UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF WASHINGTON

ALTAIR LOGIX LLC, Plaintiff, v. **TORADEX, INC.,** Defendant.	Case No. 19-cv-449 COMPLAINT FOR PATENT INFRINGEMENT DEMAND FOR JURY TRIAL

Plaintiff Altair Logix LLC files this Original Complaint for Patent Infringement against Toradex, Inc., and would respectfully show the Court as follows:

I. NATURE OF THE LAWSUIT

1. This is an action for patent infringement under the Patent Laws of the United States, Title 35 United States Code ("U.S.C.") resulting from Toradex, Inc. infringing, in an illegal and unauthorized manner and without authorization and/or consent from Altair Logix LLC, United States Patent No. 6,289,434 pursuant to 35 U.S.C. §271, and to recover damages, attorney's fees, and costs.

II. THE PARTIES

2. Plaintiff Altair Logix LLC ("Altair Logix" or "Plaintiff") is a Texas limited liability company with its principal place of business at 15922 Eldorado Pkwy, Suite 500 #1513, Frisco, TX 75035.

3. On information and belief, Defendant Toradex, Inc. ("Defendant") is a corporation organized and existing under the laws of Washington with a place of

business at 219 1st Ave S, Suite 410, Seattle, WA 98104. Defendant has a registered agent at Incorp Services, Inc., 4505 Pacific Hwy E Ste C2, Fife, WA, 98424-2638

III.JURISDICTION AND VENUE

4. This action arises under the patent laws of the United States, Title 35 of the United States Code. This Court has subject matter jurisdiction of such action under 28 U.S.C. §§ 1331 and 1338(a).

5. On information and belief, Defendant is subject to this Court's specific and general personal jurisdiction, pursuant to due process and the Washington Long-Arm Statute, due at least to its business in this forum, including at least a portion of the infringements alleged herein. Furthermore, Defendant is subject to this Court's specific and general personal jurisdiction because Defendant is a Washington corporation.

6. Without limitation, on information and belief, within this State and this District, Defendant has used, sold, and/or offered for sale the patented inventions thereby committing, and continuing to commit, acts of patent infringement alleged herein. In addition, on information and belief, Defendant has derived revenues from its infringing acts occurring within Washington and the Western District of Washington. Further, on information and belief, Defendant is subject to the Court's general jurisdiction, including from regularly doing or soliciting business, engaging in other persistent courses of conduct, and deriving substantial revenue from goods and services provided to persons or entities in Washington and the Western District of Washington. Further, on information and belief, Defendant is subject to the Court's personal jurisdiction at least due to its sale of products and/or services within Washington and the Western District of Washington. Defendant has committed such purposeful acts and/or transactions in Washington and the Western District of Washington such that it

reasonably should know and expect that it could be haled into this Court as a consequence of such activity.

7. Venue is proper in this district under 28 U.S.C. § 1400(b). On information and belief, Defendant is incorporated in Washington and the Western District of Washington. Under the patent laws, because Defendant is incorporated in Washington and the Western District of Washington, Washington and the Western District of Washington is the only district in which it resides. On information and belief, from and within this District Defendant has committed at least a portion of the infringements at issue in this case and has a business location within this District.

8. For these reasons, personal jurisdiction exists and venue is proper in this Court under 28 U.S.C. § 1400(b).

IV.COUNT I
(PATENT INFRINGEMENT OF UNITED STATES PATENT NO. 6,289,434)

9. Plaintiff incorporates the above paragraphs herein by reference.

10. On September 11, 2001, United States Patent No. 6,289,434 ("the '434 Patent") was duly and legally issued by the United States Patent and Trademark Office. The application leading to the '434 patent was filed on February 27, 1998. (Ex. A at cover).

11. The '434 Patent is titled "Apparatus and Method of Implementing Systems on Silicon Using Dynamic-Adaptive Run-Time Reconfigurable Circuits for Processing Multiple, Independent Data and Control Streams of Varying Rates." A true and correct copy of the '434 Patent is attached hereto as Exhibit A and incorporated herein by reference.

12. Plaintiff is the assignee of all right, title and interest in the ‘434 patent, including all rights to enforce and prosecute actions for infringement and to collect damages for all relevant times against infringers of the ‘434 Patent. Accordingly, Plaintiff possesses the exclusive right and standing to prosecute the present action for infringement of the ‘434 Patent by Defendant.

13. The invention in the ‘434 Patent relates to the field of runtime reconfigurable dynamic-adaptive digital circuits which can implement a myriad of digital processing functions related to systems control, digital signal processing, communications, image processing, speech and voice recognition or synthesis, three-dimensional graphics rendering, and video processing. (Ex. A at col. 1:32-38). The object of the invention is to provide a new method and apparatus for implementing systems on silicon or other chip material which will enable the user a means for achieving the performance of fixed-function implementations at a lower cost. (*Id.* at col. 2:64 – col. 3:1).

14. The most common method of implementing various functions on an integrated circuit is by specifically designing the function or functions to be performed by placing on silicon an interconnected group of digital circuits in a non-modifiable manner (hard-wired or fixed function implementation). (*Id.* at col. 1:42-47). These circuits are designed to provide the fastest possible operation of the circuit in the least amount of silicon area. (*Id.* at col. 1:47-49). In general, these circuits are made up of an interconnection of various amounts of random-access memory and logic circuits. (*Id.* at col. 1:49-51). Complex systems on silicon are broken up into separate blocks and each block is designed separately to only perform the function that it was intended to do. (*Id.* at col. 1:51-54). Each block has to be individually tested and validated, and then the

whole system has to be tested to make sure that the constituent parts work together. (*Id.* at col. 1:54-56). This process is becoming increasingly complex as we move into future generations of single-chip system implementations. (*Id.* at col. 1:57-59). Systems implemented in this way generally tend to be the highest performing systems since each block in the system has been individually tuned to provide the expected level of performance. (*Id.* at col. 1:59-62). This method of implementation may be the smallest (cheapest in terms of silicon area) method when compared to three other distinct ways of implementing such systems. (*Id.* at col. 1:62-65). Each of the other three have their problems and generally do not tend to be the most cost-effective solution. (*Id.* at col. 1:65-67).

15. The first way is implemented in software using a microprocessor and associated computing system, which can be used to functionally implement any system. (*Id.* at col. 2:1-2). However, such systems would not be able to deliver real-time performance in a cost-effective manner for the class of applications that was described above. (*Id.* at col. 2:3-5). Their use is best for modeling the subsequent hard-wired/fixed-function system before considerable design effort is put into the system design. (*Id.* at col. 2:5-8).

16. The second way of implementing such systems is by using an ordinary digital signal processor (DSP). (*Id.* at col. 2:9-10). This class of computing machines is useful for real-time processing of certain speech, audio, video and image processing problems and in certain control functions. (*Id.* at col. 2:10-13). However, they are not cost-effective when it comes to performing certain real time tasks which do not have a high degree of parallelism in them or tasks that require multiple parallel threads of operation such as three-dimensional graphics. (*Id.* at col. 2:13-17).

17. The third way of implementing such systems is by using field programmable gate arrays (FPGA). (*Id.* at col. 2:18-19). These devices are made up of a two-dimensional array of fine grained logic and storage elements which can be connected together in the field by downloading a configuration stream which essentially routes signals between these elements. (*Id.* at col. 2:19-23). This routing of the data is performed by pass-transistor logic. (*Id.* at col. 2:24-25). FPGAs are by far the most flexible of the three methods mentioned. (*Id.* at col. 2:25-26). The problem with trying to implement complex real-time systems with FPGAs is that although there is a greater flexibility for optimizing the silicon usage in such devices, the designer has to trade it off for increase in cost and decrease in performance. (*Id.* at col. 2:26-30). The performance may (in some cases) be increased considerably at a significant cost, but still would not match the performance of hard-wired fixed function devices. (*Id.* at col. 2:30-33).

18. These three ways do not reduce the cost or increase the performance over fixed-function systems. (*Id.* at col. 2:35-37). In terms of performance, fixed-function systems still outperform the three ways for the same cost. (*Id.* at col. 2:37-39).

19. The three systems can theoretically reduce cost by removing redundancy from the system. (*Id.* at col. 2:40-41). Redundancy is removed by re-using computational blocks and memory. (*Id.* at col. 2:41-42). The only problem is that these systems themselves are increasingly complex, and therefore, their computational density when compared with fixed-function devices is very high. (*Id.* at col. 2:42-45).

20. Most systems on silicon are built up of complex blocks of functions that have varying data bandwidth and computational requirements. (*Id.* at col. 2:46-48). As data and control information moves through the system, the processing bandwidth varies

enormously. (*Id.* at col. 2:48-50). Regardless of the fact that the bandwidth varies, fixed-function systems have logic blocks that exhibit a “temporal redundancy” that can be exploited to drastically reduce the cost of the system. (*Id.* at col. 2:50-53). This is true, because in fixed function implementations all possible functional requirements of the necessary data processing must be implemented on the silicon regardless of the final application of the device or the nature of the data to be processed. (*Id.* at col. 2:53-57). Therefore, if a fixed function device must adaptively process data, then it must commit silicon resources to process all possible flavors of the data. (*Id.* at col. 2:58-60). Furthermore, state-variable storage in all fixed function systems are implemented using area inefficient storage elements such as latches and flip-flops. (*Id.* at col. 2:60-63).

21. The inventors therefore sought to provide a new apparatus for implementing systems on a chip that will enable the user to achieve performance of fixed-function implementation at a lower cost. (*Id.* at col. 2:64 – col. 3:1). The lower cost is achieved by removing redundancy from the system. (*Id.* at col. 3:1-2). The redundancy is removed by re-using groups of computational and storage elements in different configurations. (*Id.* at col. 3:2-4). The cost is further reduced by employing only static or dynamic ram as a means for holding the state of the system. (*Id.* at col. 3:4-6). This invention provides a way for effectively adapting the configuration of the circuit to varying input data and processing requirements. (*Id.* at col. 3:6-8). All of this reconfiguration can take place dynamically in run-time without any degradation of performance over fixed-function implementations. (*Id.* at col. 3:8-11).

22. The present invention is therefore an apparatus for adaptively dynamically reconfiguring groups of computations and storage elements in run-time to process multiple separate streams of data and control at varying rates. (*Id.* at col. 3:14-

18). The ‘434 patent refers to the aggregate of the dynamically reconfigurable computational and storage elements as a “media processing unit.”

