

The Overall Leader, Combining World-Class Features in Three Areas — RX Family



Renesas Technology Corp.

Contents



1. Concept of RX Family
2. Features of RX CPU
 - 2-1 Improved Performance
 - 2-2 Improved Code Efficiency
 - 2-3 Enhanced Flexibility
3. RX Family Product Evolution
 - 3-1 Lower Power Consumption
 - 3-2 Large-Capacity, High-Speed Flash Memory
 - 3-3 MCU Development Platform
4. Development Environment
5. Summary

Contents



1. Concept of RX Family
2. Features of RX CPU
 - 2-1 Improved Performance
 - 2-2 Improved Code Efficiency
 - 2-3 Enhanced Flexibility
3. RX Family Product Evolution
 - 3-1 Lower Power Consumption
 - 3-2 Large-Capacity, High-Speed Flash Memory
 - 3-3 MCU Development Platform
4. Development Environment
5. Summary



Next-generation CPU

RX Family

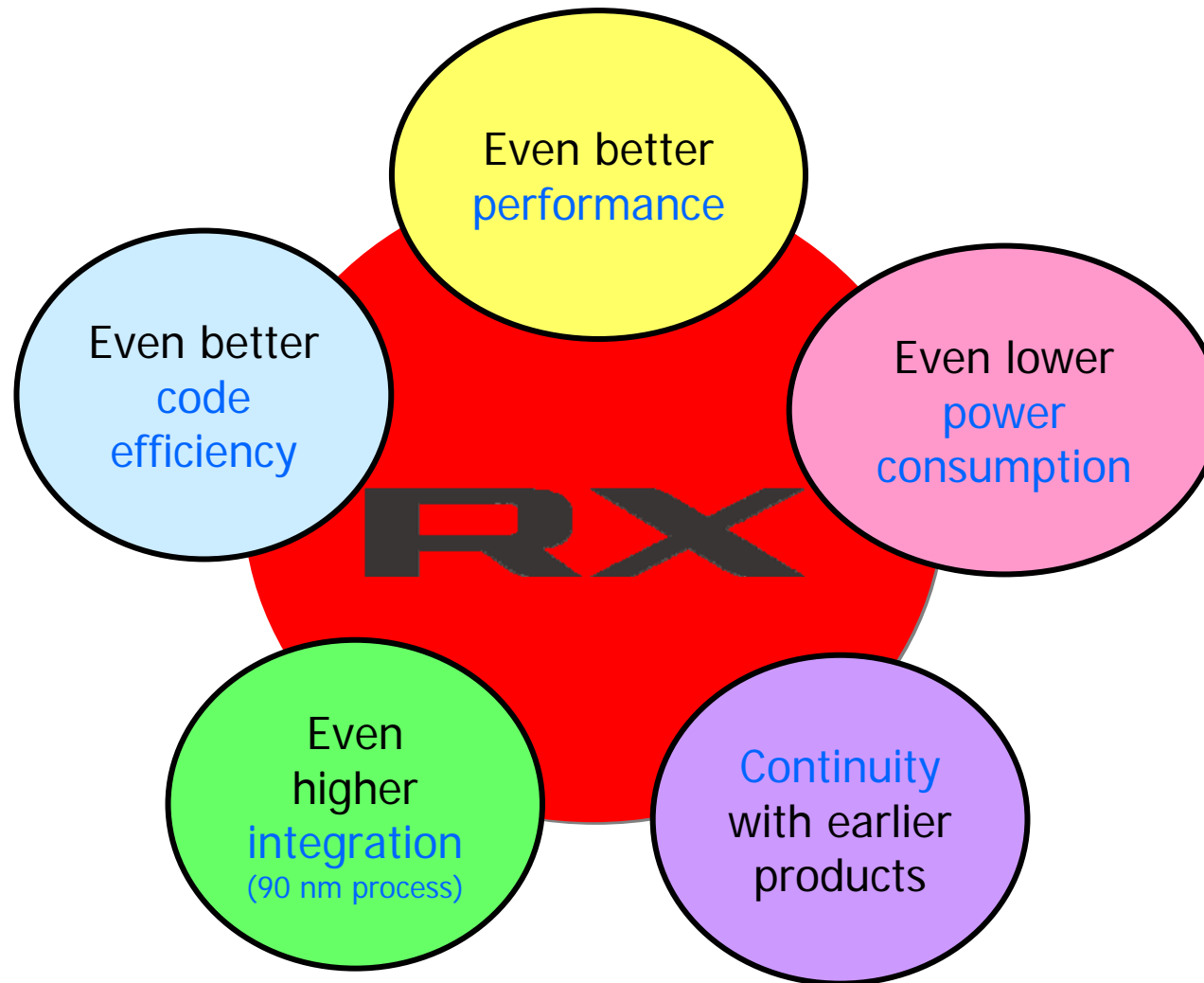


RX stands for "Renesas eXtreme" and designates a product combining outstanding performance and superb ease of use.

The RX Concept



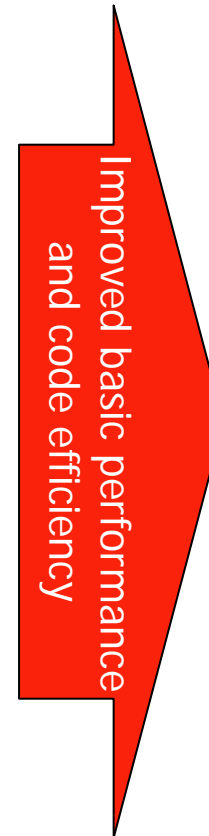
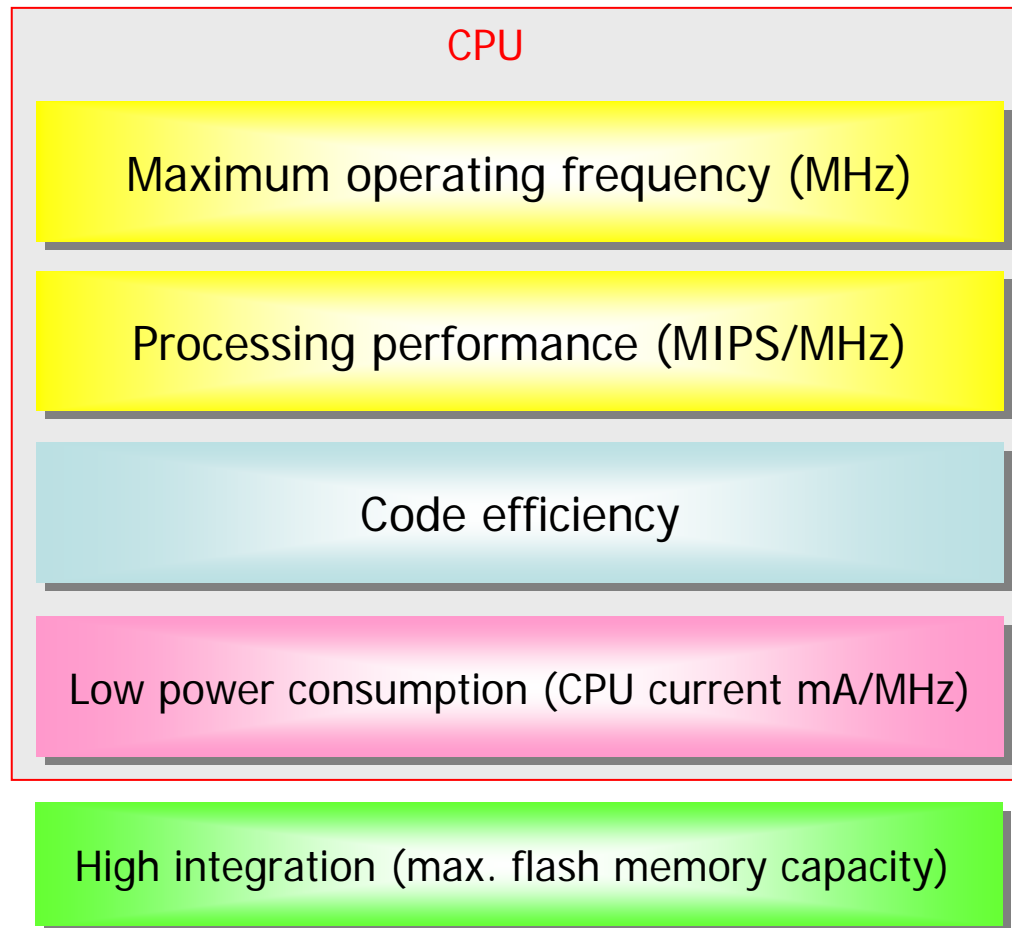
- Overcome problems and meet market demand by developing a new CPU.



