Crossover Embedded Processors – Bridging the gap between performance and usability

Overview

At the heart of thousands of smart, connected IoT products are embedded processors which have a unique ability to deliver tailored functionality and performance. As we look ahead at the expanding possibilities of a connected world, driven by the promise of edge computing with embedded artificial intelligence, we find an increasing need for low-power embedded solutions that are cost-effective and scalable, but also deliver high performance computation, security, and enhanced user experience. Traditional embedded processors address some of these needs, but are limited in scalability across all the applications. We believe that scalability gap can be bridged, but it requires breaking the boundary between applications processors and MCUs to create a new class of 'crossover' embedded processors. The proposed crossover processors are applications processors built with an MCU core, architected to deliver high performance and functional capabilities of applications processors, but with the ease-of-use and real-time low-power operation of traditional MCUs. In this white paper, we describe how such crossover processors can be architected and the valuable benefits they provide.

Table of Contents

Applications Processors vs MCUs

For many years, embedded processing has fallen into two distinct camps based on the necessity of the design. Situations calling for affordable and flexible usability would rely on MCUs. Designs that exceeded the capacity of an MCU would instead turn to an applications processor. However, it has not been easy for embedded designers to seamlessly scale across MCUs and applications processors. Typical pain points that the embedded designers face today when deciding between an MCU and an applications processor, include:

- Needing capabilities beyond what an MCU can offer (greater performance, more displays, increased connectivity options), without increasing cost or complexity
- Lacking experienced staff and/or budgetary resources to support Linux-based applications processor designs
- Requiring both a real-time system and applications processor-level of performance and integration
- Having to reduce overall bill-of-materials cost in an applications processor-based design, while maintaining the performance level

Growing consumer demand for enhanced user experience and increasing functionality in their smart, secure products has driven the growth of both MCUs and applications processors. However, the edge is getting smarter. The future of the connected world is the endless possibilities of edge computing with embedded artificial intelligence. The essentials for this future are edge devices that have enhanced computational capability, support for advanced display and graphics, and provide reliable security and assured privacy. In many of these new use cases, traditional applications processors or MCUs perfectly address the needs. A perfect example of this issue would be if an appliance designer wanted to add Internet of Things (IoT) capabilities (i.e., data processing, edge computation, wireless connectivity, display support) to their product design without drastically increasing the per-unit cost or greatly extending their time-to-market with an extensive redesign.

The 'Crossover' Processor—From Performance Gap to New Solution Space

We believe that the evolution of the smarter, connected world should be driven by innovation and not by a constrained choice between an MCU and an applications processor. Embedded product designers should be able to freely choose a processor that best delivers the innovation in their designs, rather than have the processor choice limit the innovation that is possible in their end design. The flexibility to scale across MCUs and applications processors is possible, but it requires a new class of crossover processors that break down the technological chasm between high-end MCUs and low-end applications processors. These **crossover processors** targeted for consumer, industrial and the growing IoT applications, deliver performance, functionality and capabilities of an applications processor-based design, but with the ease-of-use, low-power, and real-time operation with low interrupt latency of an MCU-based design. Furthermore, the crossover processors are architected to lower overall system cost by eliminating the need for embedding flash, external DDR memories and power management ICs.

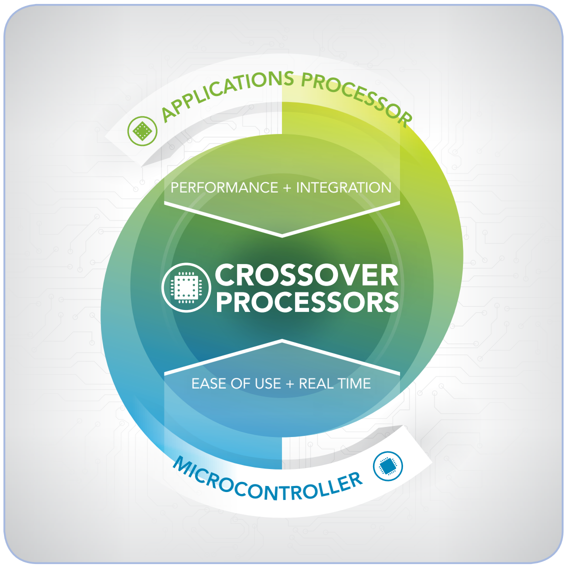


Figure 1: The Crossover Space

Crossover processors are built using applications processors chassis, delivering a high level of integration, high speed peripherals, enhanced security, and engines for enhanced user experience (for example, 2D/3D graphics); but powered by a low-power MCU core running a real-time operating system like RTOS. Crossover processing, therefore, defines a new, much-needed space in the market by helping MCU customers move up to applications processor-level of performance while staying with their current tool chain—and potentially without having to add time, cost, or complexity of Linux (or other higher level operating systems) software development to their product design cycle.

Lowering Cost—Eliminating On-Chip Flash Memory

In an ideal world, embedded processors have the highest performance when the executable code and data is stored in on-chip SRAM with CPU core operations being executed from this storage. Even within the on-chip SRAM, only 'Tightly Coupled Memory' (TCM) provides single cycle access to the core. The number of CPU cycles required increases for any access to memories outside of TCM, with increasing penalty for access from level-2 cache to on-chip flash to external flash. Therefore, only a high density of on-chip SRAM that is configured as TCM provides the highest possible theoretical performance.

In the practical world, however, integrating SRAM is expensive in terms of dollar-per-square-millimeter, especially for MCUs which typically are manufactured in older process technology nodes. Therefore, embedded flash became a desired complement to the on-chip SRAM in MCUs . The embedded flash in MCUs is used to store the executable program as well as valuable user/system data, so that the core has fast access to these items without having to fetch them from an external memory.

