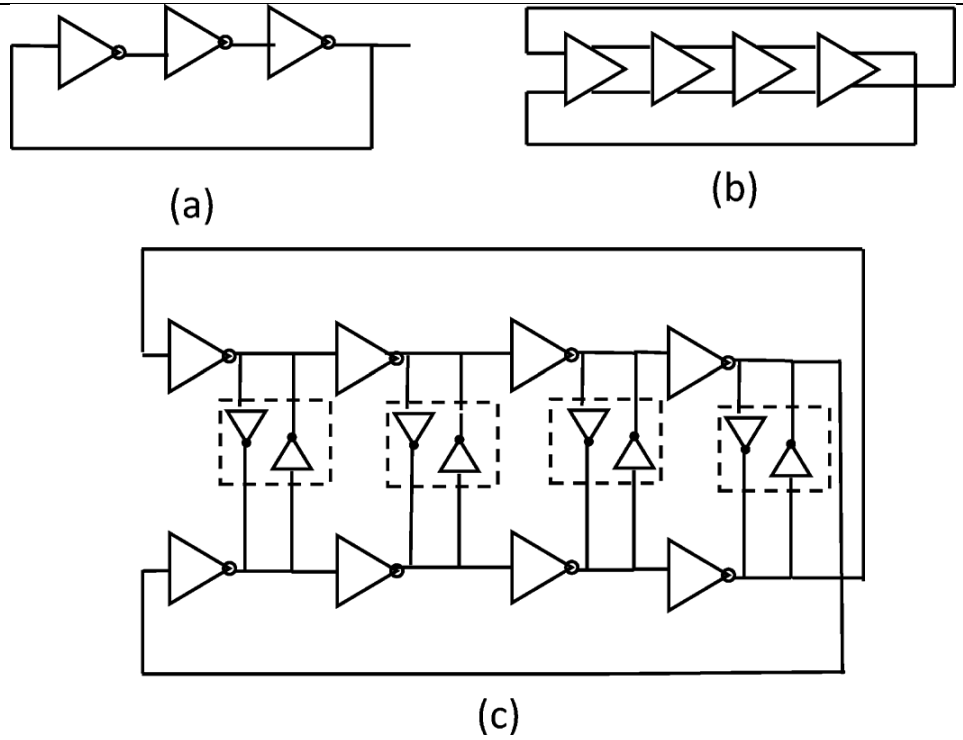


Figure 26.10.3: Digitally controlled oscillator (DCO) schematic.

ISSCC 2004 / Session 26 / *Optical and Fast IO 26.10* [TPLBN051495-TPLBN051504].

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



Architectures and Circuit Techniques for Multi-Purpose Digital Phase Lock Loops [TPLBN051483-TPLBN051494], Fig. 3. "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)." *Id.*

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 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 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 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.

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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