23. The claimed apparatus has addressable memory for storing data and a plurality of instructions that can be provided through a plurality of inputs/outputs that is couple to the input/output of a plurality of media processing units. (*Id.* at col. 55:21-30). The media processing unit comprises a multiplier, an arithmetic unit, and arithmetic logic unit and a bit manipulation unit. (*Id.* at col. 55:31 – col. 56:20). The ‘434 patent provides examples to explain each of the parts of the media processing unit. (*Id.* at col. 16:27-61 (multiplier and adder); *id.* at col. 16:62 – col. 17:1-9 (arithmetic logic unit); and *id.* at col. 17:10 – col. 17:43 (bit manipulation unit)). Each of the parts has a data input coupled to the media processing unit input/output, an instruction input coupled to the mediate processing unit input/output, and a data output coupled to the mediate processing unit input/output. (*Id.* at col. 55:31 – col. 56:20). Furthermore, the arithmetic logic unit must be capable of operating concurrently with either the multiplier and arithmetic unit. (*Id.* at col. 56:6-12). And the bit manipulation unit must be capable of operating concurrently with the arithmetic logic unit and at least either the multiplier or the arithmetic unit. (*Id.* at col. 56:13-20). Each of the plurality of media processing units must be capable of performing an operating simultaneously with the performance of other operations by other media processing units. (*Id.* at col. 56:21-24). An operation comprises the media processing unit receiving an instruction and data from memory, processing the data responsive to the instruction to produce a result, and providing the result to the media processor input/output. (*Id.* at col. 56:26-33).

24. An exemplary block diagram of the claimed systems is shown in Figure 3 of the ‘434 patent:

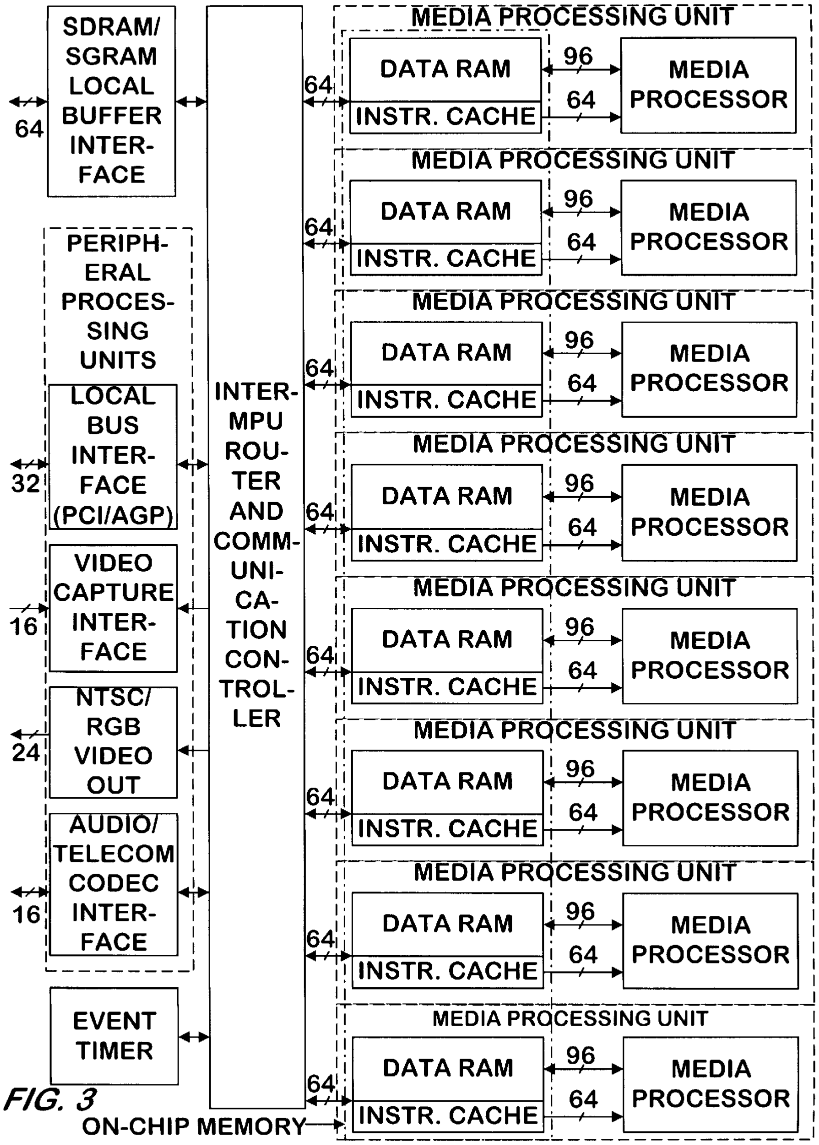

SDRAM/ SGRAM LOCAL BUFFER INTERFACE
64
PERIPHERAL PROCESSING UNITS
LOCAL BUS INTERFACE (PCI/AGP)
32
VIDEO CAPTURE INTERFACE
16
NTSC/ RGB VIDEO OUT
24
AUDIO/ TELECOM CODEC INTERFACE
16
EVENT TIMER
INTER-MPU ROUTER AND COMMUNICATION CONTROLLER
MEDIA PROCESSING UNIT
DATA RAM
INSTR. CACHE
MEDIA PROCESSOR
96
64
ON-CHIP MEMORY

FIG. 3

(*Id.* at Fig. 3). Exemplary architecture and coding for the apparatus is disclosed in the '599 patent. (*E.g.*, *id.* at col. 16:15 – col. 52:20; Figs. 9 – 106).

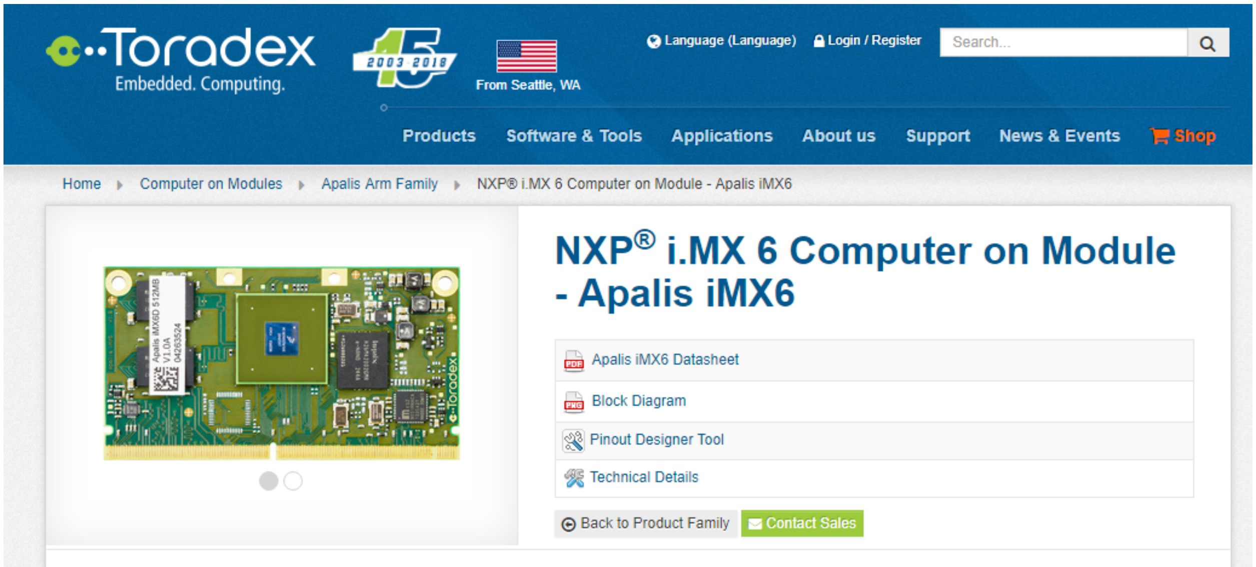
25. As further demonstrated by the prosecution history of the '434 patent, the claimed invention in the '434 patent was unconventional. Claim 1 of the '434 patent was an originally filed claim that issued without any amendment. There was no rejection in the prosecution history contending that claim 1 was anticipated by any prior art.

26. A key element behind the invention is one of reconfigurability and reusability. (*Id.* at col. 13:26-27). Each apparatus is therefore made up of very high-speed core elements that on a pipelined basis can be configured to form a more complex function. (*Id.* at col. 13:27-30). This leads to a lower gate count, thereby giving a smaller die size and ultimately a lower cost. (*Id.* at col. 13:30-31). Since the apparatuses are virtually identical to each other, writing software becomes very easy. (*Id.* at col. 13:32-33). The RISC-like nature of each of the media processing units also allows for a consistent hardware platform for simple operating system and driver development. (*Id.* at col. 13:33-36). Any one of the media processing units can take on a supervisory role and act as a central controller if necessary. (*Id.* at col. 13:36-37). This can be very useful in set top applications where a controlling CPU may not be necessary, further reducing system cost. (*Id.* at col. 13:37-40). The claimed apparatus is therefore an unconventional way of implementing processors that can achieve the performance of fixed-function implementations at a lower cost. (*Id.* at col. 2:64 – col. 3:11).

27. **Direct Infringement.** Upon information and belief, Defendant has been directly infringing claim of the '434 patent in Washington and the Western District of Washington, and elsewhere in the United States, by making, using, selling, and/or offering for sale an apparatus for processing data for media processing that satisfies each and every limitation of claim 1, including without limitation the Apalis iMX6 ("Accused Instrumentality"). (*E.g.*, https://www.toradex.com/computer-on-modules/apalis-arm-family/nxp-freescale-imx-6; https://docs.toradex.com/101770-apalis-imx6-datasheet.pdf; https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-

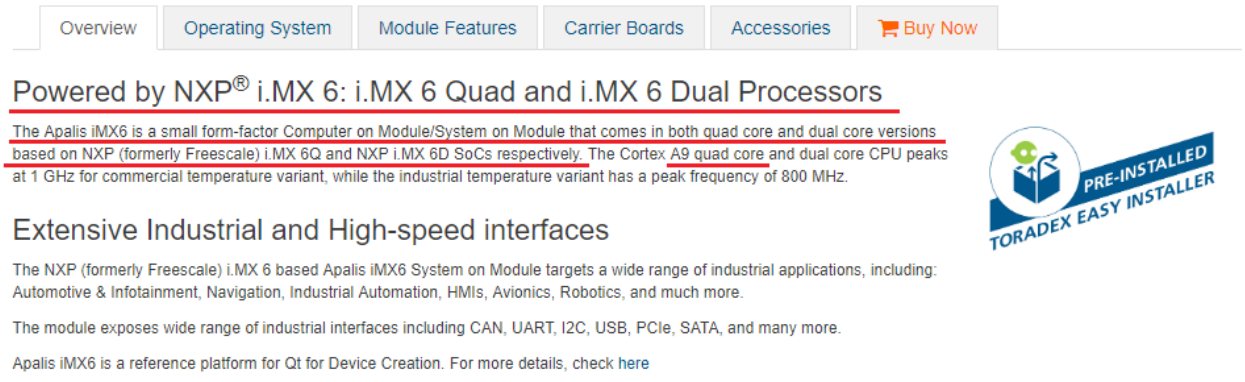
and-mcus/i.mx-applications-processors/i.mx-6-processors/i.mx-6quad-processors-high-performance-3d-graphics-hd-video-arm-cortex-a9-core:i.MX6Q).