But we believe that the technical limitations of on-chip flash, coupled with associated costs, are too great a burden to bear when bridging customer needs between MCUs and applications processors. Embedding flash is expensive and challenging—especially with advanced process technology nodes. The additional processing needed to embed flash significantly increases the cost of silicon. Shedding the burden of on-chip flash, not only reduces the cost, but it also helps enable higher frequency operation for increased processor performance—which in turn lets product designers boost capabilities, increase efficiency and add more features.

The crossover processors are built on applications processors' chassis fully utilizing the scaling benefits of advanced technology nodes (40nm and smaller), which have significantly smaller SRAM bitcells making it cost effective to integrate high densities of SRAM. SRAM is further configured within the crossover design architecture to function as TCM with 'zero-wait' single cycle access to dramatically increase system performance. This key design feature enables the crossover processor's effective performance to be significantly better than any MCU counterpart.

Unifying High Performance, Low Latency, Power Efficiency, and Security

High Performance

Consider a typical MCU with extremely small TCM and cache (on-chip SRAM), but with a high density of embedded flash. During the execution of the application code, the most recent instructions and data are typically stored in TCM/cache for fast subsequent access from the CPU core. However, since the cache density is small, most of the instructions and the data must be stored either in embedded flash or in an external NOR or NAND flash. Therefore, for every access that the CPU core has to make to a non-TCM or non-cache memory (also known as 'cache miss rate'), there is a significant degradation in the effective performance as the CPU must wait for several tens of cycles for the data to arrive. Thus, the higher the cache miss rate, the lower the effective performance of the MCU.

With a correctly architected crossover embedded processor with high density of TCM and cache, however, the effective performance will always be greater than an MCU—even with a cache miss rate as high as 5%. That is, on an average CPU core fails one in twenty times to find required instruction or data in the cache memory, therefore, requiring access to embedded flash or external flash. However, crossover processors with high density of on-chip TCM or cache can have a cache miss rate as low as 1-2% for the most common IoT applications, thus delivering significantly higher net effective performance than what is possible with MCUs.

Low Interrupt Latency

Interrupts play a very important role in embedded systems in coordinating timely response to internal and external hardware events. They particularly play a critical role in real time systems that have user interaction, since the external events triggered by user inputs require immediate response from the CPU with low latency and determinism. Consider a streaming video IoT application or a thermostat with user input. If the system does not respond to user input in real-time, the user experience is severely compromised and the user may deem the product unresponsive. Therefore, interrupt response time is one of the key characteristics of crossover processor applications.

The term interrupt latency refers to the number of clock cycles required for the CPU to respond to an interrupt request from an internal or external event. For most IoT applications with real time response requirements, the interrupt latency becomes a very important barometer for effective performance.

For generic applications processors running high level operating system (non-real time operating system), servicing an interrupt routine is a secondary priority. The CPU typically does not respond to an Interrupt Service Routine (ISR) request until the current instruction execution is completed. This can result in up to hundreds of cycles of delay between an interrupt request and when the core begins to service the request. This is not so in the case of MCUs running real-time operating systems where interrupt response is a priority for the core. Even within the interrupts, priority hierarchy can be assigned to ensure lowest possible interrupt latency for highest priority interrupts.

The crossover processors are built with an MCU core, and therefore, inherit this desired feature of low interrupt latency even though they are built on an applications processors chassis. Crossover processors can have interrupt latency as low as 10-20ns, as compared to latency as long as a microsecond, that is often seen with applications processors. Crossover processors, therefore are most well-suited for IoT applications.

Power Efficiency

Crossover embedded processors are optimized for power efficiency by integrating a DC-DC converter and by effective power gating techniques. In fact, in 40nm process platform that is widely used for the embedded processors, it’s possible to achieve run power as low as 100uA/MHz with crossover processors —less than half that is currently possible with leading MCUs. For many MCUs in the market today which do not have an integrated DC-DC, or when the system does not use an external DC-DC converter or PMIC (to reduce system cost), the current draw can be as high as 300uA/MHz, making them less desirable for battery operated applications.

In MCUs, one of the benefits of embedded flash is the reduction in dynamic power that is achievable as compared to accessing external memory. This is to be expected because external memory access requires the processor to consume more energy as the data must be driven through capacitive I/Os. In crossover processors, however, as described previously, the access to external flash is very limited because of high density on-chip SRAM. Additionally, crossover processors can be built in advanced technology nodes with low-power process platforms and integrated with low-power SRAM, resulting in an overall reduction of dynamic power compared to traditional MCUs.

Furthermore, memory manufacturers currently offer ultra-low power serial NOR flash at very low cost points, which enables lower systems-level power consumption. Power efficiency can even be taken one step further when a crossover embedded processor featuring low-power serial flash is packaged into a SiP due to lower capacitive loading of the I/O pins. If SiP cost can be justified for certain applications (considering the form factor benefits that SiP provides), then solutions that combines crossover processing with additional memory can be quite attractive.

Security

The reliable security of data stored in embedded flash is often cited as a reason for using embedding flash in processor designs. Critical data and sensitive application programs stored on-chip are immediately at-hand and not transferred in and out of the chips through external pins. Anytime data is transferred in and out of the chip, it provides an attack opportunity *(side channel attack)* for hackers to intercept or corrupt the data by monitoring the electrical signals at the data pins. This susceptibility is especially concerning if the data is not encrypted (data transferred 'in the clear').

With crossover embedded processors, however, this is not a concern because they share a key architectural similarity to applications processors. Crossover solutions are built with hardware-accelerated cryptographic blocks. This allows data to be stored in external memory in an encrypted format. When needed, the encrypted data is brought into the chip, where it gets decrypted 'on-the-fly' at the pin itself—meaning there are no wait cycles for decryption. These advanced cryptographic accelerators in crossover processors substantially improve the throughput of encryption/decryption, and thereby obviate the need for on-chip non-volatile memories for security needs.

