

White Paper

Introducing RXv3 Core: Superior Performance with Excellent Power Efficiency

October 2019

Introduction

Microcontrollers (MCUs) used in advanced control systems must constantly evolve to meet increasing requirements. Their development takes place with three main goals:

- To increase CPU core performance
- To increase CPU resources (advance peripherals, Flash program memory and RAM data memory)
- To reduce power consumption

In a real-time control embedded system, there are many extensive resources needed to manage multiple tasks. One such application is in a motor control application where the CPU core performance is a key requirement to manage all the resources.

There are many families of MCUs in the market for use in systems that require high computing power. One of the most common ways to benchmark the computing power is to use the EEMBC's Coremark® as a benchmark to measure the performance of MCUs and microprocessors (MPUs) used in embedded systems. The result is given in the form of performance per MHz of the clock frequency.

The latest version of the RX series core, RXv3, marks 5.82 Coremark®/MHz. It exceeds the ARM Cortex-M7 core (5.05 Coremark®/MHz).

An RX MCU core performance comparison is presented in Figure 1.

An overview of basic RX core's properties are presented in Figure 2.

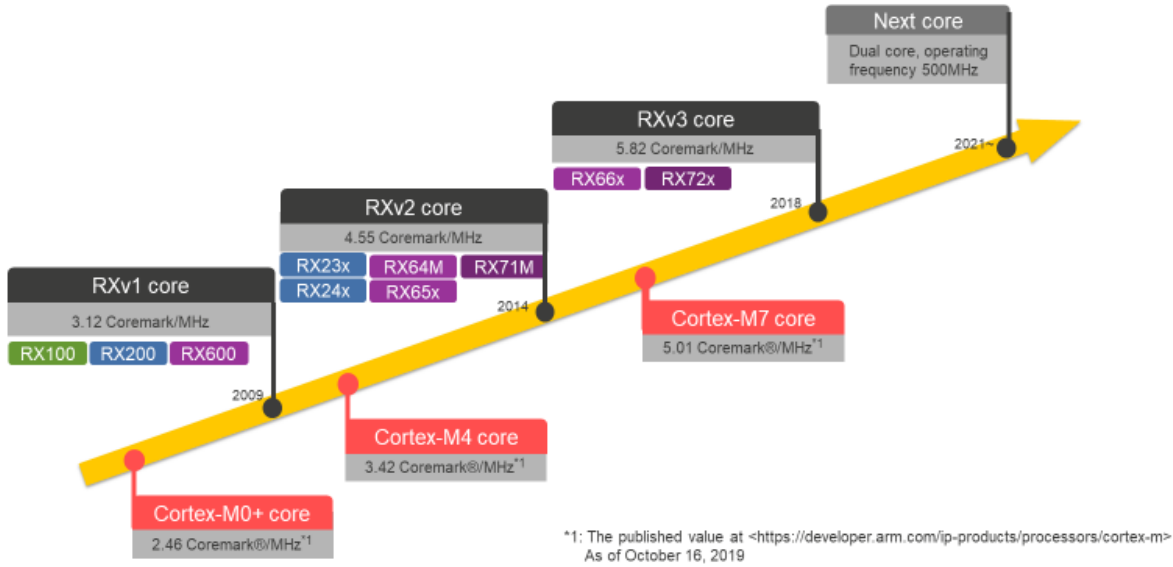


Figure 1. Comparison of RX and ARM Coremark®/MHz core's performance

| RX core | RXv1 | RXv2 | RXv3 |
|----------------------------------|---|--|--|
| Architecture | 32bit CISC、Harvard architecture | | |
| General purpose registers | 32bit x 16ch | | |
| Compatibility | RXv1 | Downward compatible with RXv1 | Downward compatible with RXv1/RXv2 |
| Instruction set | 90 instructions | 109 instructions (90 RXv1 instructions + 19 instructions) | 113 instructions (109 RXv2 instructions + 4 instructions) *2 out of 113 instructions are for register bank save function. |
| Pipeline | 5-stage | Improved 5-stage pipeline Improved IPC through enhanced pipeline (enhanced performance through parallel execution of memory access and operations) | Improved 5-stage pipeline Improved IPC through enhanced pipeline (enhanced performance through improved combination of simultaneously executable instructions) |
| DSP function instructions | Single-cycle MAC instructions(16-bit), Accumulator x 1 | Single-cycle MAC instructions (32bits x 32bits + 72bits) Accumulator x 2 | |
| FPU | Supports IEEE754 compliant data types and exceptions Pipeline processing | | |

| | | | |
|--------------------|------------------------|------------------------|---|
| Performance | Up to 3.12CoreMark/MHz | Up to 4.55CoreMark/MHz | 5.82 CoreMark/MHz |
| Others | - | - | Register bank save function (optional) Double precision floating-point processing instructions (optional) *Availability of optional functions depends on product specifications |