RX Target Indices



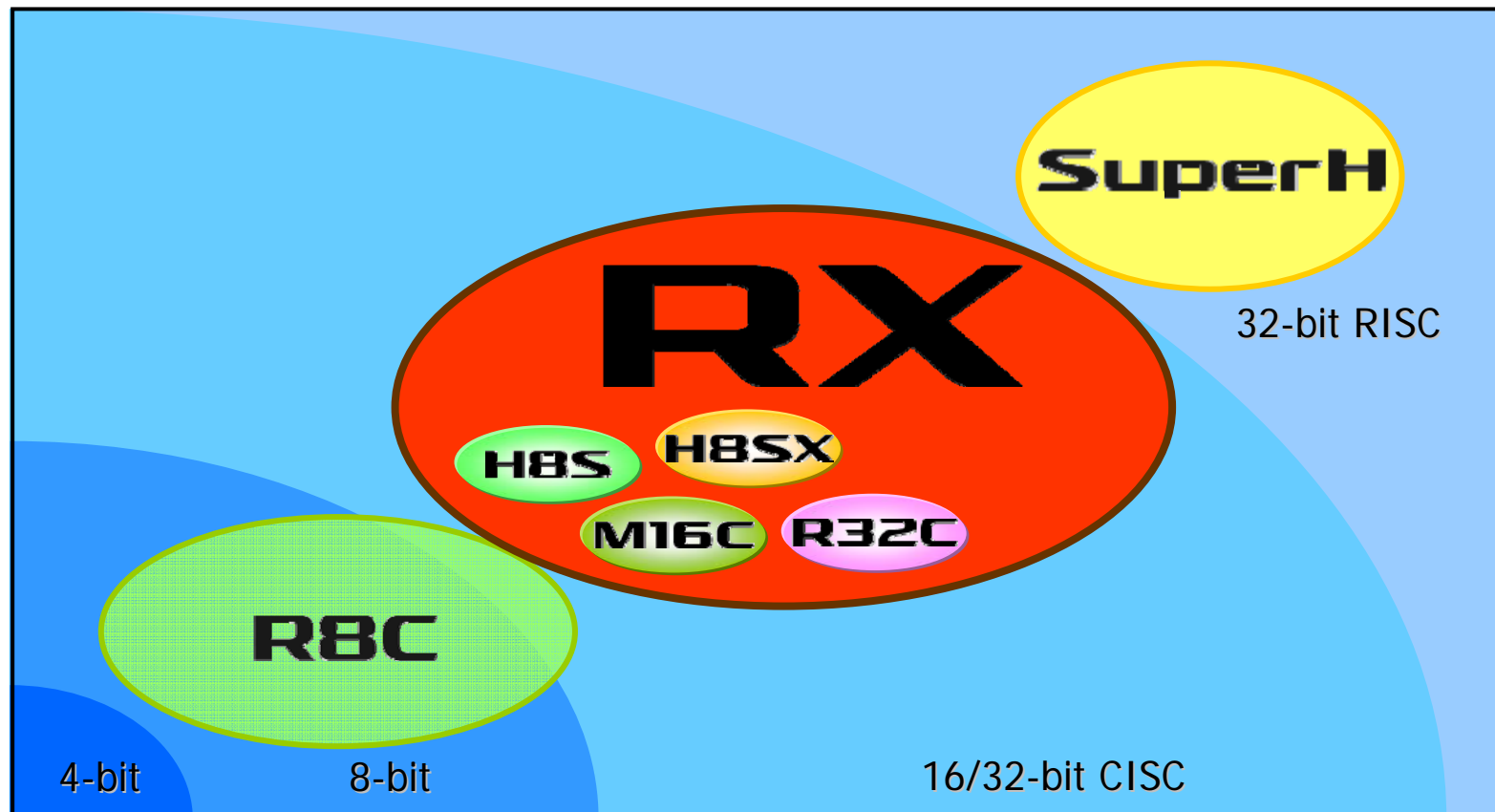
- The aim is to realize, in a 90 nm node device, higher maximum operating frequency, better performance, improved code efficiency, and lower power consumption than previous products.



Comparison with previous products	Design target
5 ×	200 MHz
2 ×	1.25 MIPS/MHz
30% improvement	
1/3	0.03 mA/MHz
4 ×	4 MB

A Renesas MCU Lineup with the Future in Mind

- Maintain continuity with the H8S, H8SX, M16C, and R32C families in the 16-bit and 32-bit zones. Further expand the market by extending the RX Family product lineup.
- In the embedded MCU market, cover requirements for 8-bit, 16-bit, and 32-bit products with the R8C, RX, and SH lineups.



Contents



1. Concept of RX Family
2. Features of RX CPU
 - 2-1 Improved Performance
 - 2-2 Improved Code Efficiency
 - 2-3 Enhanced Flexibility
3. RX Family Product Evolution
 - 3-1 Lower Power Consumption
 - 3-2 Large-Capacity, High-Speed Flash Memory
 - 3-3 MCU Development Platform
4. Development Environment
5. Summary

Features of RX CPU – Efforts to Improve Performance

■ Five-stage pipeline

The five-stage pipeline configuration speeds up instruction execution.

■ Harvard architecture

Parallel execution of instruction fetches and memory accesses boosts pipeline performance.

■ DSP arithmetic functions

DSP functions enable memory-to-memory multiply-accumulate operations and register-to-register multiply-accumulate operations.