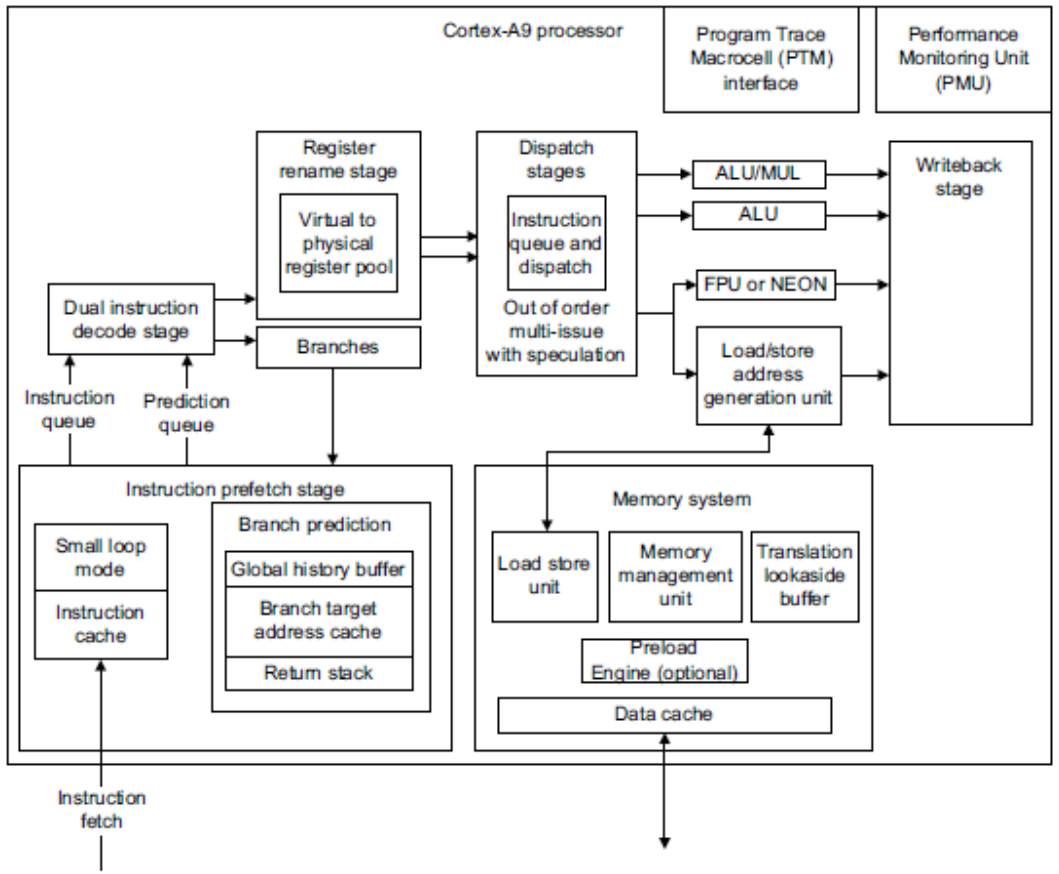


Figure 2-1 Cortex-A9 processor top-level diagram

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

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

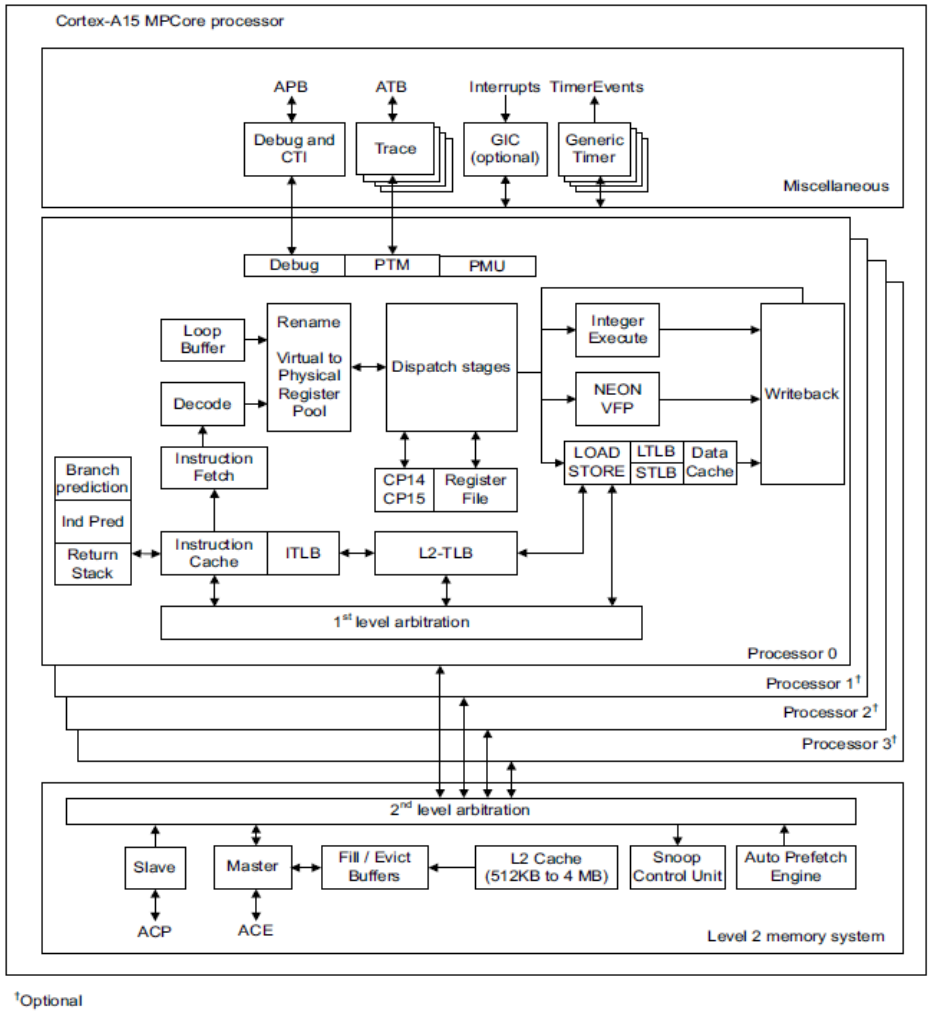
	Figure 2-1 shows a top-level functional diagram of the Cortex-A15 MPCore processor.  **Figure 2-1 Block diagram** Cortex-A15 MPCore Processor Technical Reference Manual p. 2-2.
a second plurality of stack elements	On information and belief, the second plurality of stack elements is configured as random