Figure 2. Basic features of Renesas' RX core

Key Benefits of RXv3 Core

The RXv3 core realizes:

- High performance 5.82 Coremark®/MHz
- Small memory footprint with compact instruction set
- Double-precision FPU capabilities
- Fast interrupt response time

The performance of the MCU is critical in high-speed embedded system control, i.e. electric motor control system, robotics, etc. This allows sophisticated software to incorporate an RTOS to be run and executed by the microcontroller.

Currently, three families of MCUs are available with the RXv3 core:

- RX66T clocked up to 160MHz optimized to control the power of electric motors
- RX72T clocked up to 200MHz optimized to control the power of electric motors in very demanding applications, i.e. robot drives
- RX72M clocked up to 240MHz optimized to control the system and industrial network communications

Instruction Set Architecture

CPU architecture has a decisive impact on core performance and code size. There are many successful implementations of cores in both CISC and RISC architectures; however, the most effective are those which combine advantages of both solutions. The advantage of CISC is the compact code and smaller Flash memory (ROM) needed. The RISC code is performed faster and allows pipelining.

The RX instruction set has a compact architecture with carefully selected instructions to reduce it to the number similar to a RISC-based architecture. It uses a variable length instruction set between 1 ~ 8 byte(s). As a result, it helps to achieve higher performance, code density and lower power consumption. Most of the common instructions used optimized addressing mode with short Op Code.

Figure 3 illustrates a code size analysis of the RX and a RISC-based MCU with three different types of applications. The implementation delivers up to 46% reduction in static code size and up to 30% reduction in dynamic code size relative to RISC architectures. Small static code size makes a significant contribution in decreasing ROM size, and by extension, costs. Small dynamic code size delivers low power consumption.