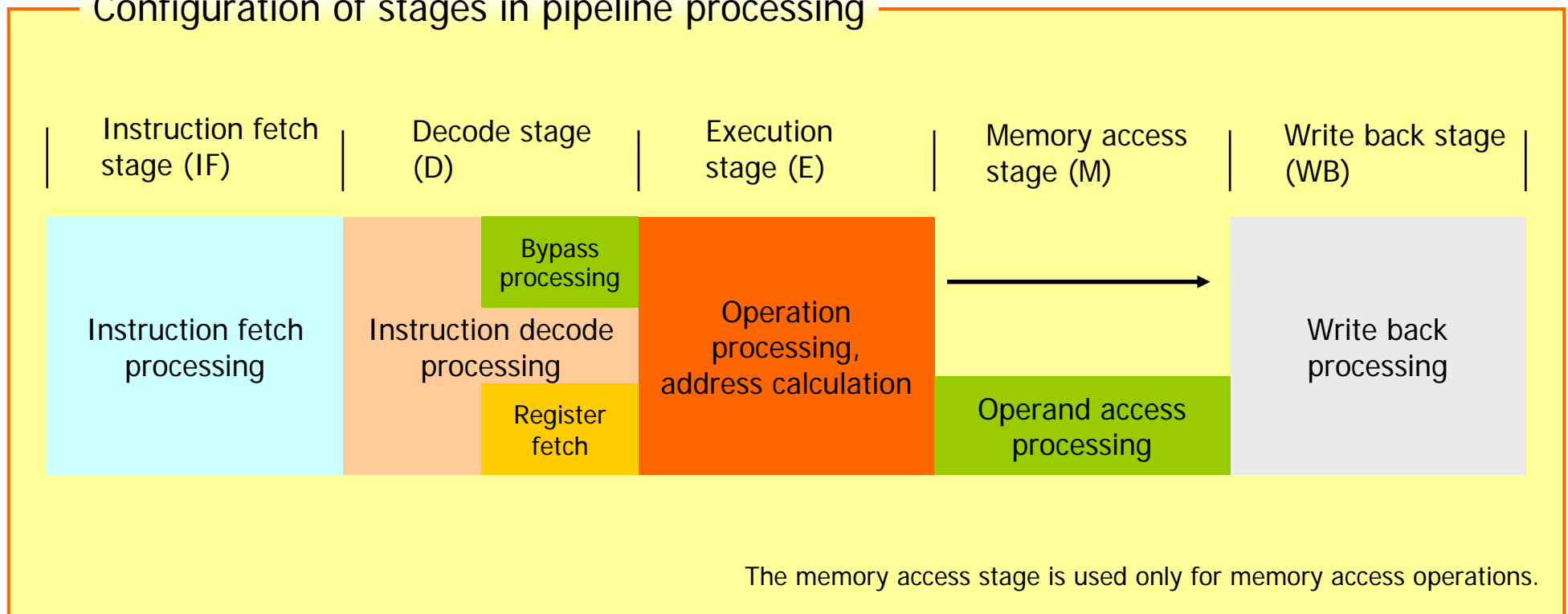
■ Single-precision floating-point unit

The single-precision floating-point unit uses general registers for operations.

Five-Stage Pipeline

- The five-stage pipeline configuration enables a speed increase up to 200 MHz.