Even in cases where the hardware acceleration of the cryptography is not desired or viable, software acceleration can be implemented utilizing the high performance of the crossover processor core. This is a definite advantage over traditional MCUs which must rely on silicon implementation of cryptographic blocks to deliver the cryptographic throughput desired for IoT applications.

Looking ahead at the progress made by the industry in developing advanced memories, back-end storage type of non-volatile memories (eg. MRAM, RRAM, etc.) appear to be viable for integration into embedded processors within the next two to three years. Crossover processors are poised to derive maximum benefit from these emerging memories as they can be easily integrated monolithically to provide multiple functionality(that includes replacing on-chip level 2/3 cache) instant-on user experience, on-chip secure data bank, as well as traditional uses of embedded flash.

Relevant Industries and Applications

The purposeful intersection of performance and usability features allow crossover solutions to target a range of specific applications that were previously underserved in some capacities by both applications processor and MCU products. Some applications where crossover processors can be of tremendous difference, but not limited to, are:

- **Audio Subsystem** designs for high-end, consumer audio devices, including specialty equipment such as guitar pedals.
- **General Embedded** designs for mass-market applications such as metering, medical equipment, vending machines, and IoT gateways.
- **Human Machine Interface (HMI)** graphics capabilities for home and building automation (e.g., HVAC climate control, security, lighting control panels), industrial computing designs (e.g., EBS, PLCs, factory automation, test and measurement, M2M, HMI control, assembly line robotics), and consumer products (e.g, smart appliances, cameras, LCDs and other high-end displays).
- **Motor Control and Power Conversion** for professional appliances like 3D printers, thermal printers, and drones—as well as consumer products, such as robotic vacuum cleaners.

The i.MX RT Crossover Processor

Building upon our decades of leadership experience in providing both MCUs and applications processors for the consumer, industrial, and the IoT markets, NXP has developed a new series of embedded processors that fully embodies all the features of crossover described in the sections above. The new i.MX RT crossover series processors are based on ARM® Cortex®-M7 core, but architected from the versatile i.MX 6ULL applications processor. The i.MX RT processors offer the first solutions in the market that truly bridge the gap between MCUs and applications processors. They bring applications processor-level performance into the MCU world in support of real-time, performance-intensive processing functions such as camera and display capabilities.

High-Performance, Real-Time Processing

The i.MX RT series can operate at a core speed of up to 600 MHz (as opposed to existing MCUs with a maximum speed of 400 MHz). This is the highest performing Cortex-M7 solution, delivering 3036.5 CoreMark® (~ 5 CM/MHz) 1284 DMIPS (> 2 DMIPS/MHz).

The i.MX RT series also boasts ultra-fast real-time responsiveness thanks to a high density of 512KB TCM SRAM. The interrupt latency is as low as 20 nanoseconds, and is the shortest interrupt latency of any ARM Cortex-based product in the world thanks to a combination of high performance and the Cortex-M7 core. All told, i.MX RT crossover processors outperform competitors by a factor of 2x.

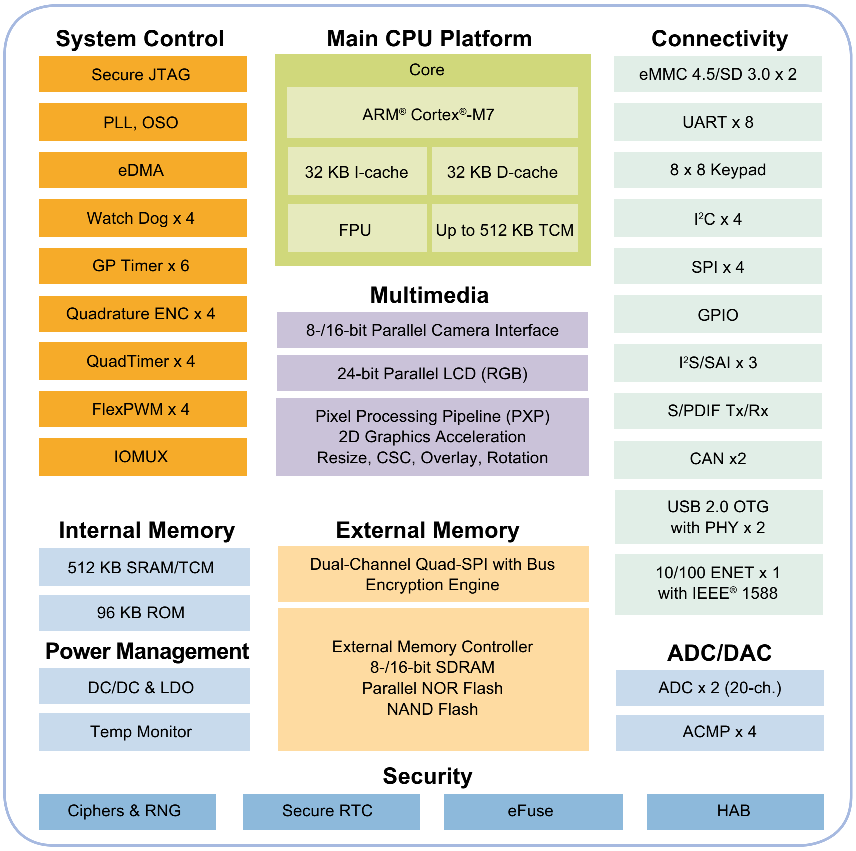
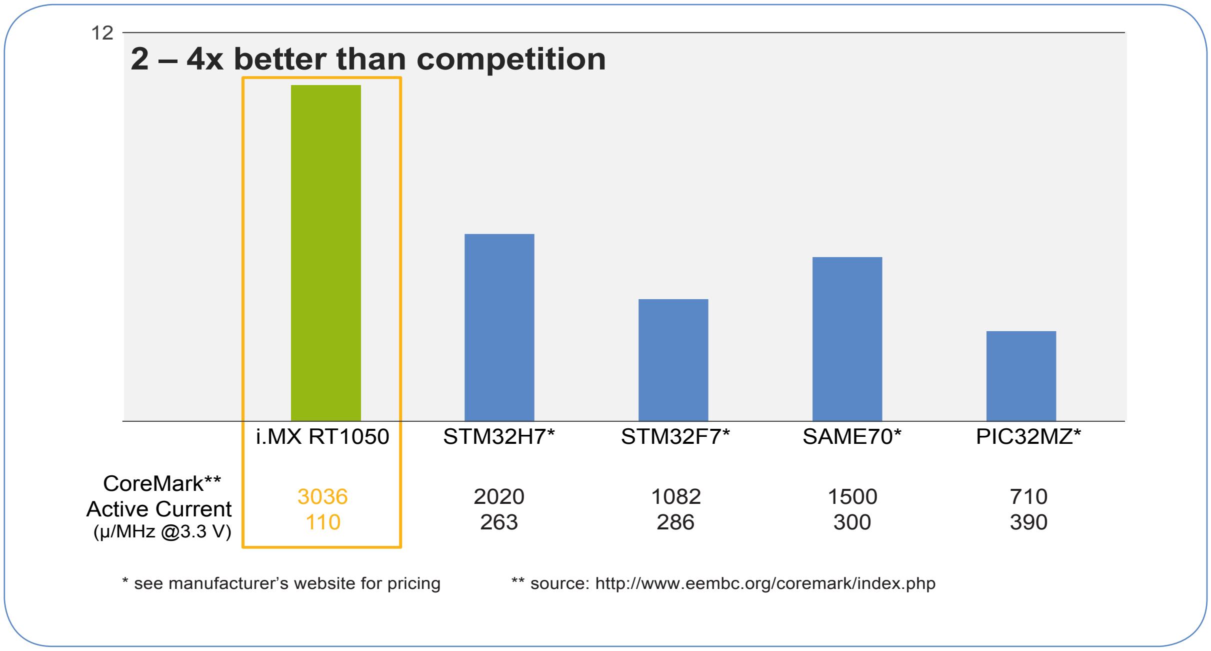