28. The Accused Instrumentality comprises an addressable memory (*e.g.*, memory system of the accused product) for storing the data, and a plurality of instructions, and having a plurality of input/outputs, each said input/output for providing and receiving at least one selected from the data and the instructions. As shown below, the accused product comprises a memory system which is coupled to multicore ARM processors through multiple internal inputs/outputs. The memory system provides instructions and stored data for processing and receives processed data.



NXP® i.MX 6 Computer on Module - Apalis iMX6

(*E.g.*, https://www.toradex.com/computer-on-modules/apalis-arm-family/nxp-freescale-imx-6).



Powered by NXP® i.MX 6: i.MX 6 Quad and i.MX 6 Dual Processors

The Apalis iMX6 is a small form-factor Computer on Module/System on Module that comes in both quad core and dual core versions based on NXP (formerly Freescale) i.MX 6Q and NXP i.MX 6D SoCs respectively. The Cortex A9 quad core and dual core CPU peaks at 1 GHz for commercial temperature variant, while the industrial temperature variant has a peak frequency of 800 MHz.

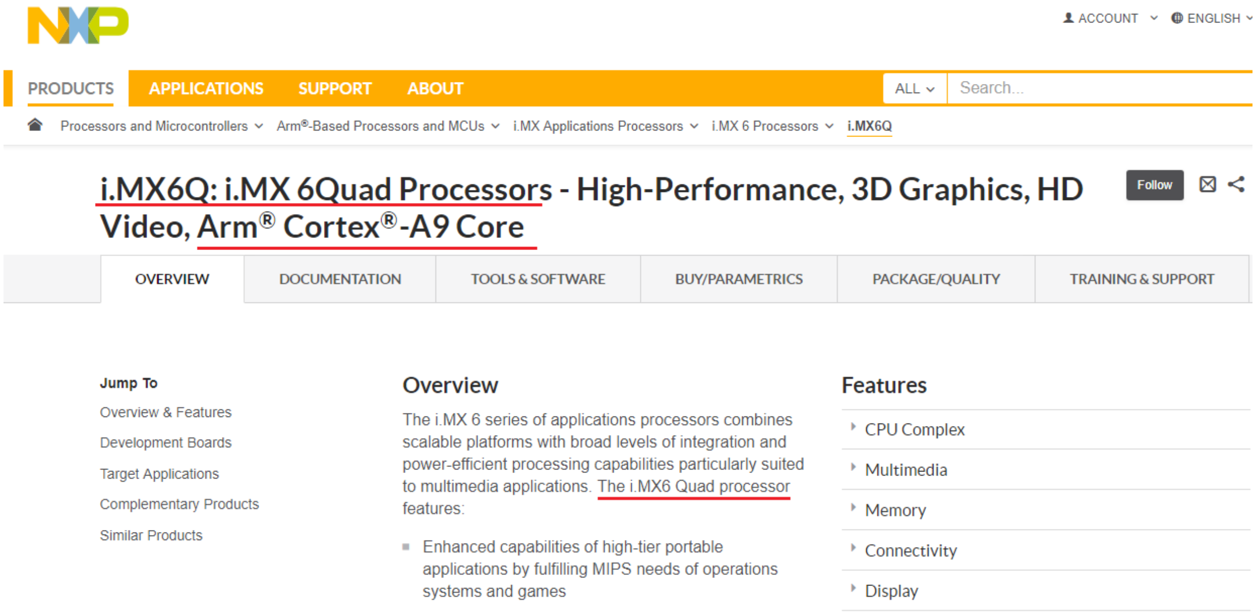
Extensive Industrial and High-speed interfaces

The NXP (formerly Freescale) i.MX 6 based Apalis iMX6 System on Module targets a wide range of industrial applications, including: Automotive & Infotainment, Navigation, Industrial Automation, HMIs, Avionics, Robotics, and much more.

The module exposes wide range of industrial interfaces including CAN, UART, I2C, USB, PCIe, SATA, and many more.

Apalis iMX6 is a reference platform for Qt for Device Creation. For more details, check here

(*E.g.*, https://docs.toradex.com/101770-apalis-imx6-datasheet.pdf).



i.MX6Q: i.MX 6Quad Processors - High-Performance, 3D Graphics, HD Video, Arm® Cortex®-A9 Core

Overview

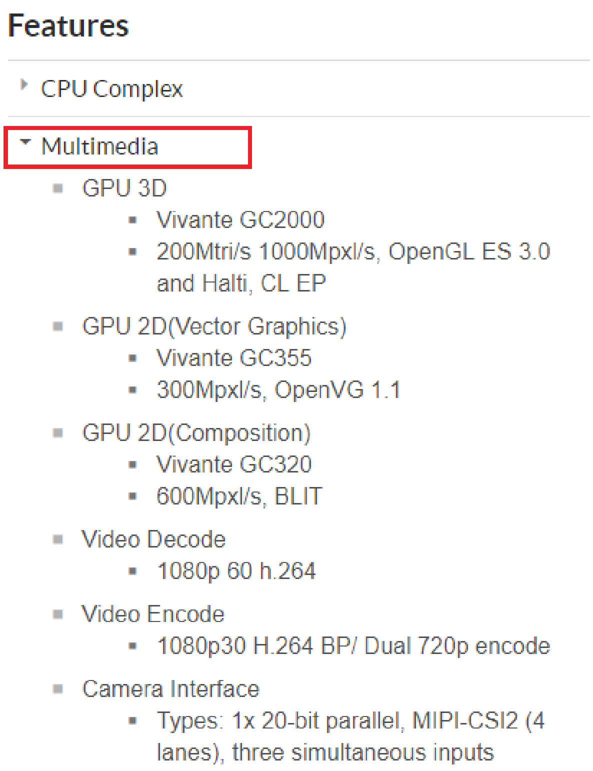
The i.MX 6 series of applications processors combines scalable platforms with broad levels of integration and power-efficient processing capabilities particularly suited to multimedia applications. The i.MX6 Quad processor features:

- Enhanced capabilities of high-tier portable applications by fulfilling MIPS needs of operations systems and games

Features

- CPU Complex
- Multimedia
- Memory
- Connectivity
- Display

(*E.g.*, https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/i.mx-applications-processors/i.mx-6-processors/i.mx-6quad-processors-high-performance-3d-graphics-hd-video-arm-cortex-a9-core:i.MX6Q).

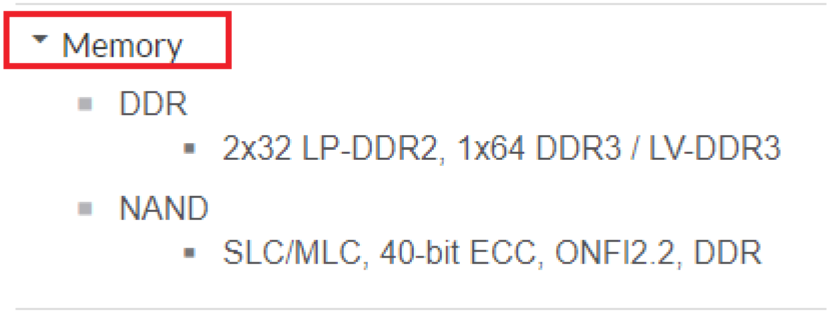


Features

- CPU Complex
- Multimedia
  - GPU 3D
    - Vivante GC2000
    - 200Mtri/s 1000Mpxl/s, OpenGL ES 3.0 and Halti, CL EP
  - GPU 2D(Vector Graphics)
    - Vivante GC355
    - 300Mpxl/s, OpenVG 1.1
  - GPU 2D(Composition)
    - Vivante GC320
    - 600Mpxl/s, BLIT
  - Video Decode
    - 1080p 60 h.264
  - Video Encode
    - 1080p30 H.264 BP/ Dual 720p encode
  - Camera Interface
    - Types: 1x 20-bit parallel, MIPI-CSI2 (4 lanes), three simultaneous inputs

(*E.g.*, *id.*).

29. The accused product comprises a plurality of media processing units (*e.g.*, ARM cortex A9 Dual/Quad Core processors), each media processing unit having an input/output coupled to at least one of the addressable memory input/outputs. As shown below, the accused product comprises ARM cortex A9 Dual/Quad Core processors, each processor comprises a NEON media coprocessor and acts as a media processing unit. The ARM processors are coupled to the memory system. The processors receive instructions and data from the memory system by multiple internal inputs and provides processed data to the memory system by multiple internal outputs.