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

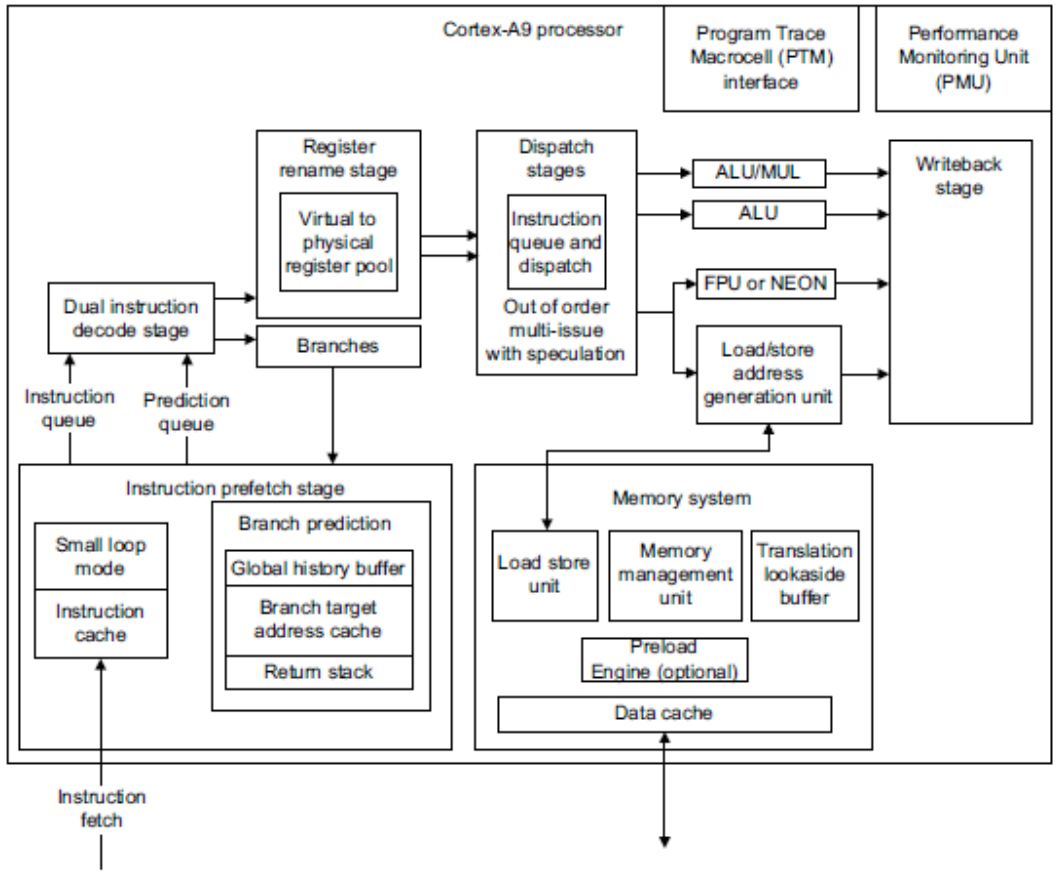
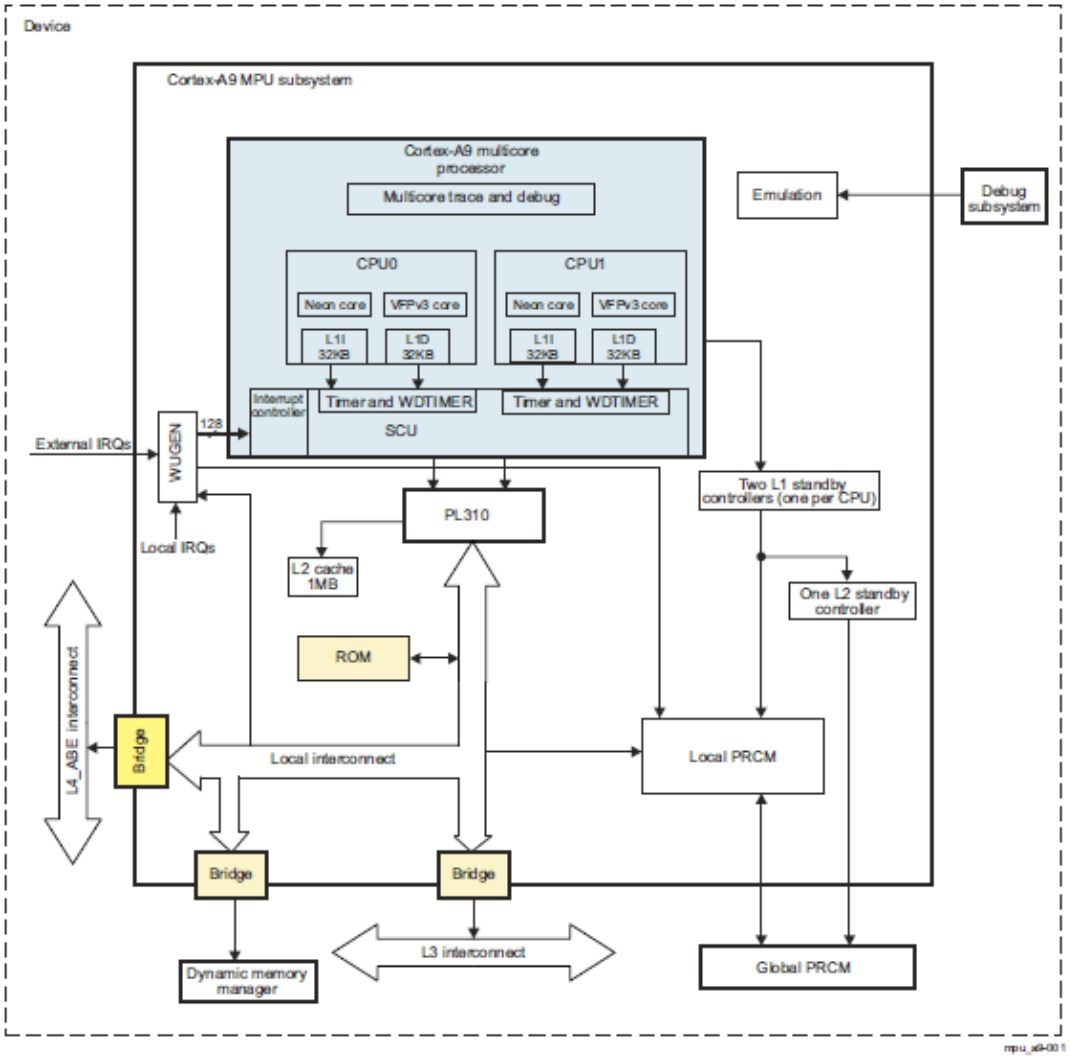
configured as a random access memory,	access memory (L1 cache or L2 cache).  **Figure 2-1 Cortex-A9 processor top-level diagram** Cortex-A9 Technical Reference Manual, p.2-2.