Configuration of stages in pipeline processing

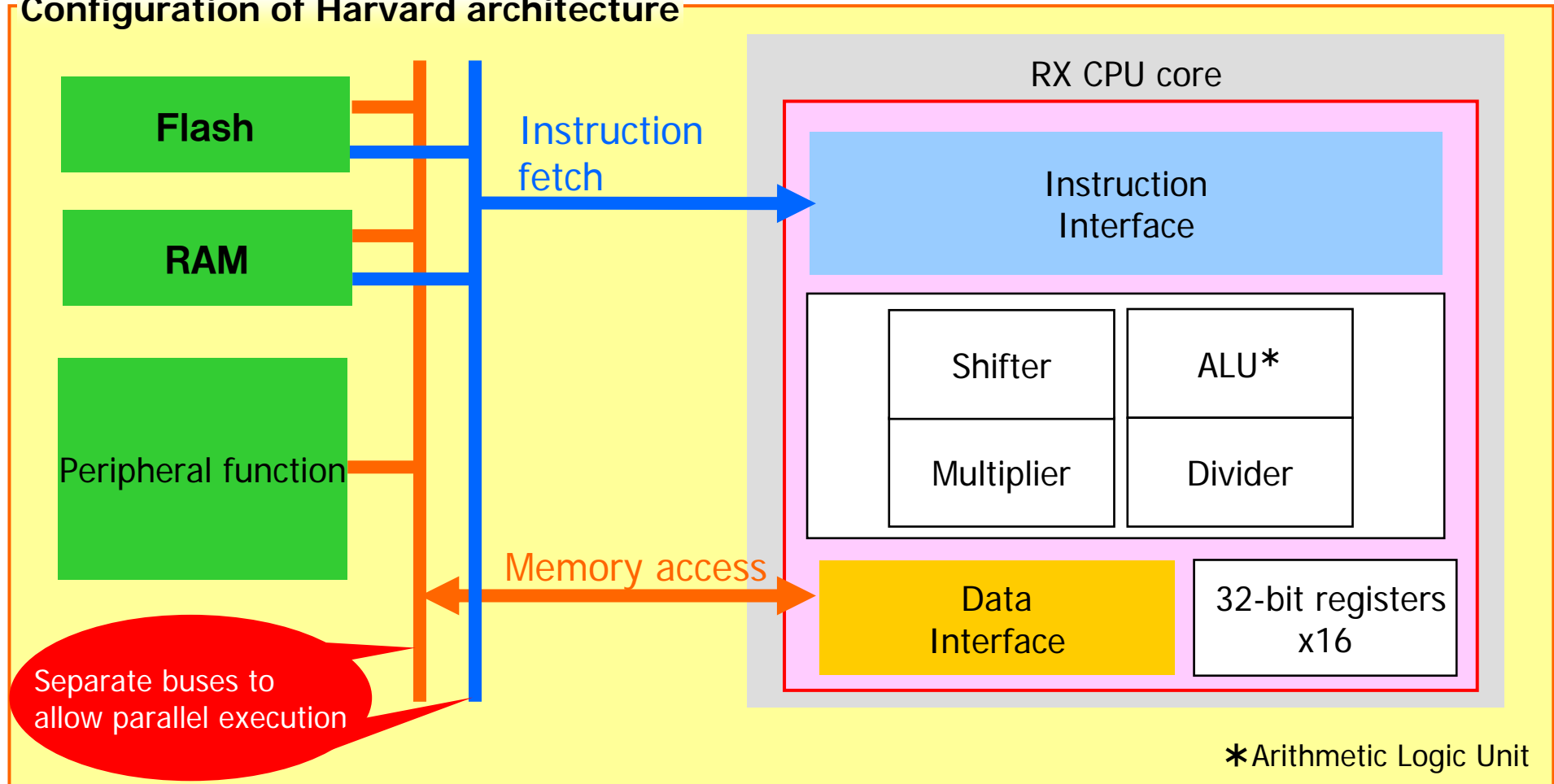


Harvard Architecture



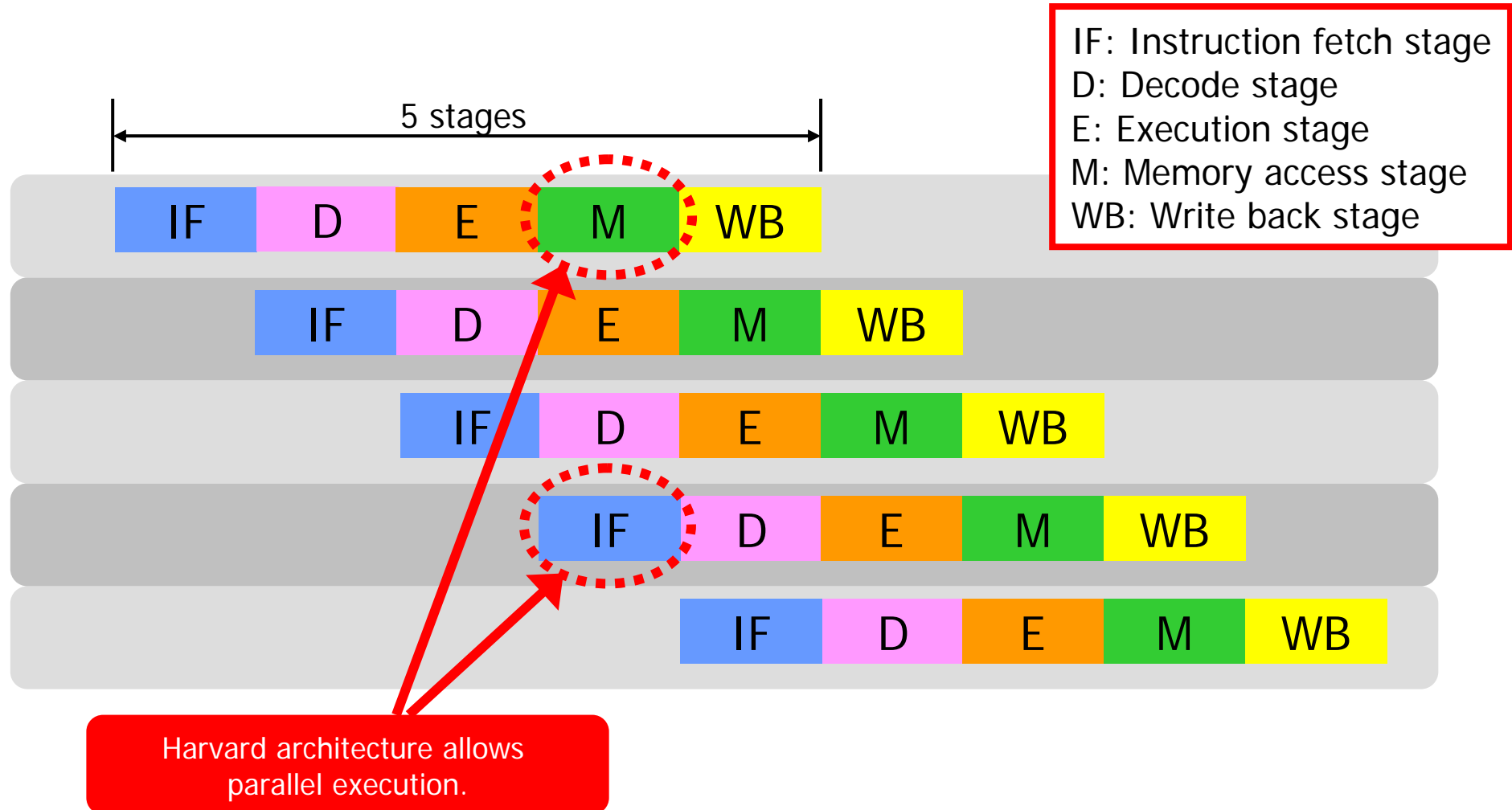
- Parallel execution of instruction fetches and memory accesses boosts pipeline performance.

Configuration of Harvard architecture



Basic Pipeline Operations

- Parallel execution of instruction reads and memory accesses realizes the ideal execution rate of one instruction per clock cycle.

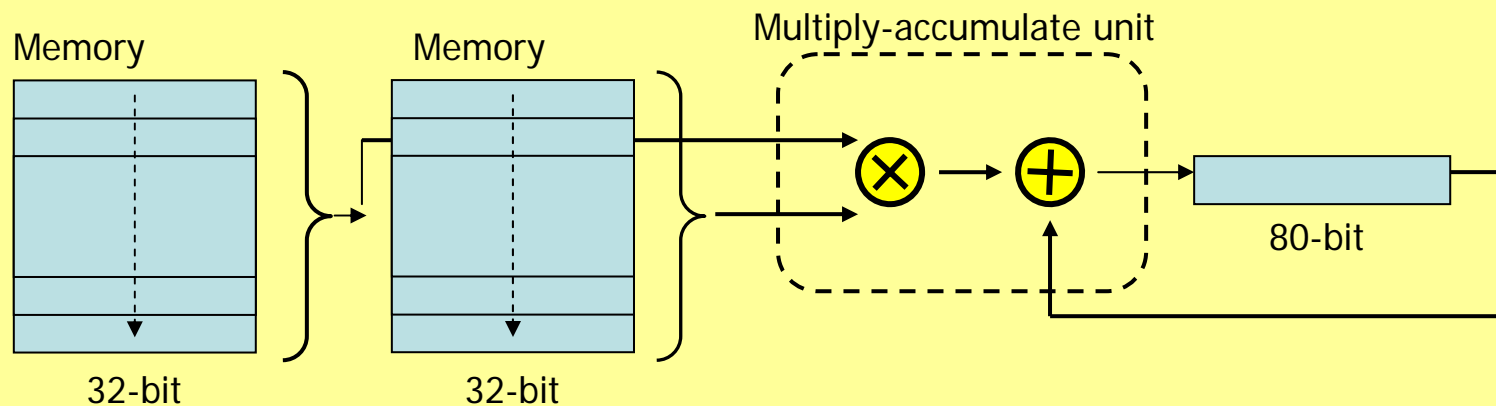


DSP Arithmetic Functions



- The memory-to-memory multiply-accumulate operation instruction (RMPA) is ideal for sequential multiply-accumulate operations on accumulated data, resulting in more precise 32-bit arithmetic operations.

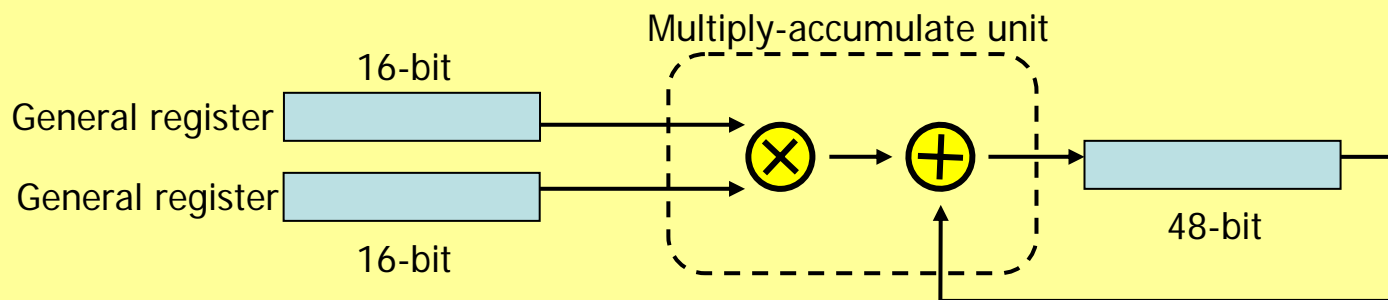
Memory-to-memory multiply-accumulate operation instruction (RMPA)



- The register-to-register multiply-accumulate operation instruction (MAC) eliminates the need for data transfer from general registers to memory, resulting in high-speed operation.