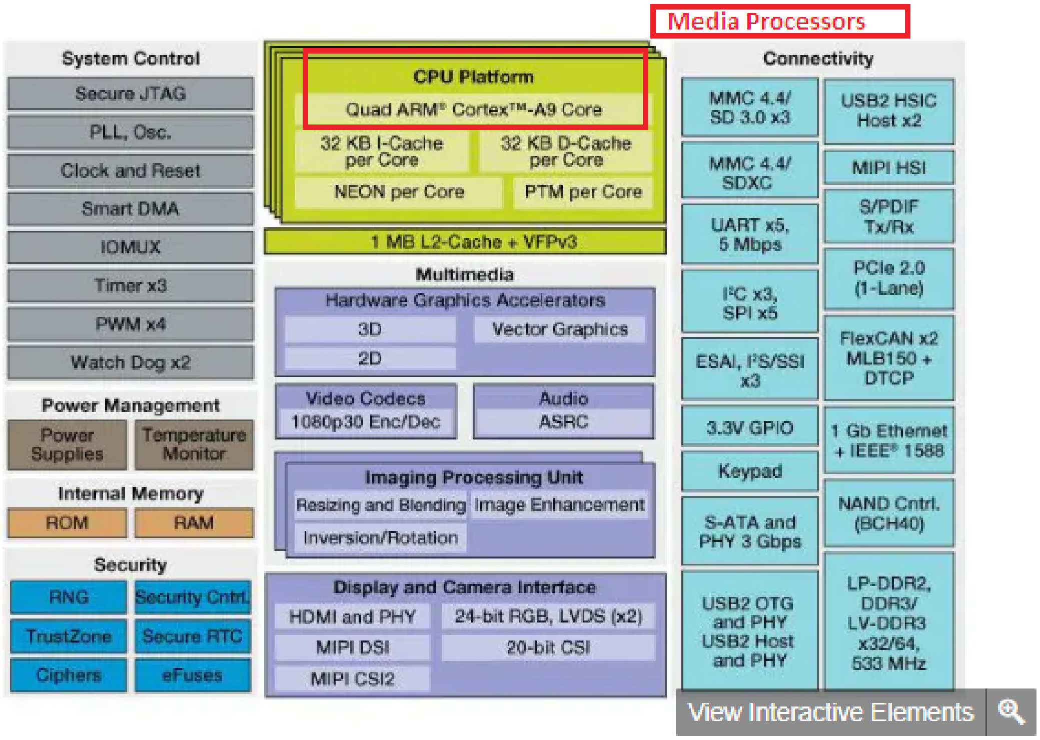


▾ Memory

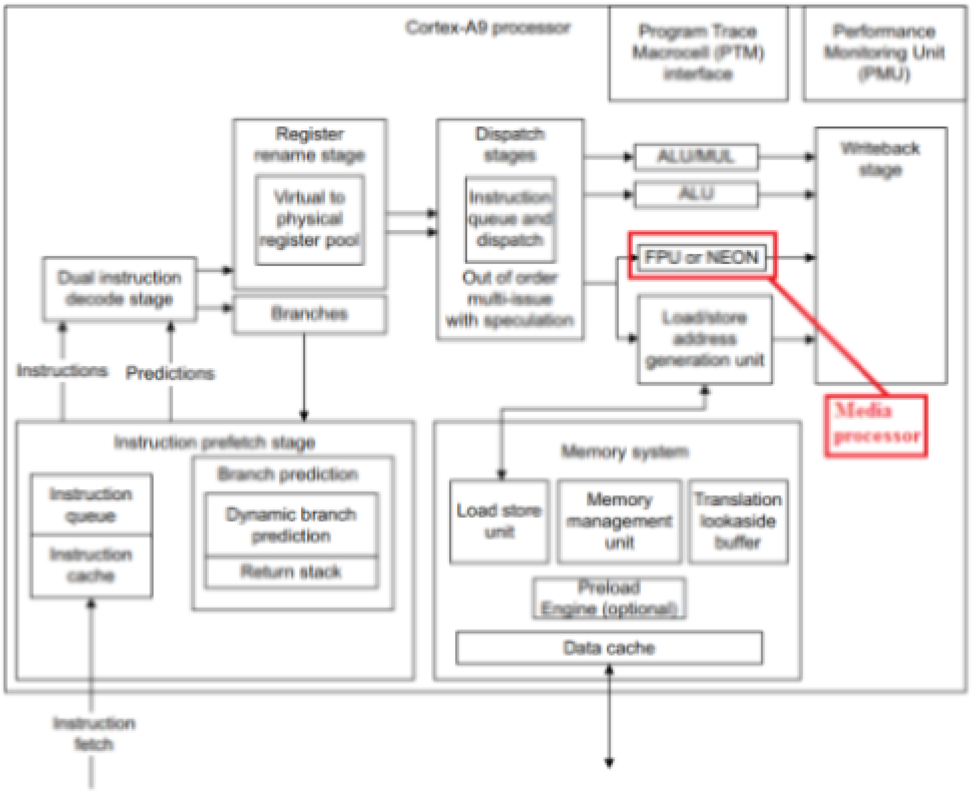
- DDR
  - 2x32 LP-DDR2, 1x64 DDR3 / LV-DDR3
- NAND
  - SLC/MLC, 40-bit ECC, ONFI2.2, DDR

(*e.g.*, *id.*).

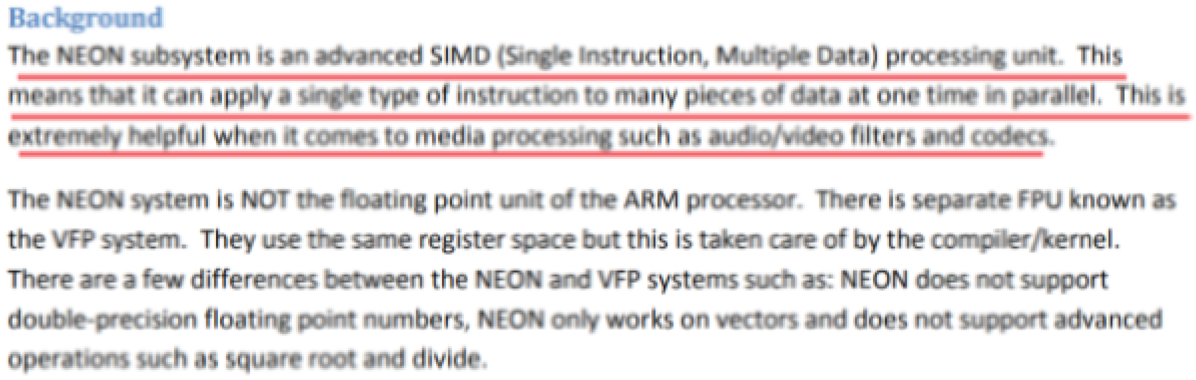
i.MX 6Quad Multimedia Applications Processor Block Diagram



(*E.g.*, *id.*).


Cortex-A9 processor
Program Trace Macrocell (PTM) interface
Performance Monitoring Unit (PMU)
Register rename stage
Virtual to physical register pool
Dispatch stages
Instruction queue and dispatch
Out of order multi-issue with speculation
ALU/MUL
ALU
FPU or NEON
Load/store address generation unit
Writeback stage
Media processor
Dual instruction decode stage
Branches
Instructions
Predictions
Instruction prefetch stage
Instruction queue
Instruction cache
Branch prediction
Dynamic branch prediction
Return stack
Memory system
Load store unit
Memory management unit
Translation lookaside buffer
Preload Engine (optional)
Data cache
Instruction fetch

(*E.g.*, http://infocenter.arm.com/help/topic/com.arm.doc.ddi0388f/DDI0388F_cortex_a9_r2p2_trm.pdf).

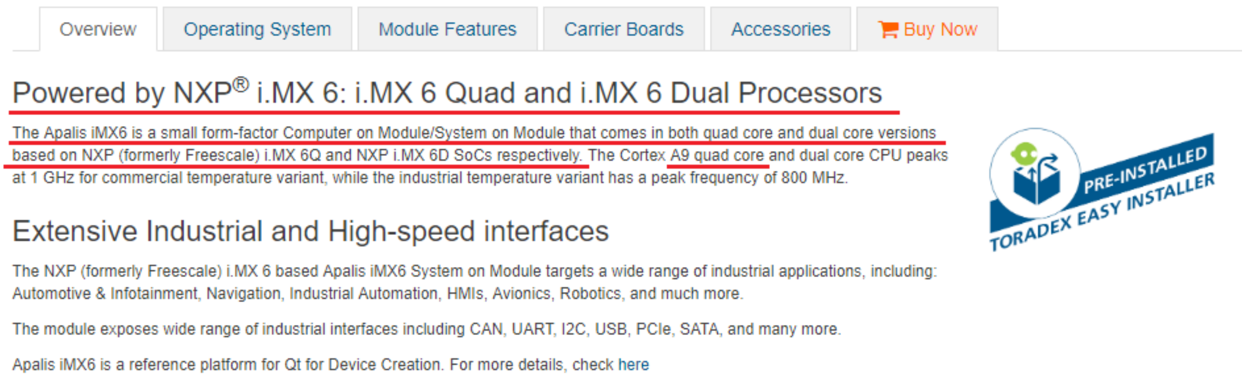

Background

The NEON subsystem is an advanced SIMD (Single Instruction, Multiple Data) processing unit. This means that it can apply a single type of instruction to many pieces of data at one time in parallel. This is extremely helpful when it comes to media processing such as audio/video filters and codecs.

The NEON system is NOT the floating point unit of the ARM processor. There is separate FPU known as the VFP system. They use the same register space but this is taken care of by the compiler/kernel. There are a few differences between the NEON and VFP systems such as: NEON does not support double-precision floating point numbers, NEON only works on vectors and does not support advanced operations such as square root and divide.

(*e.g.*, http://www.add.ece.ufl.edu/4924/docs/arm/ARM%20NEON%20Development.pdf).

30. The accused product comprises media processors with each processor comprising a multiplier (*e.g.*, an Integer MUL or FP MUL) having a data input coupled to the media processing unit input/output, an instruction input coupled to the media processing unit input/output, and a data output coupled to the media processing unit input/output. As shown below, the accused product comprises multiple ARM cortex-A9 Dual/Quad core processor, each processor comprises a NEON media coprocessor and acts as a media processing unit. NEON media coprocessor comprises a multiplier which is coupled to the inputs/outputs of the processor. Upon information and belief, the multiplier comprises a data input, an instruction input, and a data output coupled to the input/output of the processor.



Powered by NXP® i.MX 6: i.MX 6 Quad and i.MX 6 Dual Processors

The Apalis iMX6 is a small form-factor Computer on Module/System on Module that comes in both quad core and dual core versions based on NXP (formerly Freescale) i.MX 6Q and NXP i.MX 6D SoCs respectively. The Cortex A9 quad core and dual core CPU peaks at 1 GHz for commercial temperature variant, while the industrial temperature variant has a peak frequency of 800 MHz.

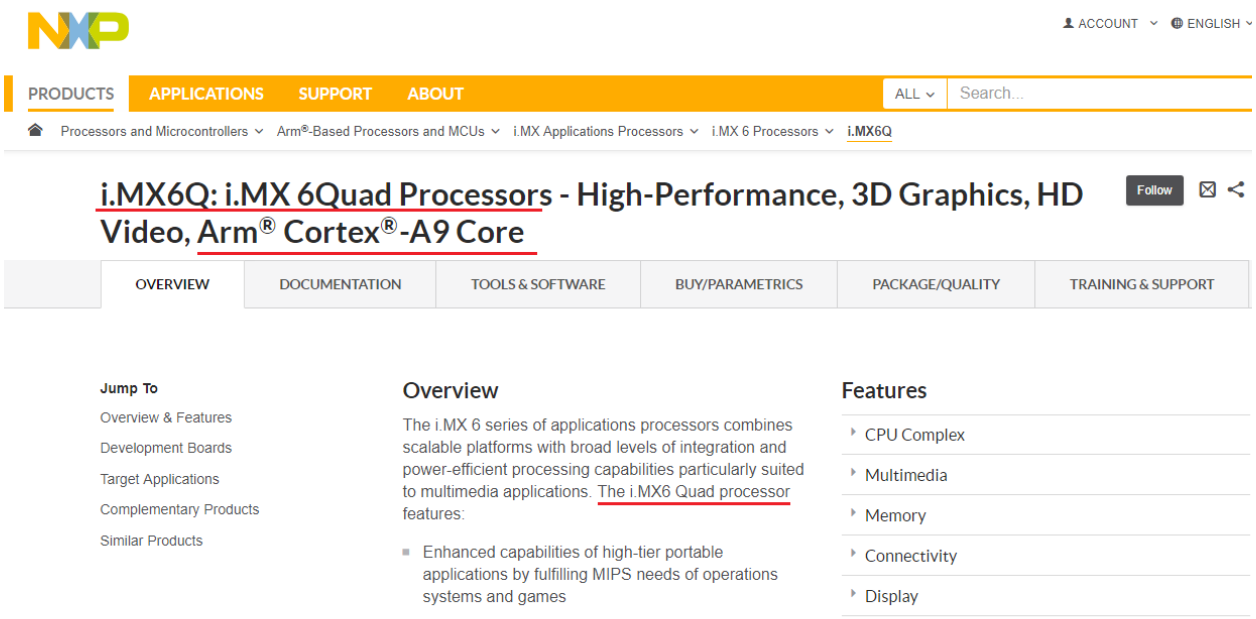
Extensive Industrial and High-speed interfaces

The NXP (formerly Freescale) i.MX 6 based Apalis iMX6 System on Module targets a wide range of industrial applications, including: Automotive & Infotainment, Navigation, Industrial Automation, HMIs, Avionics, Robotics, and much more.