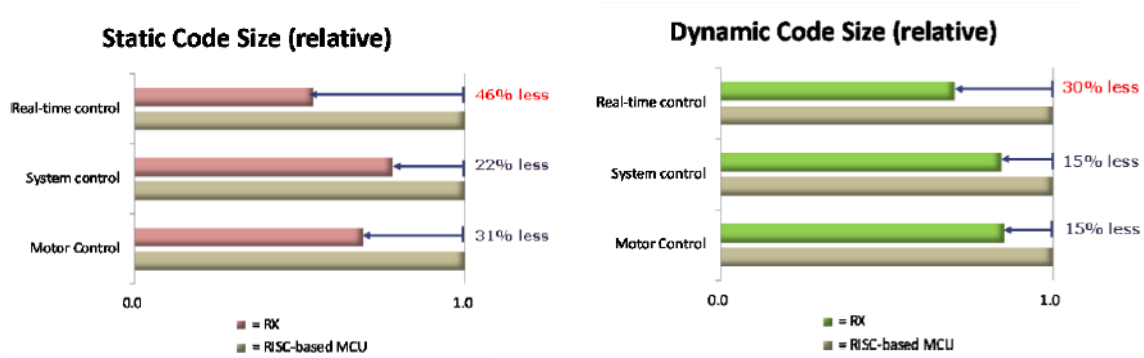


Figure 3. Code size analysis of the RX and a RISC-based MCU

| Arithmetic/Logic | | | Data Transfer | | PUSHC | SAVE* | DSP | | | Data Trans. (DP-FPU) | |
|------------------|-------|------|---------------------|--------|--------|---------|---------------------|---------|---------|----------------------|--------|
| ABS | MAX | RORC | BFMOV | MOVU | PUSHM | SCCnd | EMACA | MSBLO | MVTACGU | DMOV* | MVFDc* |
| ADC | MIN | ROTL | BFMOVZ | POP | REVL | STNZ | EMSBA | MULHI | MVTACHI | DPOPM* | MVFDR* |
| ADD | MUL | ROTR | MOV | PUSH | REVW | STZ | EMULA | MULLH | MVTACLO | DPUSHM* | MVTDC* |
| AND | NEG | SAT | MOVCO | POPC | RSTR* | XCHG | MACHI | MULLO | RACL | Floating-point (DP) | |
| CMP | NOP | SATR | MOVLI | POPM | Branch | Strings | MACLO | MVFACHI | RDACL | DABS* | DSUB* |
| DIV | NOT | SBB | System manipulation | | BCnd | SCMPU | MACLO | MVFACHI | RDACL | DADD* | DTOF* |
| DIVU | OR | SHAR | BRK | MVTC | BRA | SMOVB | MSBHI | MVFACLO | RDACW | DCMPcm* | DTOI* |
| EMUL | RMPA | SHLL | CLRPSW | RTE | BSR | SMOVF | MSBLH | MVFACMI | FTOI | DDIV* | DTOU* |
| EMULU | ROLC | SHLR | INT | RTFI | JMP | SMOVU | Floating-point (SP) | | FTOU | DMUL* | FTOD* |
| SUB | TST | XOR | MVTIPL | SETPSW | JSR | SSTR | FADD | FMUL | ITOF | DNEG* | ITOD* |
| Bit manipulation | | | MVFC | WAIT | RTS | SUNTIL | FCMP | FSUB | ROUND | DROUND* | UTOD* |
| BCLR | BMCnd | BNOT | BSET | BTST | RTSD | SWHILE | FDIV | FSQRT | UTOF | DSQRT* | |

 Enhancements in RXv2
 Enhancements in RXv3
 * option

Figure 4. RX instruction set

In the process of developing the RX core, new instructions were added and some of them were improved. Figure 4 shows a list of instructions for RX. The RXv2 instruction enhancement mainly consisted of adding DSP and single-precision floating point instructions. In the RXv3 instruction enhancement, two instructions used to maintain the context in the interrupt handling procedures (RSTR, SAVE), and double-precision floating point instructions have been added.

Core Architecture

The pipelines of RXv1, RXv2 and RXv3 consist of 5 stages (Figure 5). These pipelines use a Harvard architecture, it allows instruction memory access and data memory access in the same clock cycle. Also,

these pipelines use out-of-order completion technique for memory data load instructions to get higher pipeline usage and better performance.

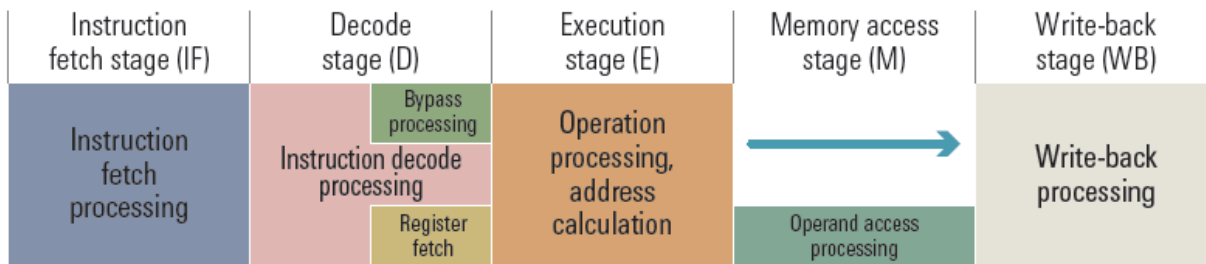


Figure 5. 5 stage pipeline

In the RXv2 core, key differences from the RXv1 are dual issue pipeline structure and pipelined FPU (single precision). Dual issue core can increase a throughput of Instruction Per Cycle (IPC). The new FPU unit adopts pipeline processing to boost throughput and shortens the latency of FPU executions.

The RXv3 core inherits the RXv2 features and has new features, performance improvements, double-precision FPU, and a register save bank function.

Performance improvements are realized by increasing a combination of dual issuable instructions. The effect of this improvement mainly appears around conditional branch instructions, which are typically used in "if-else" statements and "loop" statements in programs. Up to about 20% performance improvement in terms of cycles are achieved with various programs. In some cases, more improvement is achieved due to compiler optimizations for the RXv3 pipeline.