Register-to-register multiply-accumulate operation instruction (MAC)*

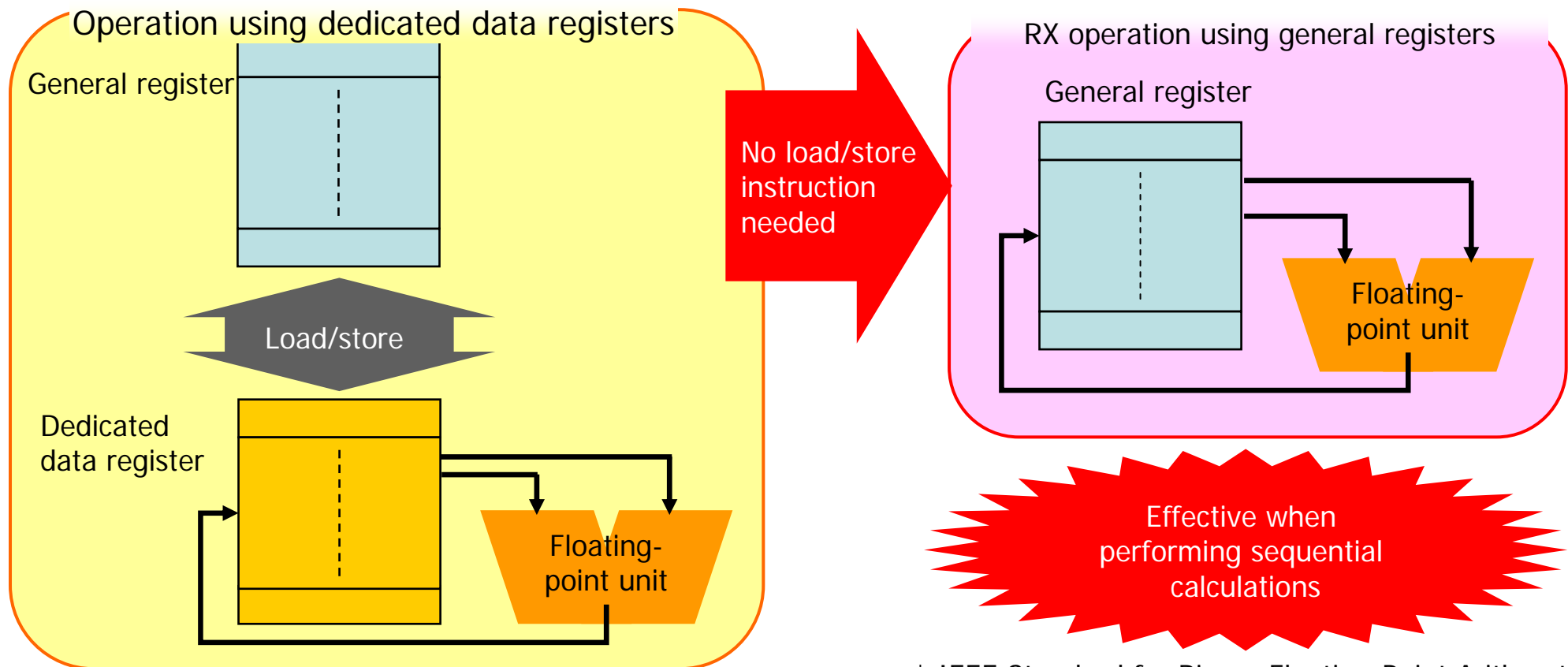
* Option



Single-Precision Floating-Point Unit



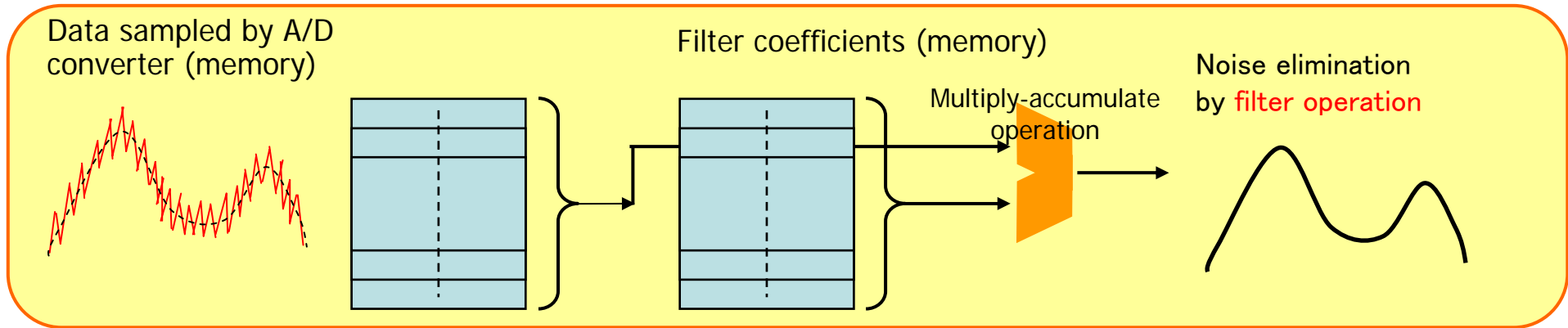
- The single-precision (32-bit) data format defined in IEEE 754* is supported.
- The exceptions defined in IEEE 754 are supported.
- subtract, multiply, divide, and integer-convert instructions general registers are supported.



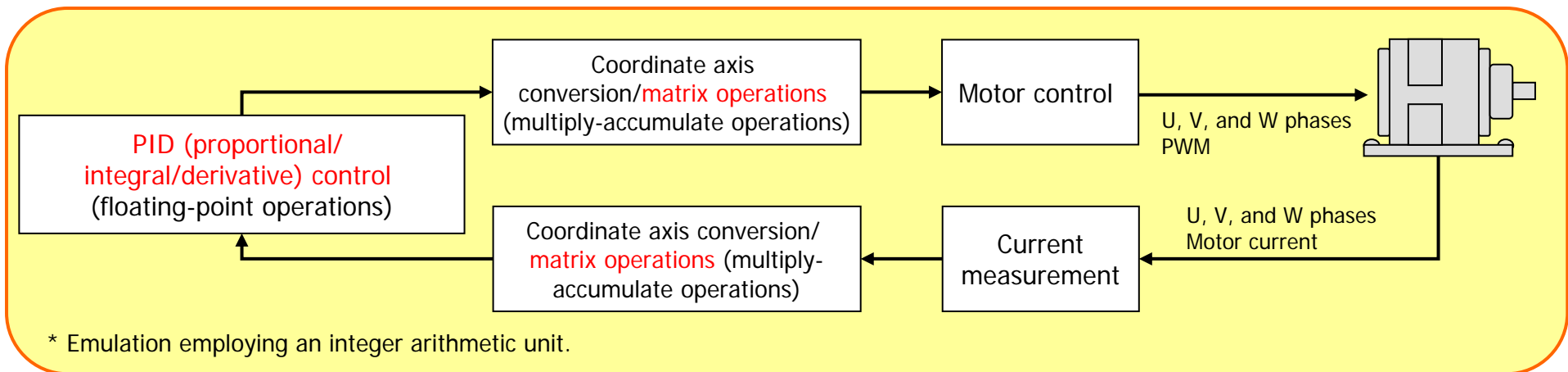
Effective when performing sequential calculations

* IEEE Standard for Binary Floating-Point Arithmetic

- The memory-to-memory multiply-accumulate operation instruction (RMPA) is ideal for filtering operations requiring high speed and accuracy.



- The register-to-register multiply-accumulate operation instruction (MAC) is ideal for speeding up matrix operations such as coordinate axis conversion .
- Floating-point unit operations using general registers are ideal for PID control involving sequential multiply and divide operations. (They are even faster than using dedicated data registers or a floating-point library.*)



* Emulation employing an integer arithmetic unit.

Contents



1. Concept of RX Family
2. Features of RX CPU
 - 2-1 Improved Performance
 - 2-2 Improved Code Efficiency
 - 2-3 Enhanced Flexibility
3. RX Family Product Evolution
 - 3-1 Lower Power Consumption
 - 3-2 Large-Capacity, High-Speed Flash Memory
 - 3-3 MCU Development Platform
4. Development Environment
5. Summary

Analysis of Instruction Frequency and Instruction Enhancement

- The instruction frequency in software applications of various types was analyzed, and enhancements were made to the most frequently used instructions.