The module exposes wide range of industrial interfaces including CAN, UART, I2C, USB, PCIe, SATA, and many more.

Apalis iMX6 is a reference platform for Qt for Device Creation. For more details, check here

(*e.g.*, https://www.toradex.com/computer-on-modules/apalis-arm-family/nxp-freescale-imx-6).



i.MX6Q: i.MX 6Quad Processors - High-Performance, 3D Graphics, HD Video, Arm® Cortex®-A9 Core

Jump To

Overview & Features
Development Boards
Target Applications
Complementary Products
Similar Products

Overview

The i.MX 6 series of applications processors combines scalable platforms with broad levels of integration and power-efficient processing capabilities particularly suited to multimedia applications. The i.MX6 Quad processor features:

- Enhanced capabilities of high-tier portable applications by fulfilling MIPS needs of operations systems and games

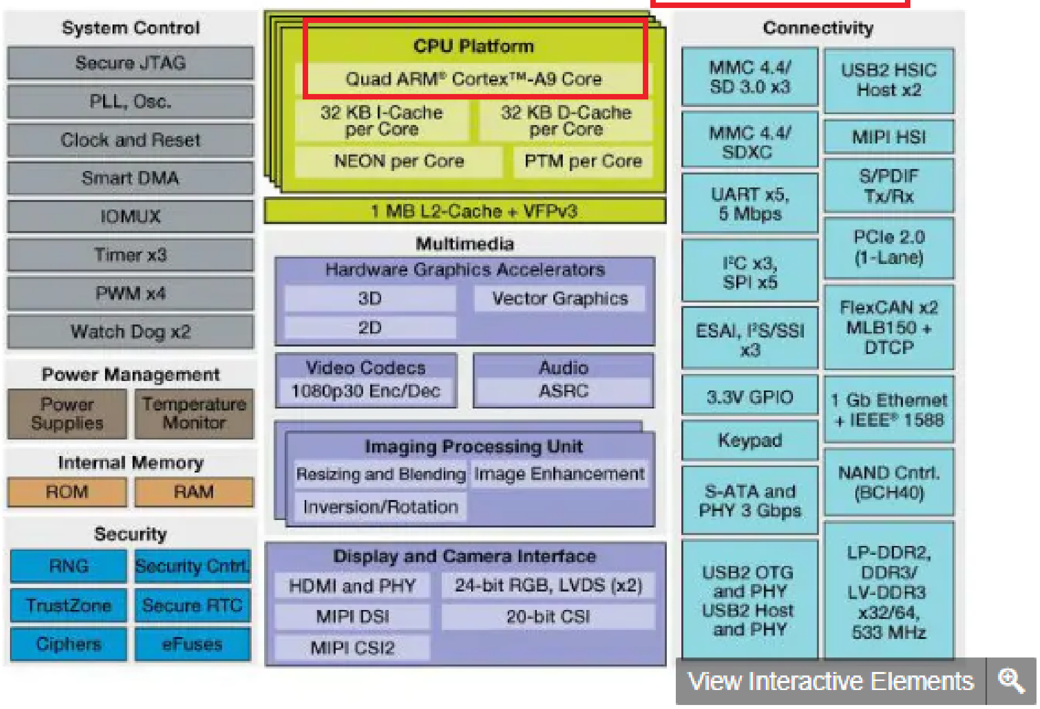
Features

- CPU Complex
- Multimedia
- Memory
- Connectivity
- Display

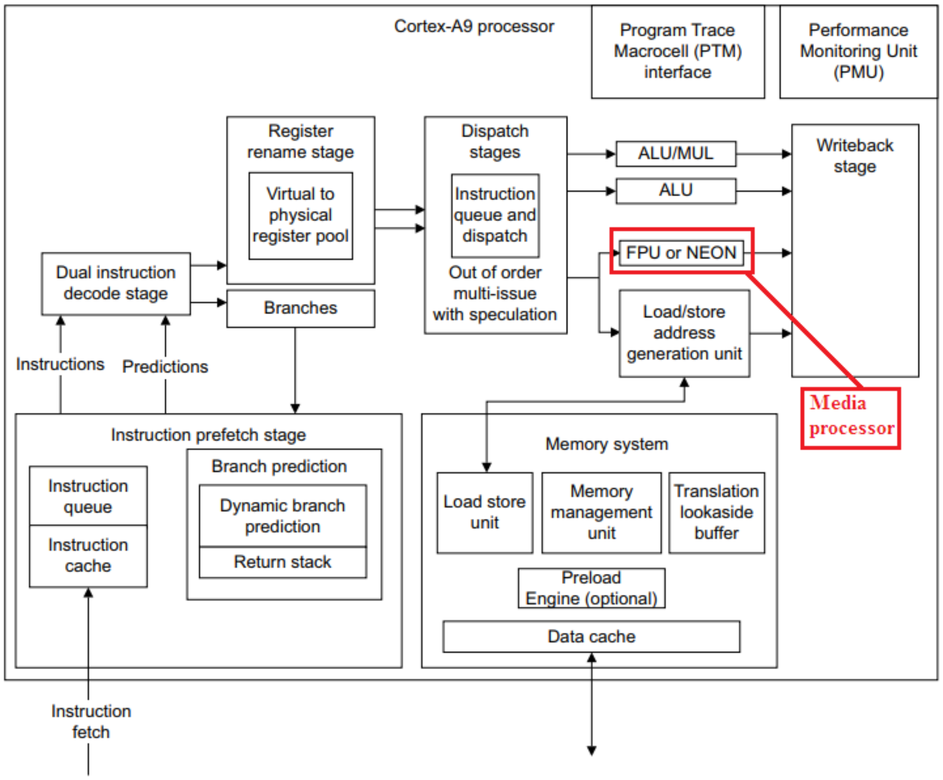
(*E.g.*, https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/i.mx-applications-processors/i.mx-6-processors/i.mx-6quad-processors-high-performance-3d-graphics-hd-video-arm-cortex-a9-core:i.MX6Q).

i.MX 6Quad Multimedia Applications Processor Block Diagram

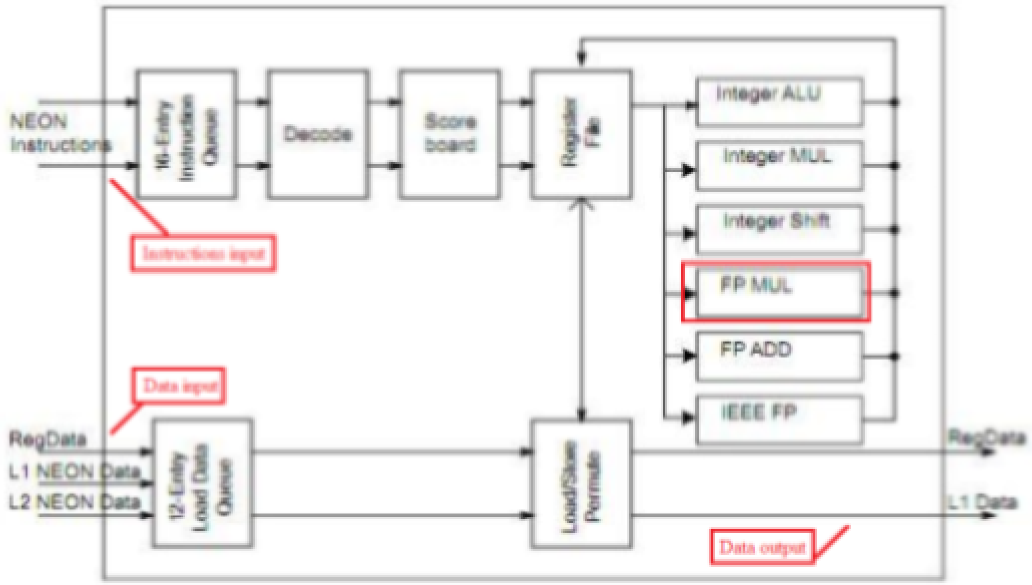
Media Processors



(*E.g.*, *id.*).