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

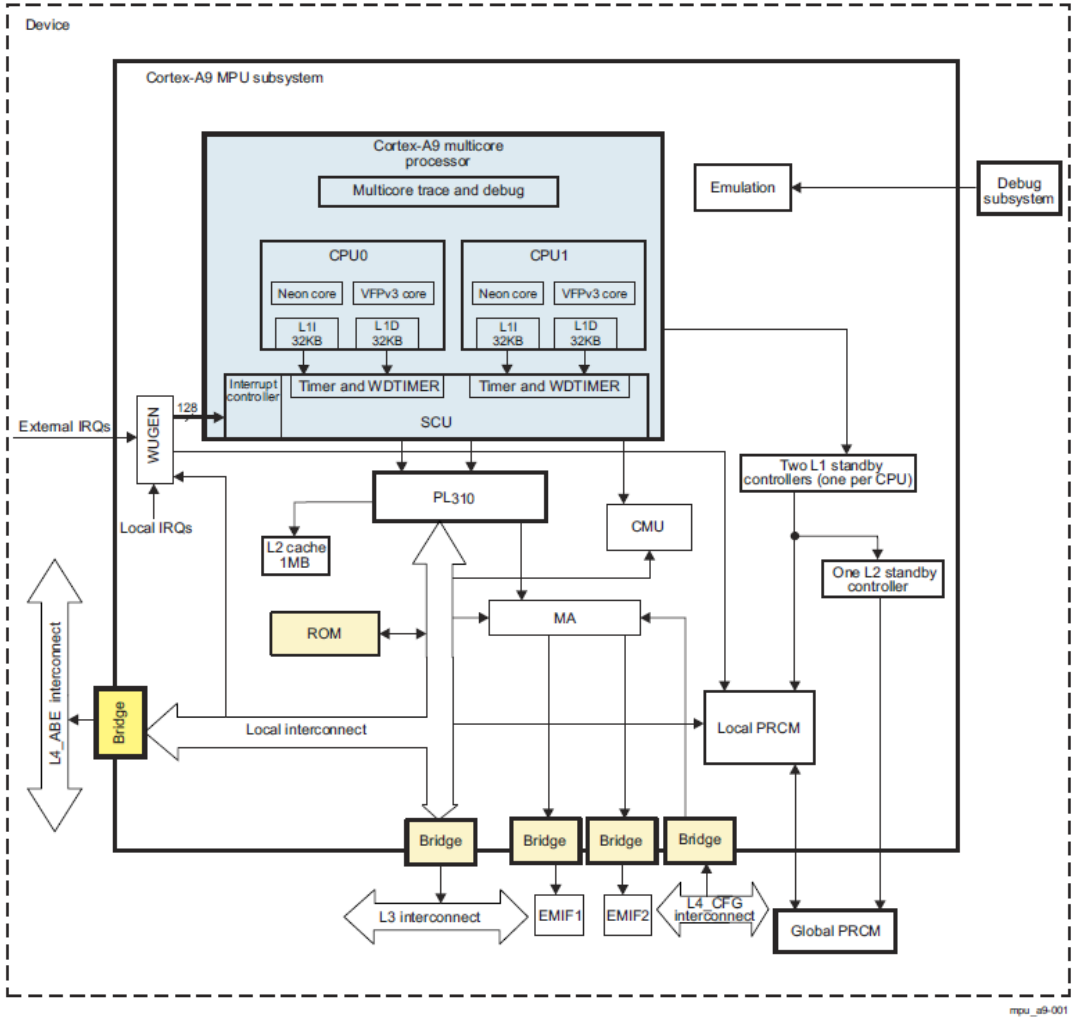
Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4430 Technical Reference Manual, p. 1077.