Analysis of instruction frequency



Enhancement on basis of instruction frequency

- Move instruction shortened by 2 bytes (MOV: immediate value to memory, memory to register)
- Indirect addressing of indexed registers
- Move instructions with post-incrementing and pre-decrementing

- Conditional branch instructions shortened by 1 byte (BEQ, BNE)

- Compare instruction shortened by 2 bytes (CMP: memory-register)

- Subroutine branch instruction shortened by 1 byte (BSR)

- Add instruction shortened by 2 bytes (ADD: memory + register, immediate value + register)
- Three-operand format

Variable-Length Byte-Unit Instructions



- Assignment of shorter code length to frequently used instructions

One-byte instructions (high-frequency conditional branch instructions)

Relative conditional branch: BEQ,¹ BNE¹
Non-relative conditional branch: BRA

Two-byte instructions (high-frequency move and compare instructions)

Move: MOV (register-register move; memory-memory move,² load,² and store²)
Compare: CMP (register-register compare, immediate value-register compare)
Add: ADD (register + register, immediate value + register)
Subroutine branch: BSR
Multiply: MUL (register × register)

Three-byte instructions (control operation instructions)

Divide: DIV (register ÷ register)
Multiply-accumulate: MAC (register × register)
Floating-point add: FADD (register + register)
Floating-point multiply: FMUL (register × register)

Notes

1. 3-bit PC forward relative
2. Memory access by register indirect addressing

Indirect Addressing of Indexed Registers

- Tables can be referenced more efficiently by using indirect addressing of indexed registers (use of address offsets).

Conventional code

```
ADD    R2,R3
MOV.B  [R3],R4
```

2 instructions reduced to 1 instruction

RX code **MOV.B [R2,R3],R4**

Register R3: Base address

Register R2: Index

Register R4: XX

Memory: XX

Address

Move

- Using indirect addressing of indexed registers (use of address offsets) involves data size scaling (W: $\times 2$, L: $\times 4$)

Conventional code

```
SHLL   #2,R2
ADD    R2,R3
MOV.L  [R3],R4
```

3 instructions reduced to 1 instruction

RX code **MOV.L [R2,R3],R4**

Register R3: Base address

Register R2: Index

Register R4: XXXXXX

Memory: XXXXXX

Address

Move

x4

Indirect Register Addressing with Post-Incrementing and Pre-Decrementing



- Tables can be referenced successively with more efficiency by using load and store instructions with post-incrementing.

Conventional code
`MOV.B [R2],R3`
`ADD #1,R2`

2 instructions reduced to 1 instruction

RX code `MOV.B [R2+],R3`

Memory

Register R2

Register R3

Move

Incremented

Note: The amount added is scaled based on the data size (W: $\times 2$, L: $\times 4$).

- Tables can be referenced successively with more efficiency by using load and store instructions with pre-decrementing.

Conventional code
`SUB #1,R2`
`MOV.B [R2],R3`

2 instructions reduced to 1 instruction

RX code `MOV.B [-R2],R3`

Memory

Register R2

Register R3

Move

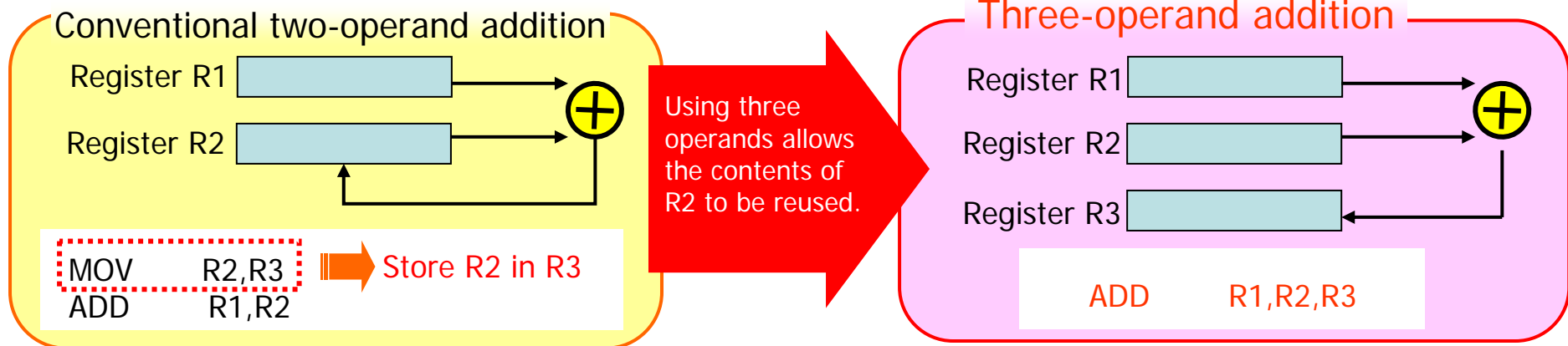
Decrement

Note: The amount subtracted is scaled based on the data size (W: $\times 2$, L: $\times 4$).

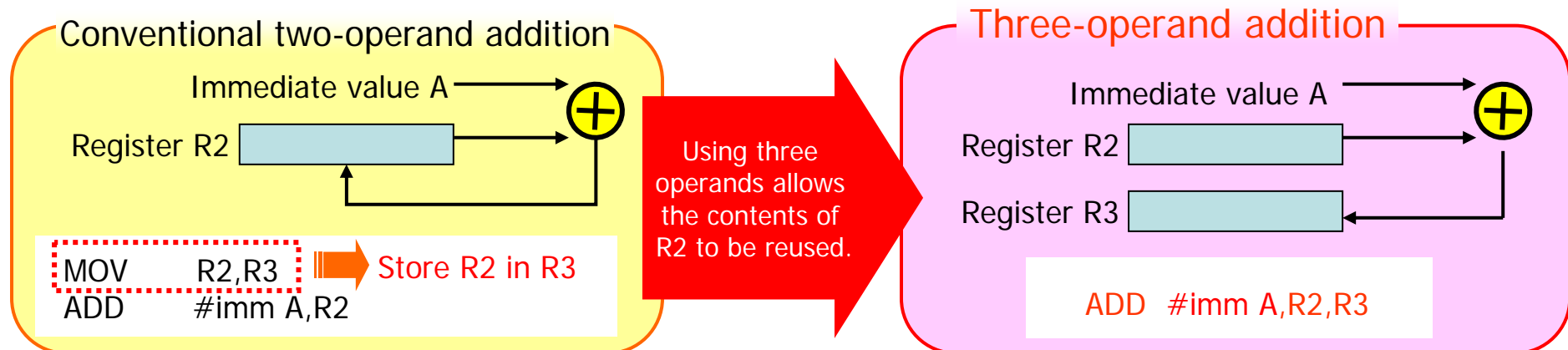
Three-Operand Format



- Three registers can be specified at once, increasing programming flexibility and code efficiency.



- An immediate value and two registers can be specified at once.