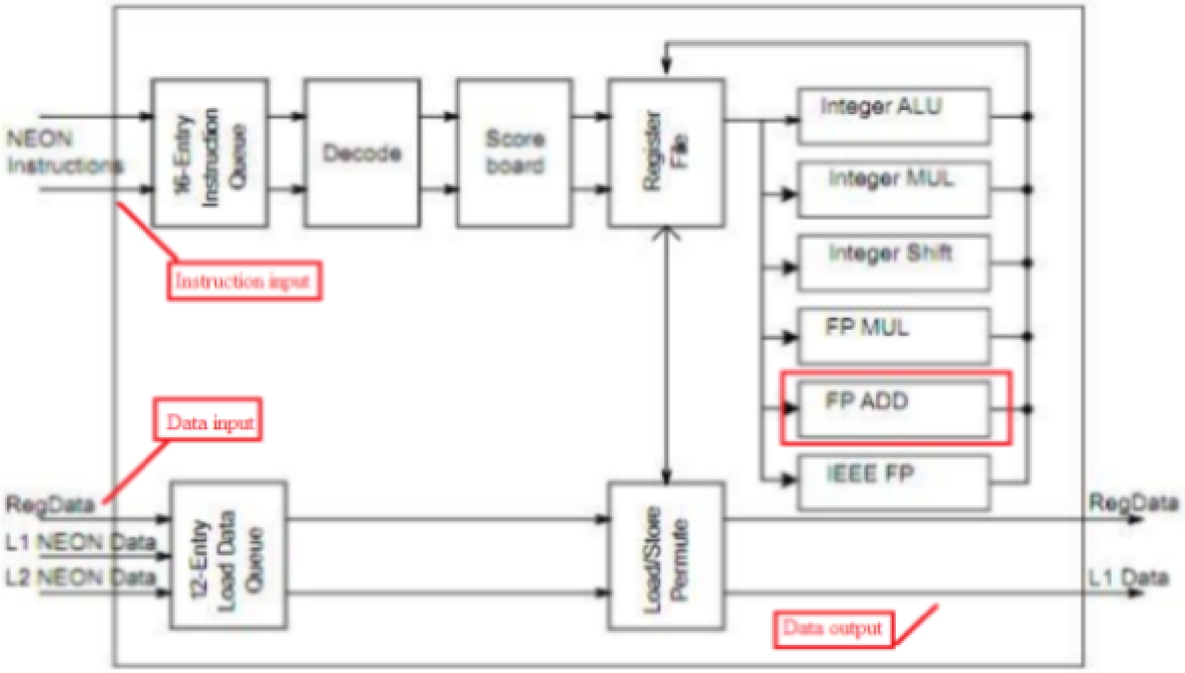

Cortex-A9 processor
Program Trace Macrocell (PTM) interface
Performance Monitoring Unit (PMU)
Register rename stage
Virtual to physical register pool
Dispatch stages
Instruction queue and dispatch
Out of order multi-issue with speculation
ALU/MUL
ALU
FPU or NEON
Load/store address generation unit
Writeback stage
Media processor
Dual instruction decode stage
Branches
Instructions
Predictions
Instruction prefetch stage
Instruction queue
Instruction cache
Branch prediction
Dynamic branch prediction
Return stack
Memory system
Load store unit
Memory management unit
Translation lookaside buffer
Preload Engine (optional)
Data cache
Instruction fetch

(*e.g.*, http://infocenter.arm.com/help/topic/com.arm.doc.ddi0388f/DDI0388F_cortex_a9_r2p2_trm.pdf).



(*E.g.*, http://www.add.ece.ufl.edu/4924/docs/arm/ARM%20NEON %20Development.pdf).

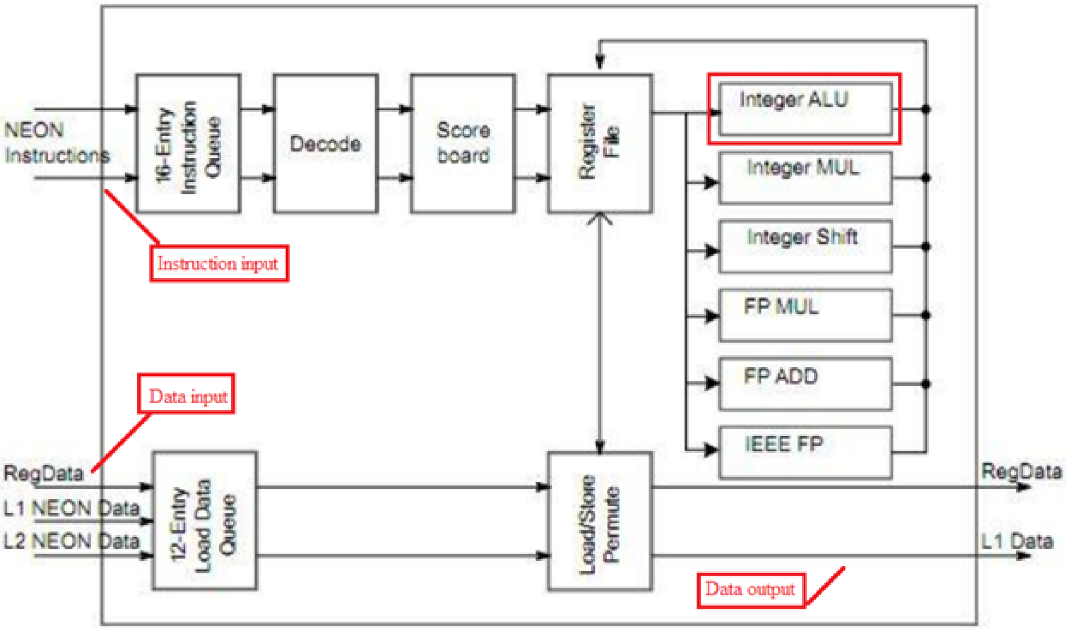
31. The accused product comprises media processors with each processor comprising an arithmetic unit (*e.g.,* an FP ADD) having a data input coupled to the media processing unit input/output, an instruction input coupled to the media processing unit input/output, and a data output coupled to the media processing unit input/output. As shown below, the accused product comprises multiple ARM cortex-A9 Dual/Quad core processor, each processor comprises a NEON media coprocessor and acts as a media processing unit. NEON media coprocessor comprises an arithmetic unit which is coupled to the inputs/outputs of the processor. Upon information and belief, the arithmetic unit comprises a data input, an instruction input, and a data output coupled to the input/output of the processor.



(*E.g.*, http://www.add.ece.ufl.edu/4924/docs/arm/ARM%20NEON%20Development.pdf).

32. The accused product comprises media processors with each processor comprising an arithmetic logic unit (*e.g.*, an ALU) having a data input coupled to the media processing unit input/output, an instruction input coupled to the media processing unit input/output, and a data output coupled to the media processing unit input/output, capable of operating concurrently with at least one selected from the multiplier (*e.g.*, an Integer MUL or FP MUL) and arithmetic unit (*e.g.*, a FP ADD). As shown below, the accused product comprises multiple ARM cortex-A9 Dual/Quad core processor, each processor comprises a NEON media coprocessor and acts as a media processing unit. NEON media coprocessor comprises an arithmetic logical unit which is coupled to the inputs/outputs of the processor. Upon information and belief, the arithmetic logical unit comprises a data input, an instruction input, and a data output coupled to the

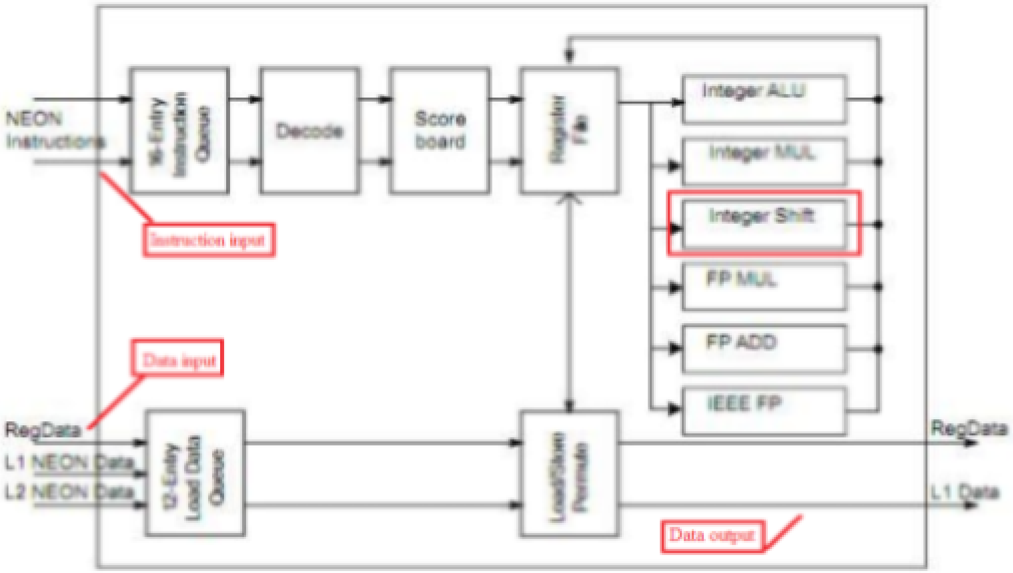
input/output of the processor. Upon information and belief, the arithmetic logical unit (*e.g.*, the Integer ALU) is capable of operating concurrently with at least one selected from the multiplier (*e.g.*, the Integer MUL or FP MUL) and arithmetic unit (*e.g.*, the FP ADD).



(*E.g.*, http://www.add.ece.ufl.edu/4924/docs/arm/ARM%20NEON%20Development.pdf).

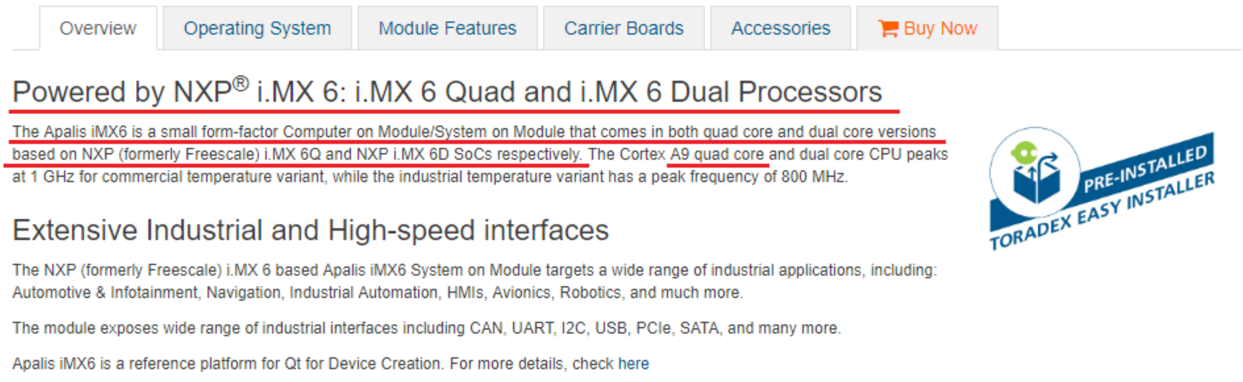
33. The accused product comprises media processors with each processor comprising a bit manipulation unit (*e.g.*, an Integer Shift unit) having a data input coupled to the media processing unit input/output, an instruction input coupled to the media processing unit input/output, and a data output coupled to the media processing unit input/output, capable of operating concurrently with the arithmetic logic unit (*e.g.,* an Integer ALU) and at least one selected from the multiplier (*e.g.,* an Integer MUL or FP MUL) and arithmetic unit (*e.g.,* a FP ADD). As shown below, the accused product comprises multiple ARM cortex-A9 Dual/Quad core processors, each processor

comprising a NEON media coprocessor that acts as a media processing unit. The NEON media coprocessor comprises an integer shift unit (i.e., bit manipulation unit) which is coupled to the inputs/outputs of the processor. Upon information and belief, the integer shift unit (*i.e.*, bit manipulation unit) comprises a data input, an instruction input, and a data output coupled to the input/output of the processor. Upon information and belief, the integer shift unit (*i.e.*, bit manipulation unit) is capable of operating concurrently with the arithmetic logic unit (*e.g.,* the Integer ALU) and at least one selected from the multiplier (*e.g.,* the Integer MUL or FP MUL) and arithmetic unit (*e.g.,* the FP ADD).