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

Figure 4-1. Cortex-A9 MPU Subsystem Overview



OMAP4470 Technical Reference Manual, p. 1091.

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

1.3 Block Diagram

Figure 1-1 illustrates the block diagram of Exynos 5250.

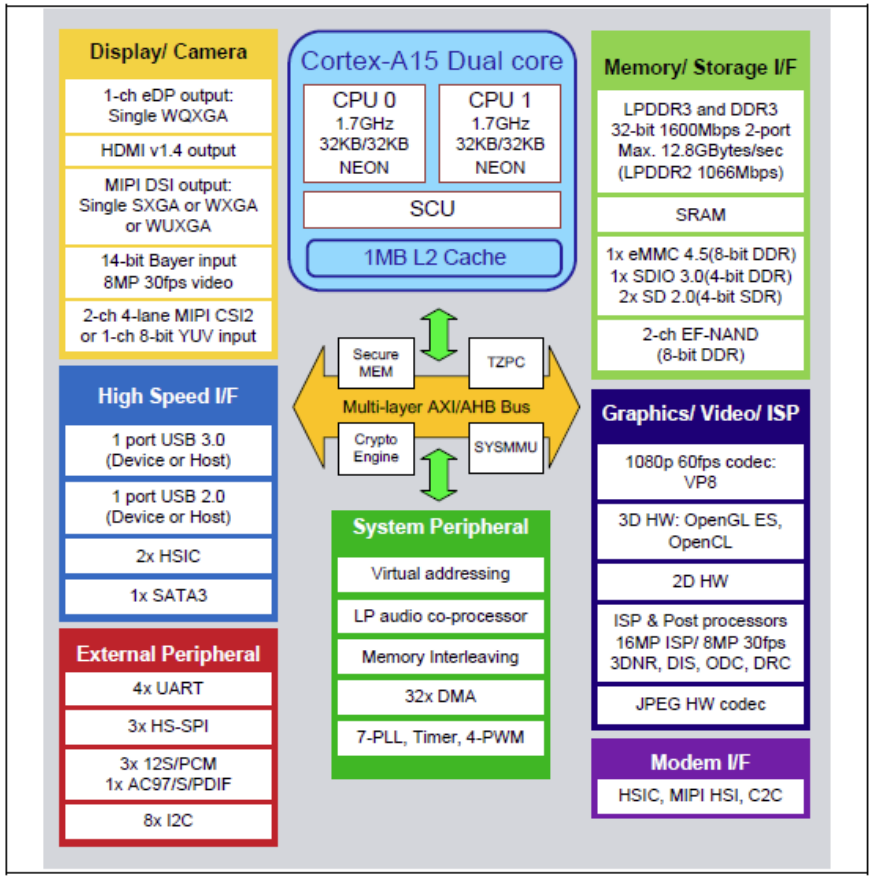


Figure 1-1 Block Diagram of Exynos 5250

Exynos 5 Dual Reference Manual, Fig. 1-1.