Contents

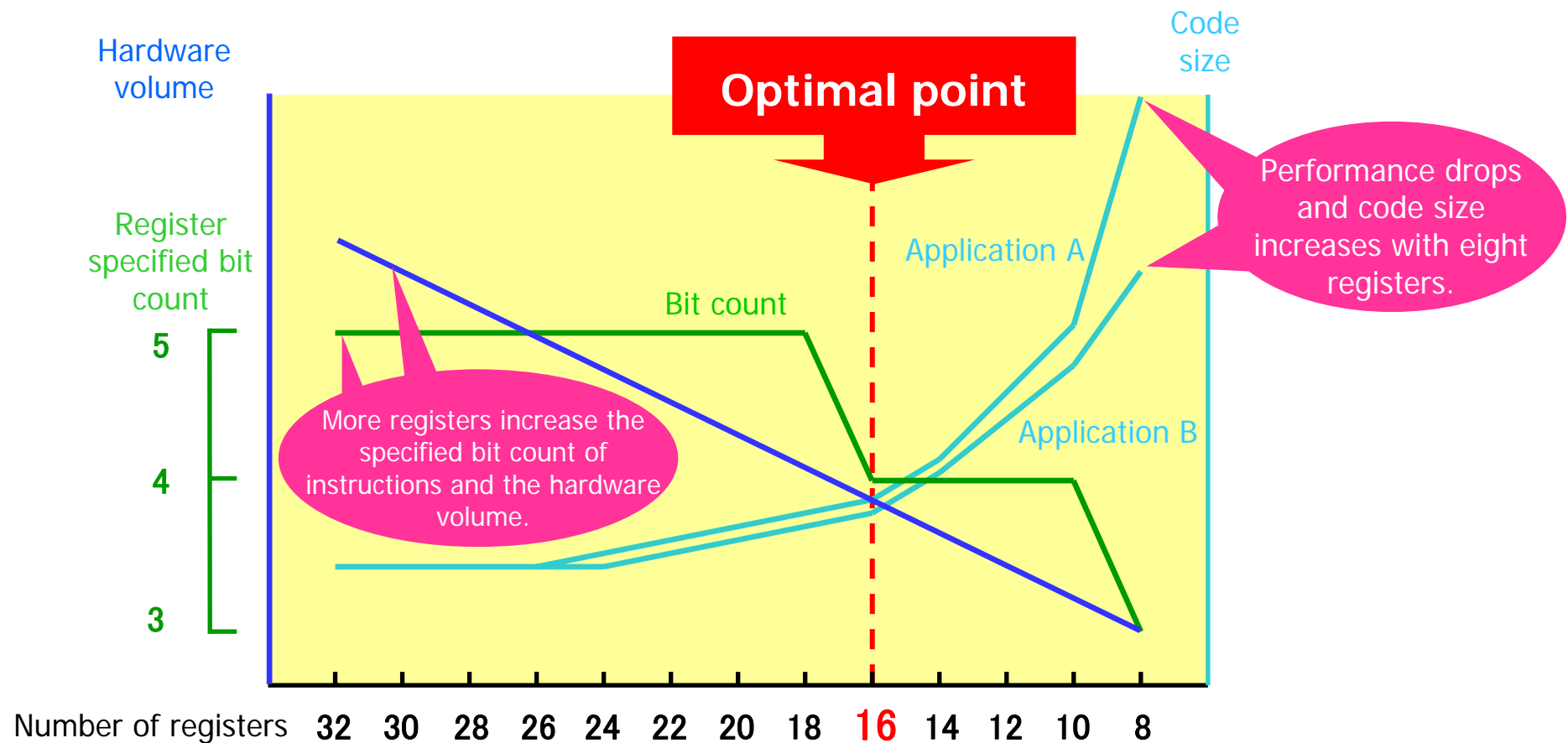


1. Concept of RX Family
2. Features of RX CPU
 - 2-1 Improved Performance
 - 2-2 Improved Code Efficiency
 - 2-3 Enhanced Flexibility
3. RX Family Product Evolution
 - 3-1 Lower Power Consumption
 - 3-2 Large-Capacity, High-Speed Flash Memory
 - 3-3 MCU Development Platform
4. Development Environment
5. Summary

Optimization of register number

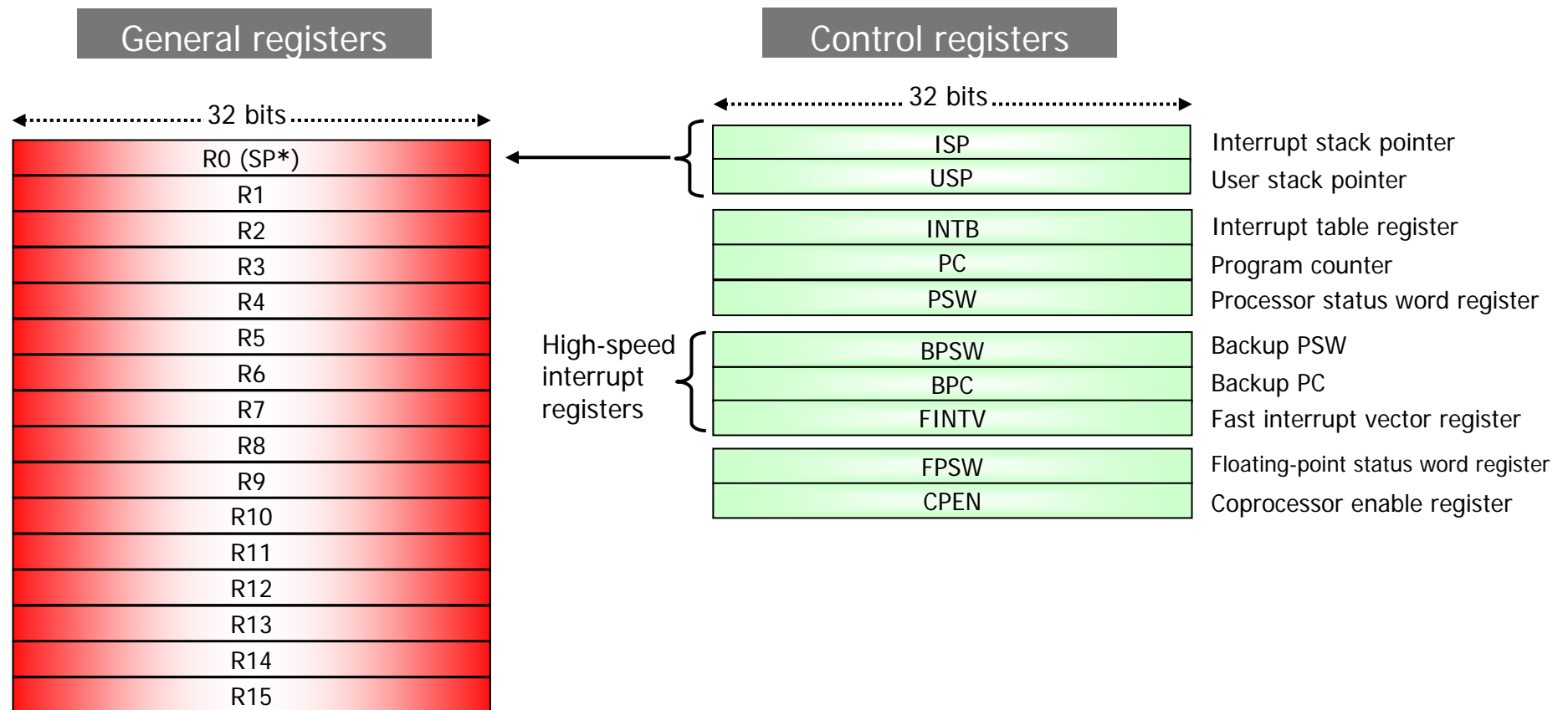
- A configuration of sixteen 32-bit general registers was chosen as providing the best performance and code size while minimizing increase in the hardware volume.