(*E.g.*, http://www.add.ece.ufl.edu/4924/docs/arm/ARM%20NEON%20Development.pdf).

34. The accused product comprises a plurality of media processors (*e.g.*, ARM cortex-A9 Dual/Quad core processors) for performing at least one operation, simultaneously with the performance of other operations by other media processing units (*e.g.*, other ARM cortex-A9 Dual/Quad core processors on the same chip).



Powered by NXP® i.MX 6: i.MX 6 Quad and i.MX 6 Dual Processors

The Apalis iMX6 is a small form-factor Computer on Module/System on Module that comes in both quad core and dual core versions based on NXP (formerly Freescale) i.MX 6Q and NXP i.MX 6D SoCs respectively. The Cortex A9 quad core and dual core CPU peaks at 1 GHz for commercial temperature variant, while the industrial temperature variant has a peak frequency of 800 MHz.

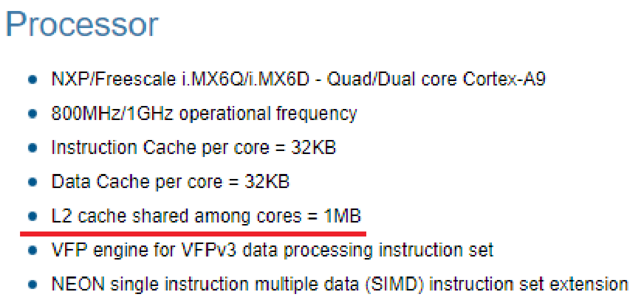
Extensive Industrial and High-speed interfaces

The NXP (formerly Freescale) i.MX 6 based Apalis iMX6 System on Module targets a wide range of industrial applications, including: Automotive & Infotainment, Navigation, Industrial Automation, HMIs, Avionics, Robotics, and much more.

The module exposes wide range of industrial interfaces including CAN, UART, I2C, USB, PCIe, SATA, and many more.

Apalis iMX6 is a reference platform for Qt for Device Creation. For more details, check here

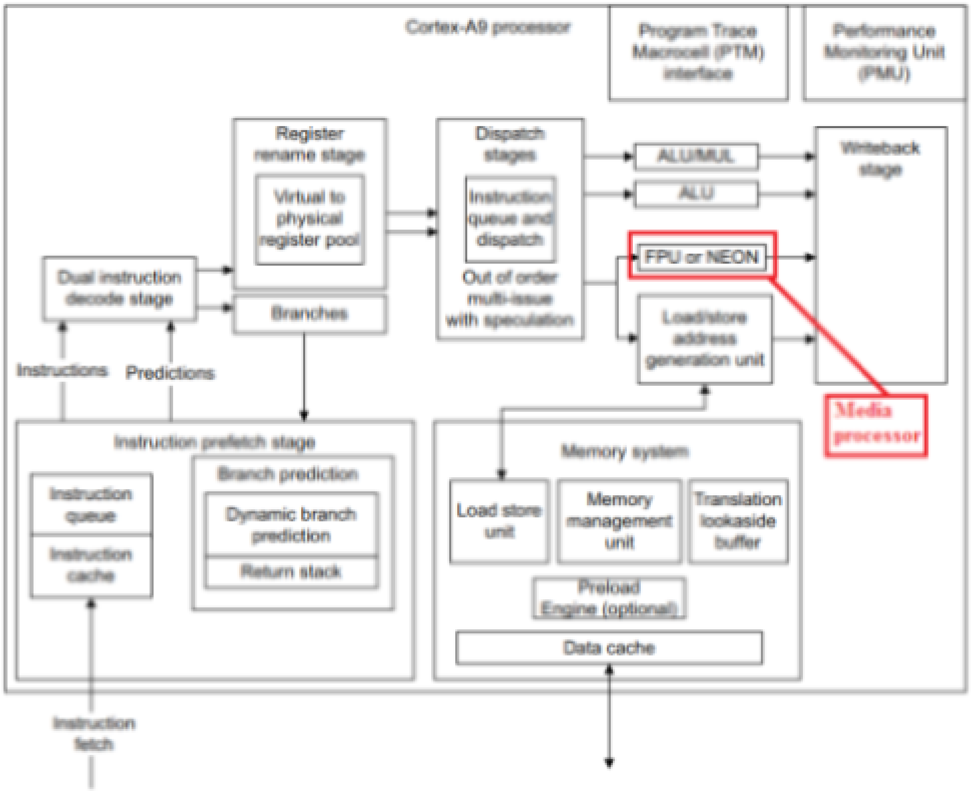
(*e.g.*, https://www.toradex.com/computer-on-modules/apalis-arm-family/nxp-freescale-imx-6).



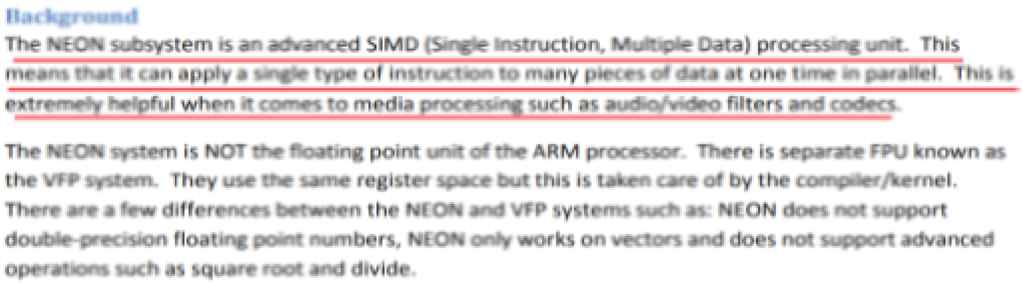
Processor

- NXP/Freescale i.MX6Q/i.MX6D - Quad/Dual core Cortex-A9
- 800MHz/1GHz operational frequency
- Instruction Cache per core = 32KB
- Data Cache per core = 32KB
- L2 cache shared among cores = 1MB
- VFP engine for VFPv3 data processing instruction set
- NEON single instruction multiple data (SIMD) instruction set extension

(*e.g.*, https://developer.toradex.com/products/apalis-imx6).


Cortex-A9 processor
Program Trace Macrocell (PTM) interface
Performance Monitoring Unit (PMU)
Register rename stage
Virtual to physical register pool
Dispatch stages
Instruction queue and dispatch
Out of order multi-issue with speculation
ALU/MUL
ALU
FPU or NEON
Load/store address generation unit
Writeback stage
Media processor
Dual instruction decode stage
Branches
Instructions
Predictions
Instruction prefetch stage
Instruction queue
Instruction cache
Branch prediction
Dynamic branch prediction
Return stack
Memory system
Load store unit
Memory management unit
Translation lookaside buffer
Preload Engine (optional)
Data cache
Instruction fetch

(*E.g.*, http://infocenter.arm.com/help/topic/com.arm.doc.ddi0388f/DDI0388F_cortex_a9_r2p2_trm.pdf).



Background

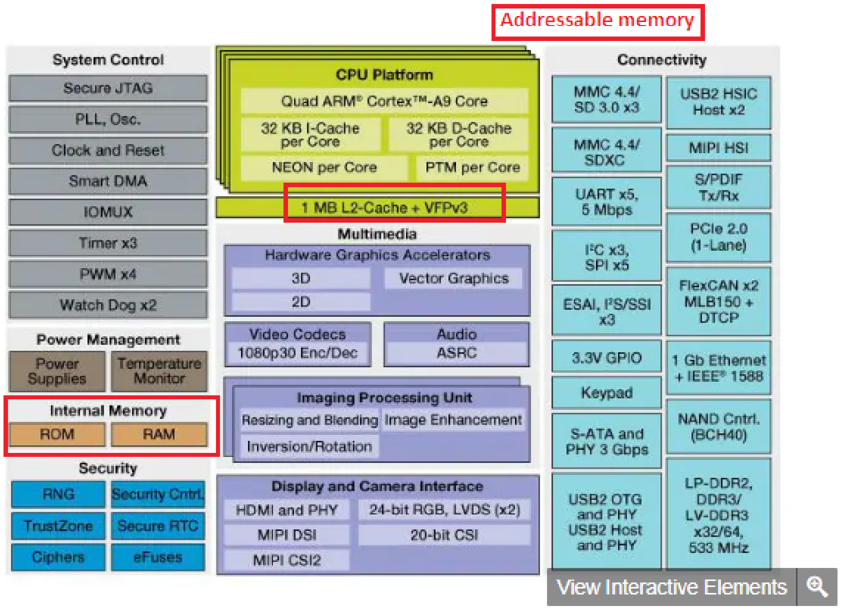
The NEON subsystem is an advanced SIMD (Single Instruction, Multiple Data) processing unit. This means that it can apply a single type of instruction to many pieces of data at one time in parallel. This is extremely helpful when it comes to media processing such as audio/video filters and codecs.

The NEON system is NOT the floating point unit of the ARM processor. There is separate FPU known as the VFP system. They use the same register space but this is taken care of by the compiler/kernel. There are a few differences between the NEON and VFP systems such as: NEON does not support double-precision floating point numbers, NEON only works on vectors and does not support advanced operations such as square root and divide.

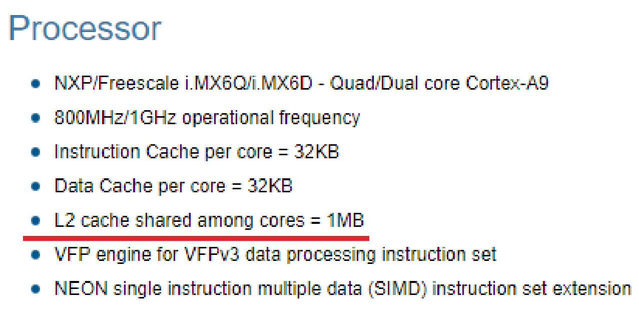
(*E.g.*, http://www.add.ece.ufl.edu/4924/docs/arm/ARM%20NEON%20Development.pdf).