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

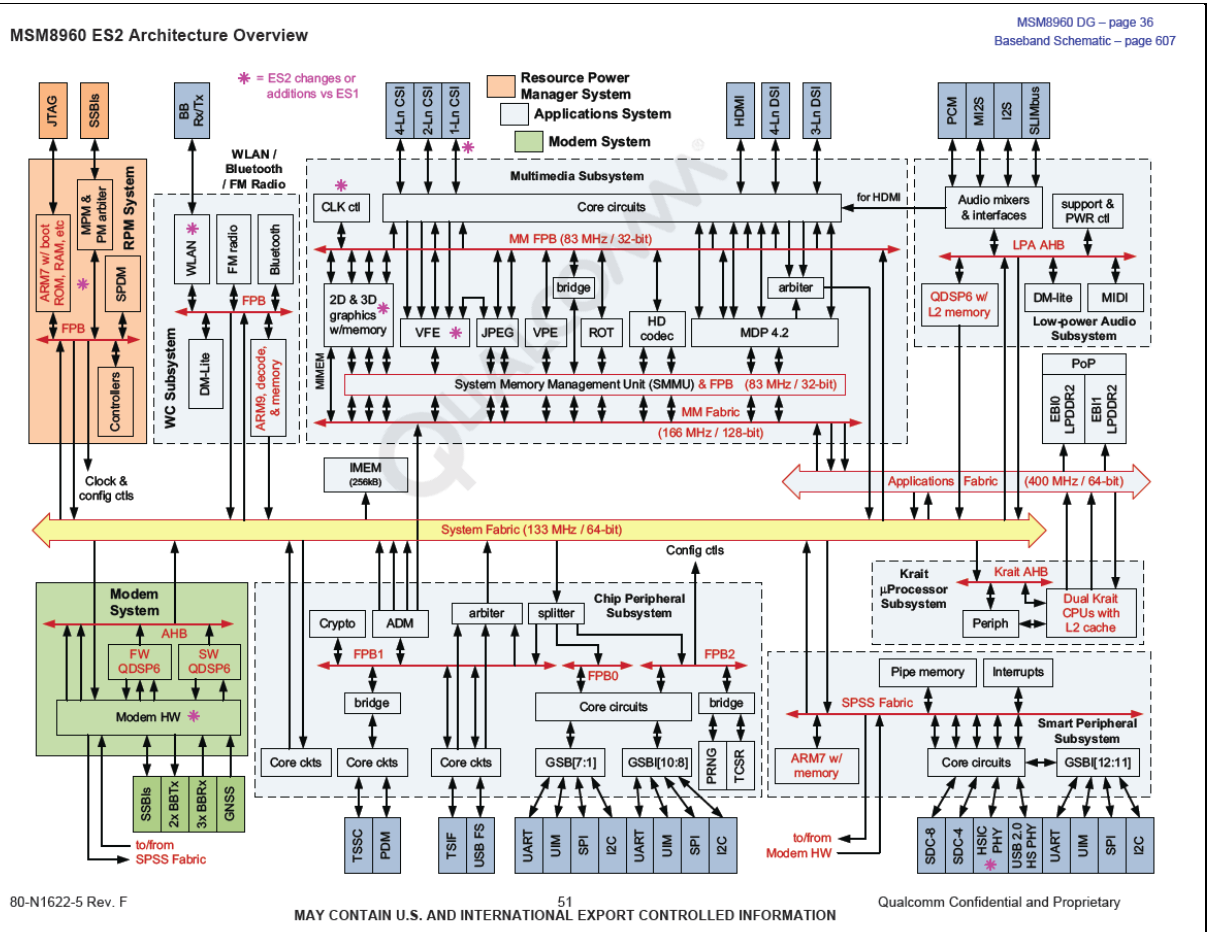
	 MSM8960 Chipset Schematics and Design Guidelines, p. 50.
said first and second plurality of stack elements and said central processing unit being provided in a single integrated circuit,	On information and belief, both the first and the second plurality of stack elements are the central processing unit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