Figure 2: i.MX RT1050 Block Diagram

While the i.MX RT series incorporates applications processor-quality performance, it also provides critical usability features from the MCU-based product design space— ease of use, low cost, and compatibility with existing real-time software infrastructure and compatibility with existing tool chains.

For comparison, in figure 3 shows the CoreMark score per MHz (CM/MHz) of i.MX RT1050 compared to other leading MCUs in the market. The i.MX RT1050 deliver 2-4x higher CM/MHz than the competitors while consuming less than half of the current in full operation mode.

Comparing Leading MCUs for CoreMark per mW (active)



	i.MX RT1050	STM32H7*	STM32F7*	SAME70*	PIC32MZ*
CoreMark** Active Current (μ/MHz @3.3 V)	3036 110	2020 263	1082 286	1500 300	710 390

* see manufacturer's website for pricing

** source: http://www.eembc.org/coremark/index.php

Figure 3: Comparing Leading MCUs for CoreMark per mW (active)

Rich Integration

The i.MX RT series supports product designs that include advanced multimedia for GUI and enhanced human machine interface (HMI) experience. Its features include advanced 2D graphics acceleration engine, LCD display controllers, camera sensor interfaces, and audio interfaces for high-performance, multi-channel audio streaming.

Just as described above, i.MX RT series of crossover processors support greater design flexibility through extensive external memory interface options, including NAND, eMMC, QuadSPI NOR Flash and Parallel NOR Flash. By having high speed interfaces between external memory, combined with on-the-fly decryption enables secure external data storage, the need for embedded flash is eliminated.

In terms of connectivity, i.MX RT crossover processors offer support for both wired (Ethernet, USB, etc.) and wireless standards such as Wi-Fi®, Bluetooth®, BLE, Zigbee®, Thread™, and other options; making thousands of applications for the connected world possible.

Easy to Use

The i.MX RT series is designed to make it easy for MCU-based developers to use these new class of crossover processors without having to make significant investment into developing new SW enablement or investing to learn higher level operating systems, like Linux or Android. Current MCU customers can significantly increase performance capabilities using i.MX RT series while leveraging their current toolchain (e.g., MCUXpresso, IAR and Keil).

i.MX RT also enables rapid prototyping and development with NXP FreeRTOS, SDK, ARM® mbedTM and the global ARM ecosystem which provide software libraries and online tools and support. Development can be accelerated even further by using the low-cost evaluation kit (EVK) compatible with Arduino™ hardware shields.

In addition, i.MX RT solutions incorporate a single-voltage input to simplify power circuit design.

Low Bill of Materials (BOM) Cost

The i.MX RT series is also designed to provide the lowest cost for a Cortex-M7 based embedded system, by combining the highest functionality with industry-leading performance and power efficiency. It uses an affordable 10x10 BGA package with .65mm pitch enabling low cost four-layer, and even two-layer, PCB design.

Building on a Cortex-M7 core also eliminates the need for expensive infrastructure. For example, a fully integrated power management functionality with a DC-DC converter erases the need for an external power supply. 512 KB of on-chip SRAM means that external DRAM can be eliminated to further reduce total bill-of-materials cost. Further, eliminating the on-chip flash not only allows higher frequency of operation, but it also brings down the cost of the i.MX RT to a fraction of the cost of the competing MCUs. At the systems level, the total BOM cost is reduced by using standard external NOR flash for data storage. Programming off-chip flash is faster and cheaper compared to integrated on-chip flash. The programming of standard off-chip flash is faster due to the simplicity of direct programming – for example, programming an external 2MB serial flash can be as much as 60% faster compared to embedded flash MCUs. The use of external flash also simplifies and lowers the cost of set-up and handling costs at the programming facilities due to lower pin count and homogeneity of external flash offerings, as compared to the handling challenges of more complex high-pin count MCUs & vendor variability of MCUs.