Double-precision FPU accelerates double precision floating point operation processing time by 10 times or more. Double-precision FPU makes it easy to port high precision control models on Model-Based Development (MBD) to MCU.

The RXv3 core is designed not only for performance, but also for power efficiency. The first new RXv3 MCUs achieve 44.8 Coremark®/mA with energy-saving cache design that reduces both access time and power consumption during on-chip flash memory reads, such as instruction fetch.

Improving Interrupt Response Time

Embedded systems that work in real time require a quick response to events. Such events may be signals from the control system of electric motors.

In the RXv3 core, an optional dedicated memory called: register save bank has been implemented for save register banks intended to store register contents overwritten by interrupt handlers. Interrupt response times are shortened by using the register save bank as shown in Figure 6. Using this function, not only interrupt response time, but also total interrupt handling time, are reduced. The interrupt service routine can use the SAVE instruction to save general-purpose registers and accumulators in one clock cycle. The RSTR instruction restores saved registers. Many bank registers allow the context for nested interrupts.

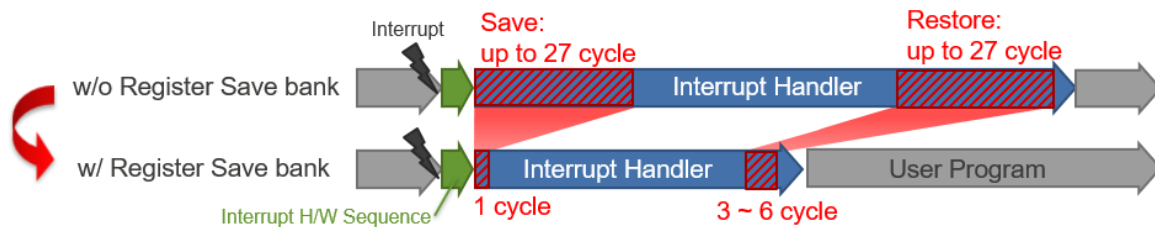


Figure 6. Interrupt response time improvement

Register save bank can be used for not only interrupt handlers, but also RTOS context switching. RTOS context switch time is up to 20% faster with the register bank save function.

Flash Memory

The code and operands of each instruction are fetched from the Flash memory. The accessing speed for flash memory is very crucial, not just the MCU core speed. For a memory access with a fast core, each memory read cycle must be extended by adding additional wait state cycles. Renesas implements RX Flash memory in 40nm MONOS technology. This allows very fast memory access up to 120Mhz without a need to insert an additional wait state.

Conclusion

Modern, advanced control systems require MCUs with increasing efficiency and capabilities. The Renesas RXv3 core achieved industry best-in-class performance with very low power consumption. Based on the EEMBC Coremark® benchmark performance, it has a score of 5.82 Coremark®/MHz, making it one of the fastest MCU core in the world today. This is made possible by the optimization of the instructions set, DSP and FPU units, the use of pipeline, and the fast memory access from a MONOS flash technology. This optimization allows for fast and efficient computation in today's real time control

applications. The RX66T, RX72T and RX72M are three MCUs using the latest RXv3 core and are already available in the market. There will be more devices coming to the market in the near future.

Learn More

[For more information about RXv3 core >](#)

[For more information about RX66T >](#)

[For more information about RX72T >](#)

[For more information about RX72M >](#)

© 2019 Renesas Electronics Corporation or its affiliated companies (Renesas). All rights reserved. All trademarks and trade names are those of their respective owners. Renesas believes the information herein was accurate when given but assumes no risk as to its quality or use. All information is provided as-is without warranties of any kind, whether express, implied, statutory, or arising from course of dealing, usage, or trade practice, including without limitation as to merchantability, fitness for a particular purpose, or non-infringement. Renesas shall not be liable for any direct, indirect, special, consequential, incidental, or other damages whatsoever, arising from use of or reliance on the information herein, even if advised of the possibility of such damages. Renesas reserves the right, without notice, to discontinue products or make changes to the design or specifications of its products or other information herein. All contents are protected by U.S. and international copyright laws. Except as specifically permitted herein, no portion of this material may be reproduced in any form, or by any means, without prior written permission from Renesas. Visitors or users are not permitted to modify, distribute, publish, transmit or create derivative works of any of this material for any public or commercial purposes.