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

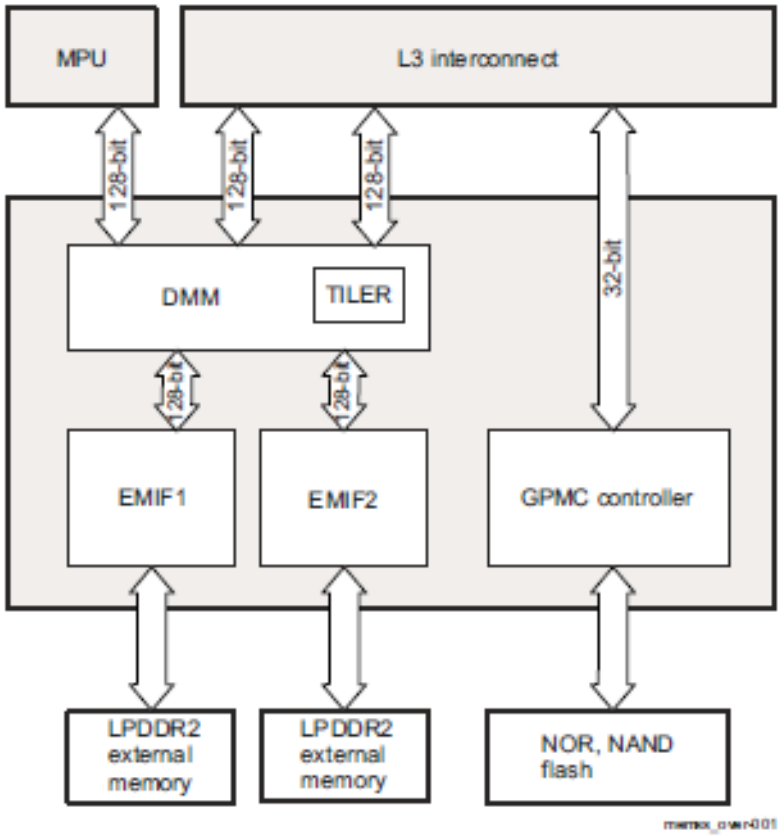
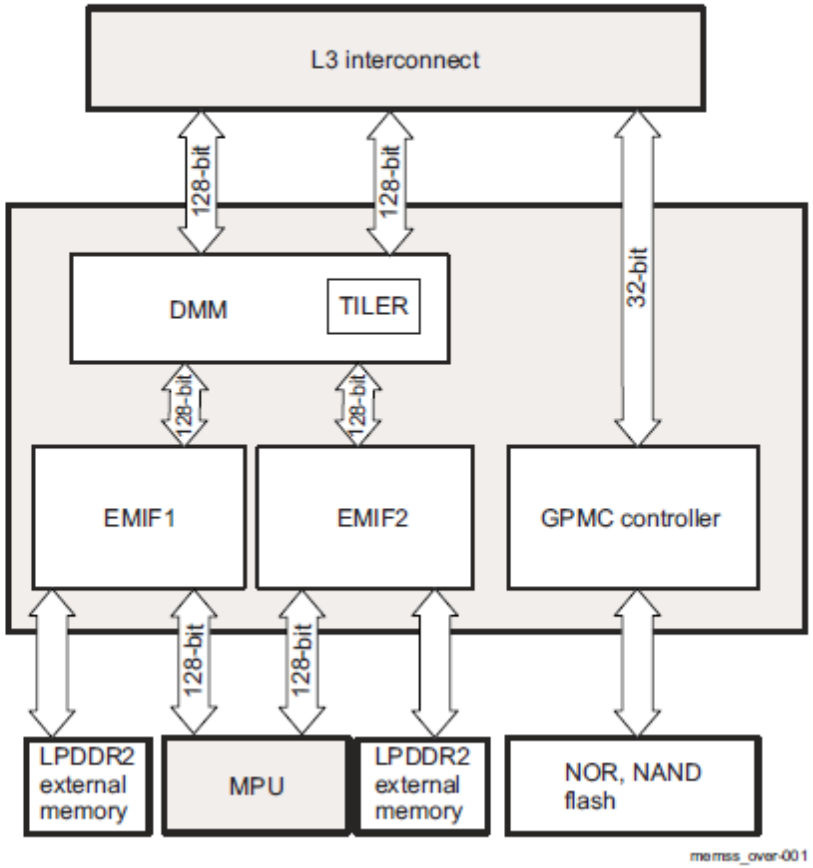
and a third plurality of stack elements configured as a random access memory external to said single integrated circuit.	On information and belief, a third plurality of stack elements is configured as external RAM (*e.g.*, SDRAM). Figure 15-1 shows a functional diagram of all memory subsystems in the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

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

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 E-2 - CLAIM CHART FOR INFRINGEMENT OF U.S. PATENT NO. 5,530,890 By Samsung

1.3 Block Diagram

Figure 1-1 illustrates the block diagram of Exynos 5250.