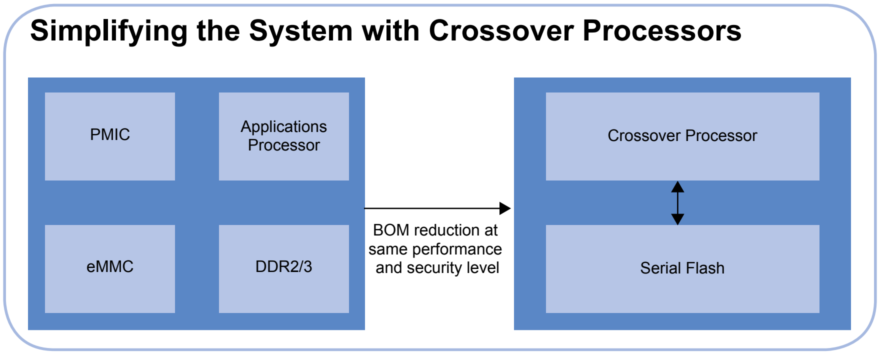
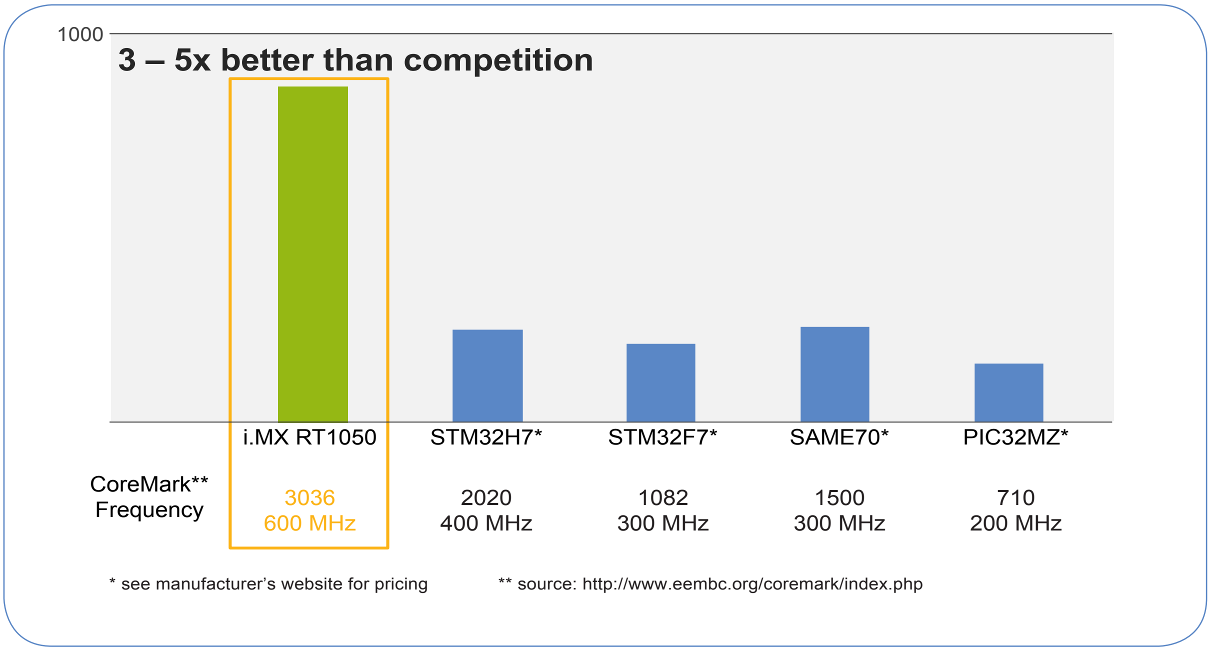

Simplifying the System with Crossover Processors
PMIC
Applications Processor
eMMC
DDR2/3
BOM reduction at same performance and security level
Crossover Processor
Serial Flash

Figure 4: BOM Diagram

The cost effectiveness of i.MX RT is evident when we compare CoreMark per $ for i.MX RT1050 against other leading MCU suppliers (figure 5). Even after including the cost of external flash (8-16MB) to the cost of i.MX RT1050, the CM/$ figure-of-merit for i.MX RT1050 is 3-5x better than the competitors.

Comparing Leading MCUs for CoreMark per $



* see manufacturer's website for pricing

** source: http://www.eembc.org/coremark/index.php

Figure 5: Comparing Leading MCUs for CoreMark per $

Software Support

Current MCU-based developers can continue using their existing base of software for i.MX RT-based products. Developers coming from the applications processor world, however, will have some questions regarding the transition into building with a processing solution that do not feature Linux code.

i.MX RT processors do run RTOS software, as opposed to the Linux-based software required by applications processors. This will require applications processor-based developers to learn more about building systems with real-time operating systems, but the effort will be worthwhile as it opens them to a vast market space for products that require high performance, enhanced user experience, but at a lower cost.

Conclusion

The increasing demand for efficient, high-performance embedded processing that can deliver an enhanced user experience and high data processing power without increasing costs and power consumption has created a gap within the embedded processor space. To address this gap, we have introduced a new class of embedded processors that deliver performance levels and security capabilities that are typical of applications processors, but with the ease of use, real-time operation, and low-power capabilities of MCUs. These crossover processors are our answer to a growing market need, architected to provide highest performance, cost-effective embedded solution for the growing smarter, secure connected world.