35. The accused product comprises a plurality of media processors (*e.g.*, ARM cortex-A9 Dual/Quad core processors), each processor receiving at the media processor input/output an instruction and data from the memory, and processing the data responsive to the instruction received to produce at least one result. As shown below, each ARM cortex-A9 Dual/Quad core media processor comprises a NEON media coprocessor which receives instructions and data from memory and processes the data responsive to the instruction received in order to produce a result.

i.MX 6Quad Multimedia Applications Processor Block Diagram



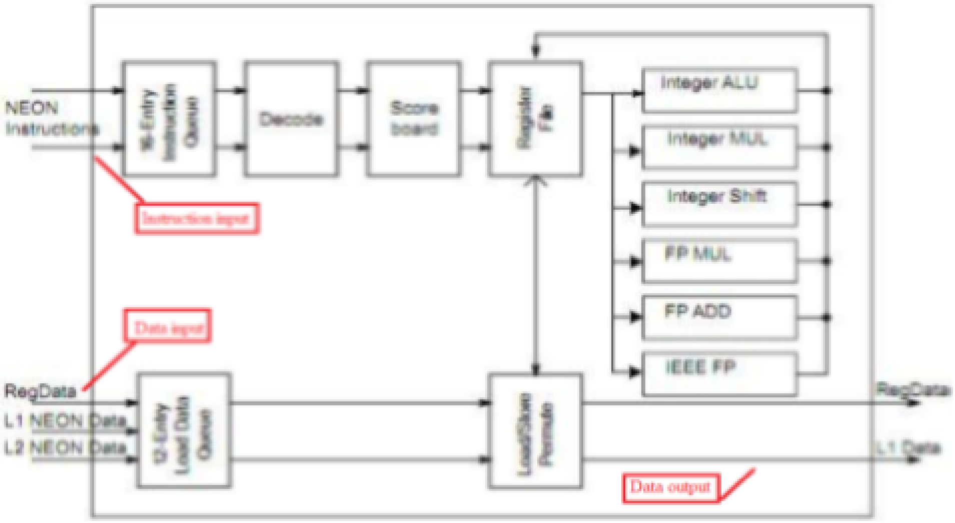
(*E.g.*, https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/i.mx-applications-processors/i.mx-6-processors/i.mx-6quad-processors-high-performance-3d-graphics-hd-video-arm-cortex-a9-core:i.MX6Q).



Processor

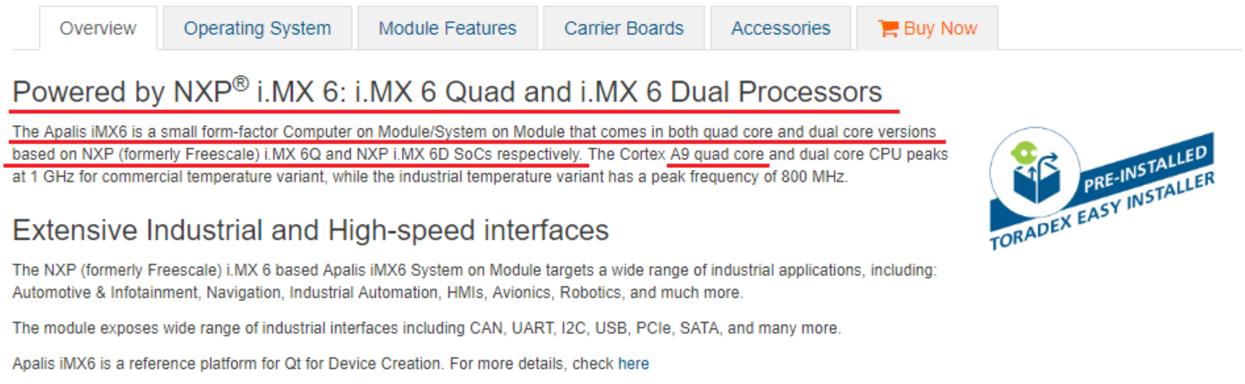
- NXP/Freescale i.MX6Q/i.MX6D - Quad/Dual core Cortex-A9
- 800MHz/1GHz operational frequency
- Instruction Cache per core = 32KB
- Data Cache per core = 32KB
- L2 cache shared among cores = 1MB
- VFP engine for VFPv3 data processing instruction set
- NEON single instruction multiple data (SIMD) instruction set extension

(*e.g.*, https://developer.toradex.com/products/apalis-imx6).



(*E.g.*, http://www.add.ece.ufl.edu/4924/docs/arm/ARM%20NEON%20Development.pdf).

36. The accused product comprises a plurality of media processors (*e.g.*, ARM cortex-A9 Dual/Quad core processors), each processor providing at least one of the at least one result at the media processor input/output. (*Id.*).



Powered by NXP® i.MX 6: i.MX 6 Quad and i.MX 6 Dual Processors

The Apalis iMX6 is a small form-factor Computer on Module/System on Module that comes in both quad core and dual core versions based on NXP (formerly Freescale) i.MX 6Q and NXP i.MX 6D SoCs respectively. The Cortex A9 quad core and dual core CPU peaks at 1 GHz for commercial temperature variant, while the industrial temperature variant has a peak frequency of 800 MHz.

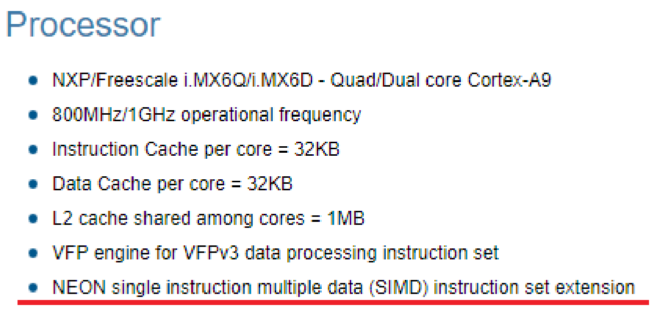
Extensive Industrial and High-speed interfaces

The NXP (formerly Freescale) i.MX 6 based Apalis iMX6 System on Module targets a wide range of industrial applications, including: Automotive & Infotainment, Navigation, Industrial Automation, HMIs, Avionics, Robotics, and much more.

The module exposes wide range of industrial interfaces including CAN, UART, I2C, USB, PCIe, SATA, and many more.

Apalis iMX6 is a reference platform for Qt for Device Creation. For more details, check here

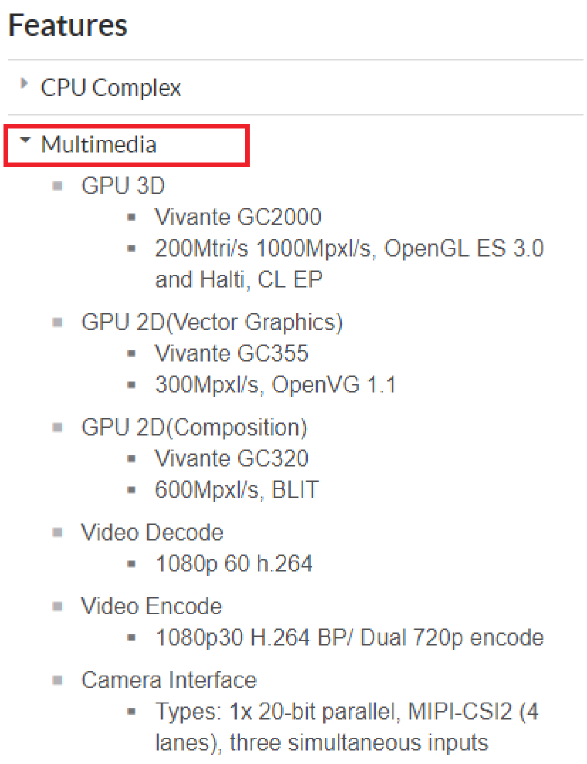
(*E.g.*, https://www.toradex.com/computer-on-modules/apalis-arm-family/nxp-freescale-imx-6).



Processor

- NXP/Freescale i.MX6Q/i.MX6D - Quad/Dual core Cortex-A9
- 800MHz/1GHz operational frequency
- Instruction Cache per core = 32KB
- Data Cache per core = 32KB
- L2 cache shared among cores = 1MB
- VFP engine for VFPv3 data processing instruction set
- NEON single instruction multiple data (SIMD) instruction set extension

(*E.g.*, https://developer.toradex.com/products/apalis-imx6).



Features

- CPU Complex
- Multimedia
  - GPU 3D
    - Vivante GC2000
    - 200Mtri/s 1000Mpxl/s, OpenGL ES 3.0 and Halti, CL EP
  - GPU 2D(Vector Graphics)
    - Vivante GC355
    - 300Mpxl/s, OpenVG 1.1
  - GPU 2D(Composition)
    - Vivante GC320
    - 600Mpxl/s, BLIT
  - Video Decode
    - 1080p 60 h.264
  - Video Encode
    - 1080p30 H.264 BP/ Dual 720p encode
  - Camera Interface
    - Types: 1x 20-bit parallel, MIPI-CSI2 (4 lanes), three simultaneous inputs

(*e.g.*, https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/i.mx-applications-processors/i.mx-6-processors/i.mx-6quad-processors-high-performance-3d-graphics-hd-video-arm-cortex-a9-core:i.MX6Q).

37. Plaintiff has been damaged as a result of Defendant's infringing conduct. Defendant is thus liable to Plaintiff for damages in an amount that adequately compensates Plaintiff for such Defendant's infringement of the '434 patent, *i.e.*, in an amount that by law cannot be less than would constitute a reasonable royalty for the use of the patented technology, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

38. On information and belief, Defendant has had at least constructive notice of the '434 patent by operation of law, and there are no marking requirements that have not been complied with.

IV. JURY DEMAND

Plaintiff, under Rule 38 of the Federal Rules of Civil Procedure, requests a trial by jury of any issues so triable by right.

V. PRAYER FOR RELIEF

WHEREFORE, Plaintiff respectfully requests that the Court find in its favor and against Defendant, and that the Court grant Plaintiff the following relief:

a. Judgment that one or more claims of United States Patent No. 6,289,434 have been infringed, either literally and/or under the doctrine of equivalents, by Defendant;

b. Judgment that Defendant account for and pay to Plaintiff all damages to and costs incurred by Plaintiff because of Defendant's infringing activities and other conduct complained of herein, and an accounting of all infringements and damages not presented at trial;

c. That Plaintiff be granted pre-judgment and post-judgment interest on the damages caused by Defendant's infringing activities and other conduct complained of herein; and

d. That Plaintiff be granted such other and further relief as the Court may deem just and proper under the circumstances.

//

//

//

//

//

March 27, 2019

Respectfully Submitted,

By */s/Philip P. Mann*
Philip P. Mann, WSBA No: 28860
MANN LAW GROUP PLLC
107 Spring St.
Seattle, Washington 98104
(206) 436-0900
Fax (866) 341-5140
phil@mannlawgroup.com

David R. Bennett
(Application for Admission Pro Hac
Vice to be filed)
Direction IP Law
P.O. Box 14184
Chicago, IL 60614-0184
(312) 291-1667
dbennett@directionip.com

Attorneys for Plaintiff Altair Logix LLC