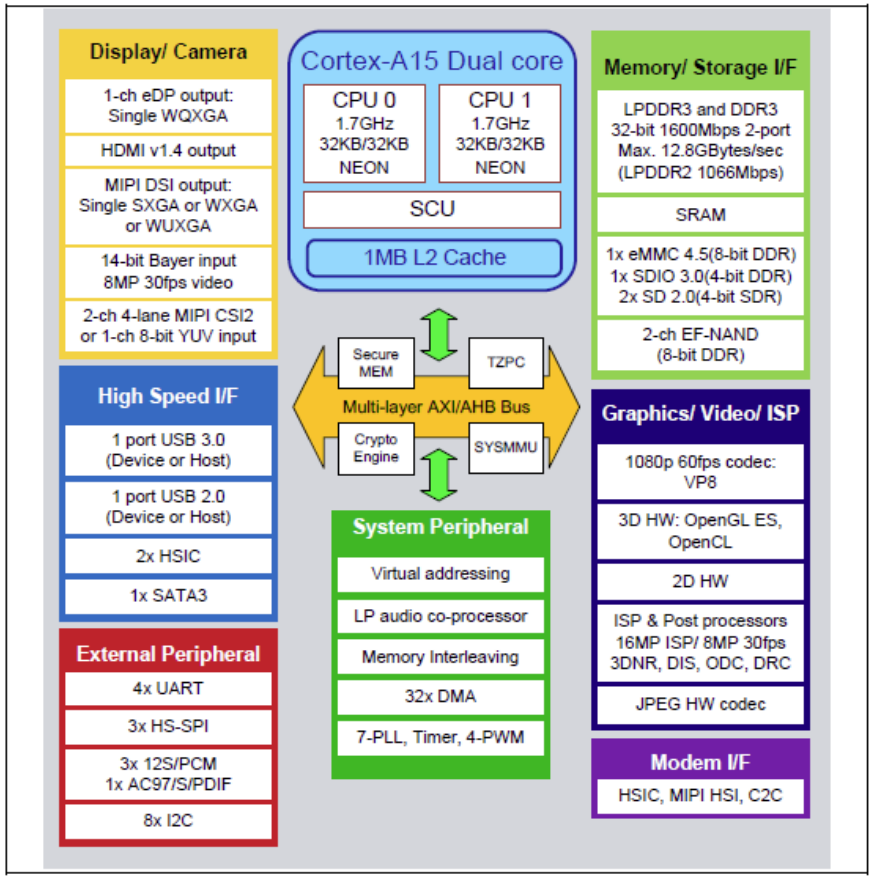
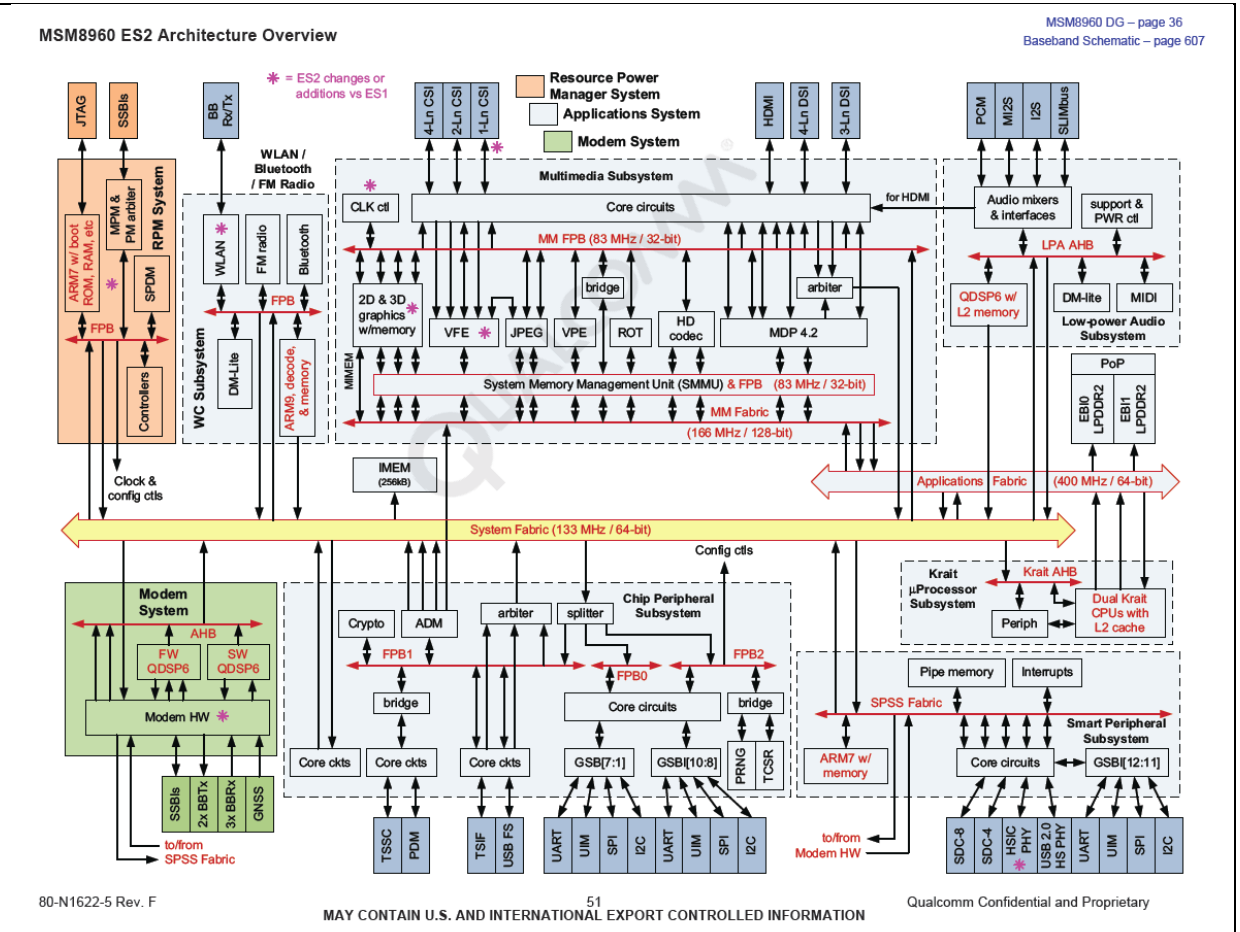


Figure 1-1 Block Diagram of Exynos 5250

Exynos 5 Dual Reference Manual, Fig. 1-1.

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



MSM8960 Chipset Schematics and Design Guidelines, p. 50.