Delivering Processor Capabilities at MCU Prices

i.MX RT Crossover MCUs

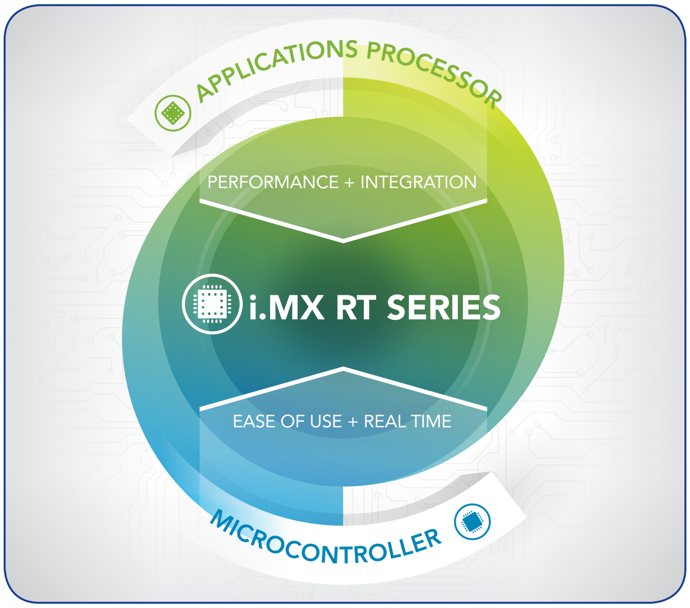
2X PERFORMANCE 2X BATTERY LIFE 2X INTEGRATION 2X FASTER DEVELOPMENT ½X THE COST

USHERING IN THE GHZ MCU ERA

NXP's crossover processors and MCUs marry the simplicity of MCUs with the complexity of applications processors into the ultimate hybrid device to address the growing consumer demand for enhanced user experiences in smart, secure, high-performance products. Included in this new class of product is the i.MX RT series of crossover MCUs that combines unprecedented performance - offering the industry's only GHz MCU - with reliability and high levels of integration and security to propel industrial, IoT and automotive applications.

PORTFOLIO HIGHLIGHTS

- Highest performing Arm® Cortex®-M based device
  - Up to 6468 total CoreMarks with Cortex-M7 @ 1GHz + Cortex-M4 @ 400 MHz
- Real-time, low-latency response
  - Up to 2 MB of SRAM
    - 512 KB of TCM with ECC for Cortex-M7
    - 256 KB of TCM with ECC for Cortex-M4
  - Fast real-time response with latency as low as 12 ns
- Low-power operation
  - Industry's lowest dynamic power with integrated DC-DC converter
  - Low-power run modes at 24 MHz
- Highly integrated
  - Advanced multimedia for GUI and enhanced HMI



- Extensive memory interface options
  - Quad/Octal SPI and Hyper Flash/RAM, SDRAM, NAND Flash, NOR Flash, SD/eMMC
- Security
  - Hardware protected keys for secure boot
  - AES engine for data encryption
  - On-The-Fly Decryption for execute-in-place (XIP) from Quad/Octal SPI / Hyper Flash

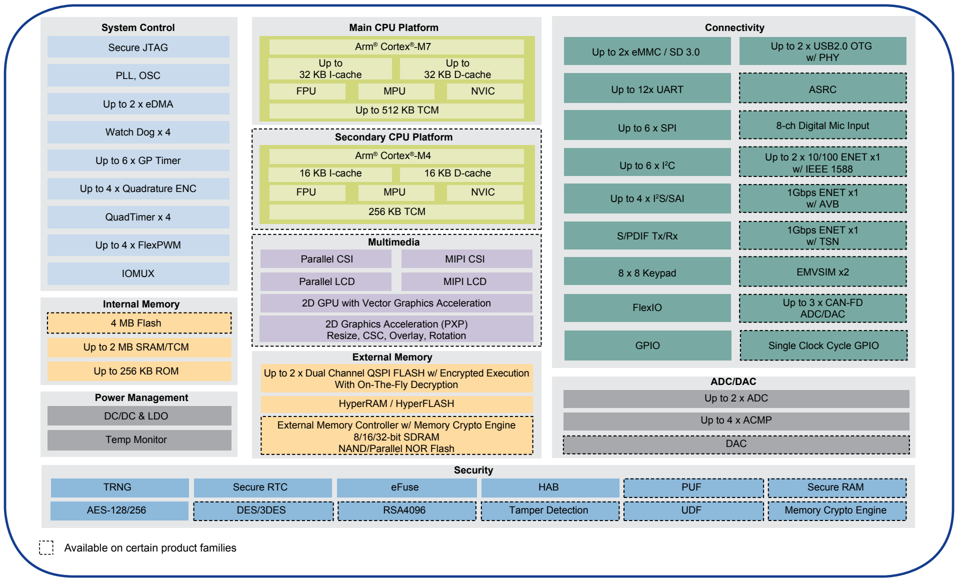
i.MX RT CROSSOVER MCUs | PORTFOLIO FEATURES

Feature	i.MX RT1010	i.MX RT1015	i.MX RT1020	i.MX RT1050	i.MX RT1060/RT1064	i.MX RT1170
Core/Speed	Cortex-M7 @ 500 MHz	Cortex-M7 @ 500 MHz	Cortex-M7 @ 500 MHz	Cortex-M7 @ 600 MHz	Cortex-M7 @ 600 MHz	Cortex-M7 @ 1GHz, Cortex-M4@ 400 MHz
Cache	16 KB-I, 8 KB-D	16 KB-I, 16 KB-D	16 KB-I, 16 KB-D	32 KB-I, 32 KB-D	32 KB-I, 32 KB-D	32 KB-I, 32 KB-D
TCM	Up to 128 KB	Up to 128 KB	Up to 256 KB	Up to 512 KB	Up to 512 KB	Up to 512 KB
On-chip RAM	128 KB	128 KB	256 KB	512 KB	1 MB	2 MB
On-chip Flash	-	-	-	-	Up to 4 MB	-
External Memory	-	-	8-/16-bit interface for SDRAM, SRAM, NOR, NAND	8-/16-bit interface for SDRAM, SRAM, NOR, NAND	8-/16-bit interface for SDRAM, SRAM, NOR, NAND	8-/16-/32-bit interface for SDRAM, SRAM, NOR, NAND
Quad/Octal SPI/ HyperBus	Dual-channel/8-bit	Dual-channel/8-bit	Dual-channel/8-bit	Dual-channel/8-bit	Up to 2 x Dual-channel/8-bit	1 x Dual-channel/8-bit 1 x Dual-channel/16-bit
SDIO	-	-	SD3.0/eMMC4.5 x 2	SD3.0/eMMC4.5 x 2	SD3.0/eMMC4.5 x 2	SD3.0/eMMC5.0 x 2
Ethernet	-	-	10/100 Mbit/s x 1 w/ IEEE 1588	10/100 Mbit/s x 1 w/ IEEE 1588	10/100 Mbit/s x 2 w/ IEEE 1588	1Gbit/s w/ AVB + 1Gbit/s w/ TSN + 10/100 Mbit/s w/ IEEE 1588
USB with PHY	OTG, HS/FS x 1	OTG, HS/FS x 1	OTG, HS/FS x 1	OTG, HS/FS x 2	OTG, HS/FS x 2	OTG, HS/FS x 2
CAN	-	-	FlexCAN x 2	FlexCAN x 2	FlexCAN x 2 + CANFD x 1	CAN-FD x 3
Graphics	-	-	-	PxP for 2D acceleration	PxP for 2D acceleration	PxP for 2D acceleration, 2D GPU with vector graphics acceleration
CSI	-	-	-	8-/10-/16-bit parallel	8-/10-/16-bit parallel	8-/10-/16- bit parallel, 2-lane MIPI CSI
LCD	-	-	-	8-/16-/18-/24-bit parallel	8-/16-/18-/24-bit parallel	8-/16-/18-/24-bit parallel, 2-lane MIPI DSI
Security	TRNG, AES-128, SHA, Secure Boot	TRNG, AES-128, SHA, Secure Boot	TRNG, AES-128, SHA, Secure Boot	TRNG, AES-128, SHA, Secure Boot	TRNG, AES-128, SHA, Secure Boot	TRNG,AES-128/256, SHA, Secure Boot, RSA4096, DES/3DES, Tamper Detection PUF, UDF, Secure RAM
UART/SPI/I^{2}C/FlexIO	4/2/2/1	4/2/2/1	8/4/4/1	8/4/4/2	8/4/4/3	12/6/6/2
I^{2}S/SPDIF	2/1	3/1	3/1	3/1	3/1	4/1
ADC	1M sample/s x 1	1M sample/s x 1	1M sample/s x 2	1M sample/s x 2	1M sample/s x 2	2M sample/s x 2
ACMP/DAC	-	-	4/-	4/-	4/-	4/1
FlexPWM/Quad Timer/ Quad ENC	1/0/0	1/1/1	2/2/2	4/4/4	4/4/4	4/4/4
GP Timer / WDOG	3/4	6/4	6/4	6/4	6/4	6/4
Package	80 LQFP	100 LQFP	100 LQFP, 144 LQFP	196 BGA	196 BGA	289 BGA
Temperature (Tj)	Consumer: 0 to 95 °C Industrial: -40 to 105 °C	Consumer: 0 to 95 °C Industrial: -40 to 105 °C	Consumer: 0 to 95 °C Industrial: -40 to 105 °C	Consumer: 0 to 95 °C Industrial: -40 to 105 °C	Consumer: 0 to 95 °C Industrial: -40 to 105 °C	Consumer: 0 to 95 °C Industrial: -40 to 105 °C Automotive: -40 to 125 °C

MEMORY EXPANSION WITH i.MX RT

i.MX RT crossover MCUs shed the burden of on-chip flash, which not only reduces the cost, but it also enables higher frequency operation for increased performance—which in turn lets you boost capabilities, increase efficiency and add more features. The i.MX RT FlexSPI interface provides memory expansion for external memories such as serial Flash / PSRAM, Quad or Octal data lines. This offers maximum design flexibility, while still ensuring a high-level of performance and security. The i.MX RT series integrates high densities of SRAM, which is further configured within the crossover design architecture to function as TCM with 'zero-wait' single cycle access to dramatically increase system performance. This key design feature enables the crossover processor's effective performance to be significantly better than any traditional MCU counterpart.

i.MX RT CROSSOVER MCUs | PORTFOLIO BLOCK DIAGRAM



TARGET APPLICATIONS

- Audio Subsystem
- Consumer Products
- Home and Building Automation
- Industrial Computing Designs
- Motor Control and Power Conversion
- Automotive

ACHIEVING END-TO-END SECURITY

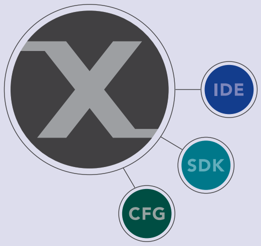
Secure development with the i.MX RT leverages years of experience gained from its applications processor lineage. The ROM firmware on the devices, as well as the tools used in the development and manufacturing processes, have been heavily used and tested. With the i.MX RT and its associated software and tools for secure boot, the foundation for meeting today's security requirements can be achieved.

MAXIMUM FLEXIBILITY WITH FlexIO

FlexIO is a highly configurable module providing a wide range of functionality including emulation of a variety of communication protocols such as UART, I²C, SPI, I²S. This means that you have the ultimate flexibility in your design to add more of the peripherals you need. Additionally, the FlexIO module consists of a flexible 16-bit timer with support for a variety of trigger, reset, enable and disable conditions.

i.MX RT CROSSOVER MCUs | SOFTWARE AND TOOLS

NXP's **MCUXpresso software and tools** offer comprehensive development solutions designed to optimize, ease and accelerate embedded system development of applications based on Cortex-M core devices, including Kinetis and LPC microcontrollers, and i.MX RT crossover MCUs.



NXP eIQ™ Machine Learning Software Development Environment

The NXP eIQ ("edge intelligence") ML software environment provides the key ingredients to do inference with neural network (NN) artificial intelligence (AI) models on embedded systems and deploy various ML algorithms on NXP microprocessors and microcontrollers for edge nodes. It includes inference engines, NN compilers, libraries, and hardware abstraction layers that support Google TensorFlow Lite, Arm NN, Arm® CMSIS-NN, and OpenCV.

GET STARTED NOW

Take advantage of the robust enablement to reduce development effort and speed time-to-market with a comprehensive offering of software and development tools.

The i.MX RT evaluation kits (EVKs) help you take your design to the next level by reducing complexity and accelerating time to market. Additionally, enjoy the ability to expand upon this feature-rich EVK with compatible Arduino™ hardware shields.

Toolchains

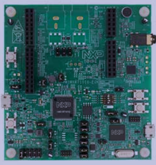
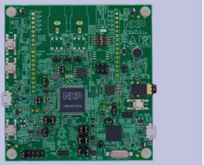
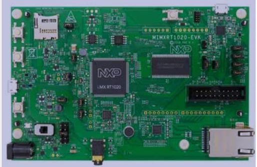
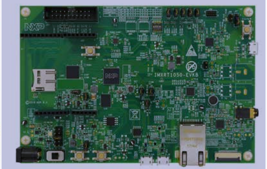
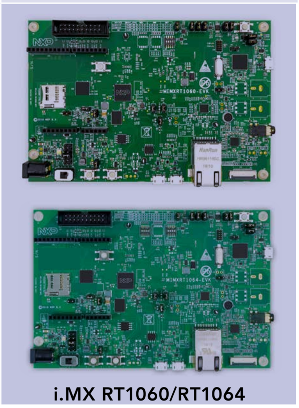
- MCUXpresso software and tools
- IAR® Embedded Workbench® IDE
- Keil® IDE

Software

- MCUXpresso SDK with Amazon FreeRTOS™
- Zephyr® Operating System
- Arm® Mbed™ and the global Arm ecosystem

i.MX RT1010, i.MX RT1015 and i.MX RT1020 EVKs are two-layer through-hole PCBs enabled with a six-axis eCompass sensor, multiple audio features and debug options.

i.MX RT1050, i.MX RT1060 and i.MX RT1064 EVKs are four-layer through-hole PCBs that also offer additional features, including camera and LCD support.

EVK	i.MX RT1010	i.MX RT1015	i.MX RT1020	i.MX RT1050	i.MX RT1060/RT1064
Processor	• MIMXRT1011DAE5A	• MIMXRT1015DAF5A	• MIMXRT1021DAG5A	• MIMXRT1052DVL6B	• MIMXRT1062DVL6A/ MIMXRT1064DVL6A
Memory	• 128 Mb QSPI Flash	• 128 Mb QSPI Flash	• 256 Mb SDRAM memory • 64 Mb QSPI Flash • TF socket for SD card	• 256 Mb SDRAM memory • 512 Mb Hyper Flash • 64 Mb QSPI Flash • TF socket for SD card	• 256 Mb SDRAM memory • 512 Mb Hyper Flash • 64 Mb QSPI Flash • TF socket for SD card
Display	N/A	N/A	N/A	• Parallel LCD connector • Camera connector	• Parallel LCD connector • Camera Sensor Module
Audio	• Audio codec • 4-pole audio headphone jack • External speaker connection • Microphone	• Audio codec • 4-pole audio headphone jack • External speaker connection • Microphone	• Audio codec • 4-pole audio headphone jack • External speaker connection • Microphone	• Audio codec • 4-pole audio headphone jack • External speaker connection • Microphone • SPDIF connector	• Audio codec • 4-pole audio headphone jack • External speaker connection • Microphone • SPDIF connector
Connectivity	• Micro USB OTG connector • Arduino® interface	• Micro USB OTG connector • Arduino® interface	• Micro USB host connector • Micro USB OTG connector • Ethernet (10/100T) connector • CAN transceivers • Arduino® interface	• Micro USB host connector • Micro USB OTG connector • Ethernet (10/100T) connector • CAN transceivers • Arduino® interface	• Micro USB host connector • Micro USB OTG connector • Ethernet (10/100T) connector • CAN transceivers • Arduino® interface
Debug	• JTAG connector • Onboard DAP-link debugger	• JTAG connector • Onboard DAP-link debugger	• JTAG connector • Onboard DAP-link debugger	• JTAG connector • Onboard DAP-link debugger	• JTAG connector • Onboard DAP-link debugger
Sensor	• 6-axis eCompass (3-axis magnetometer, 3-axis accelerometer) sensor FXOS8700CQ	• 6-axis eCompass (3-axis magnetometer, 3-axis accelerometer) sensor FXOS8700CQ	• 6-axis eCompass (3-axis magnetometer, 3-axis accelerometer) sensor FXOS8700CQ	• 6-axis eCompass (3-axis magnetometer, 3-axis accelerometer) sensor FXOS8700CQ	• 6-axis eCompass (3-axis magnetometer, 3-axis accelerometer) sensor FXOS8700CQ
Part Number	MIMXRT1010-EVK	MIMXRT1015-EVK	MIMXRT1020-EVK	IMXRT1050-EVKB	MIMXRT1060-EVK/ MIMXRT1064-EVK
Camera Sensor	N/A	N/A	N/A	N/A	MT9M114 image sensor (included)
Display	N/A	N/A	N/A	RK043FN02H-CT 4.3" (purchased separately)	RK043FN02H-CT 4.3" (purchased separately)
	 i.MX RT1010	 i.MX RT1015	 i.MX RT1020	 i.MX RT1050	 i.MX RT1060/RT1064

- **For further technical details (schematic/PCB layout, etc), enquiry or support, please contact NXP.SA@wpi-group.com.**
- **Welcome to browse WPI Group website https://www.wpgholdings.com/wpig/branches/en for more information.**