Results of software application benchmarks



Register Configuration: Sixteen 32-Bit General Registers

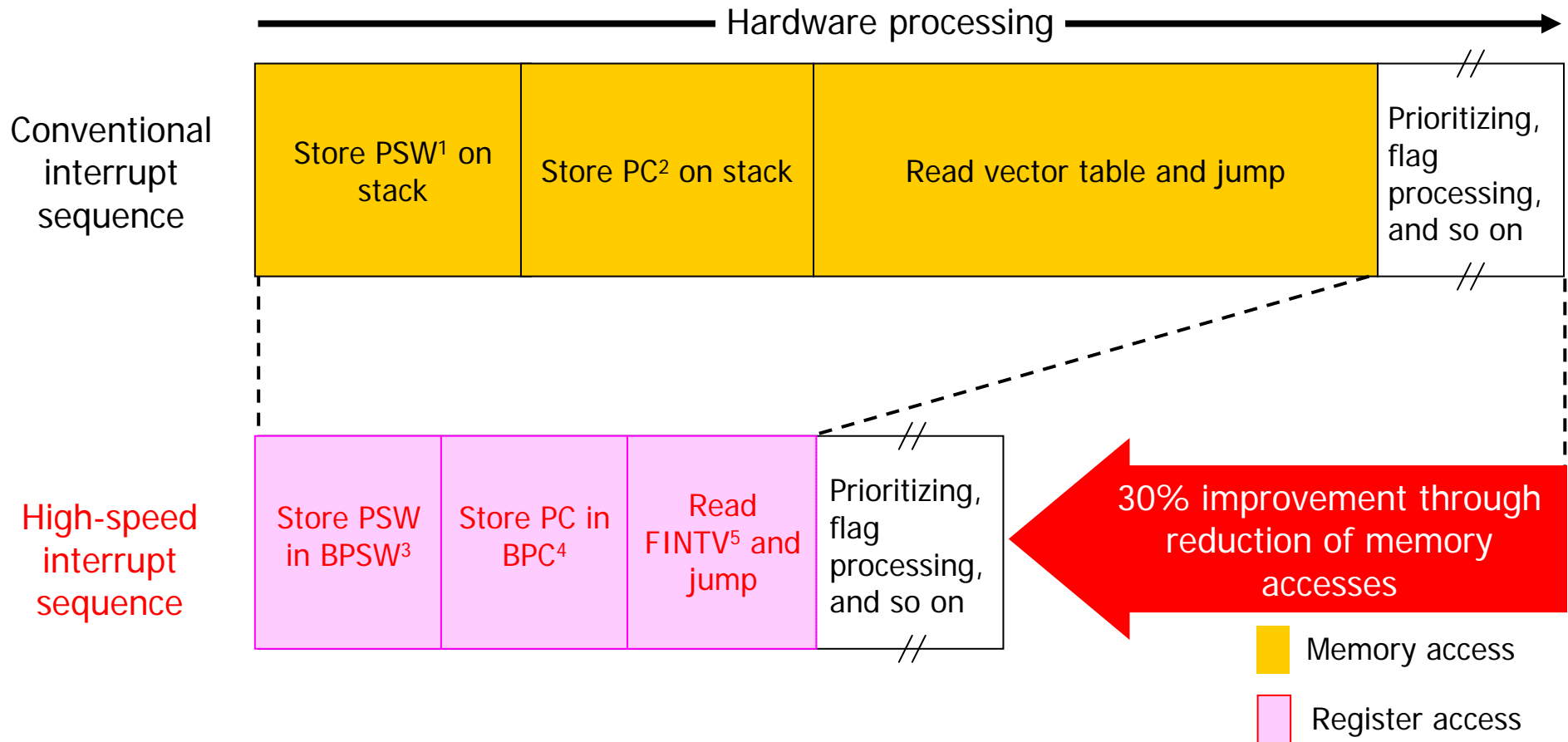
- Facilitating register-register operations reduces memory accesses.
- The use of general registers simplifies compiler optimization.
- High-speed interrupt registers speed up interrupt handling.
- Augmented general registers can be allocated for dedicated use by interrupts, enabling a further speed increase.



* Stack pointer

High-Speed Interrupt Sequence

- Processing states are reduced by about 30% through the use of high-speed interrupt registers.



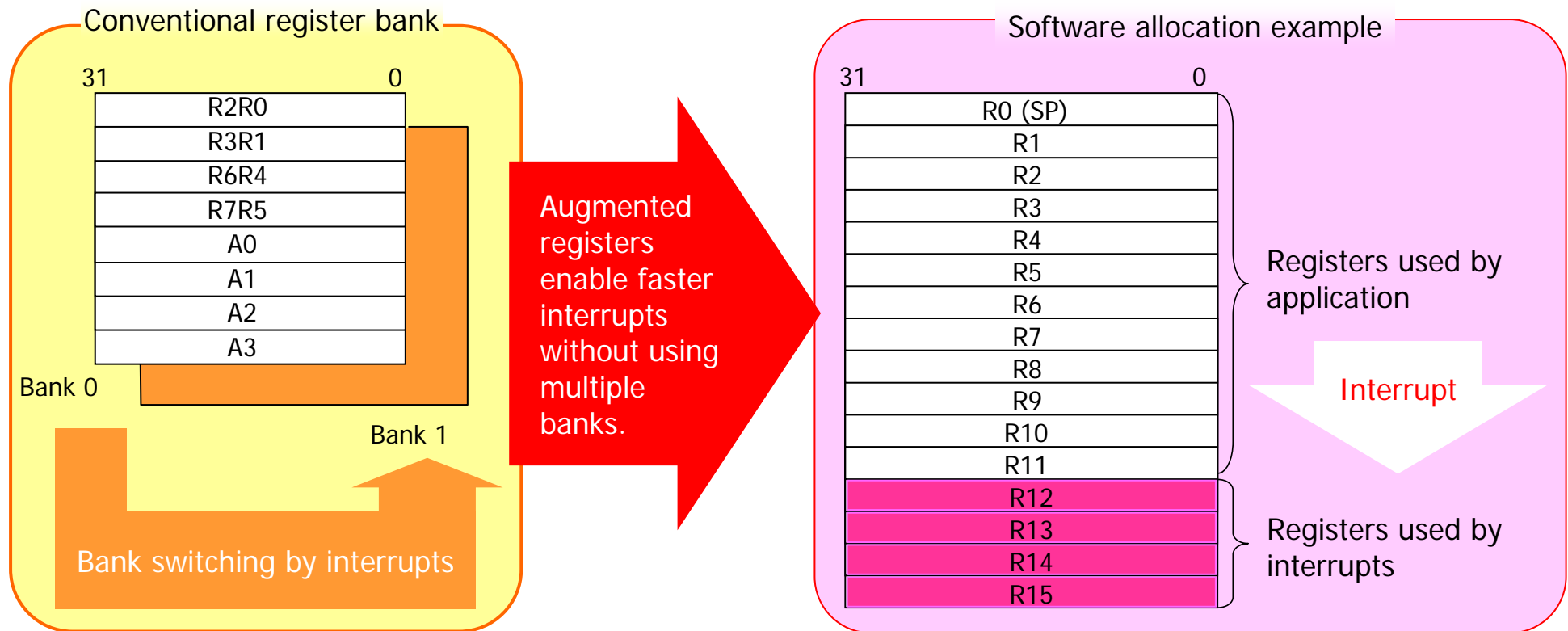
1. Program status word register
2. Program counter

3. Backup PSW
4. Backup PC
5. Interrupt vector register

} High-speed interrupt registers

Faster Interrupts through Register Allocation

- Augmented general registers can be allocated for dedicated use by interrupts* to increase speed.

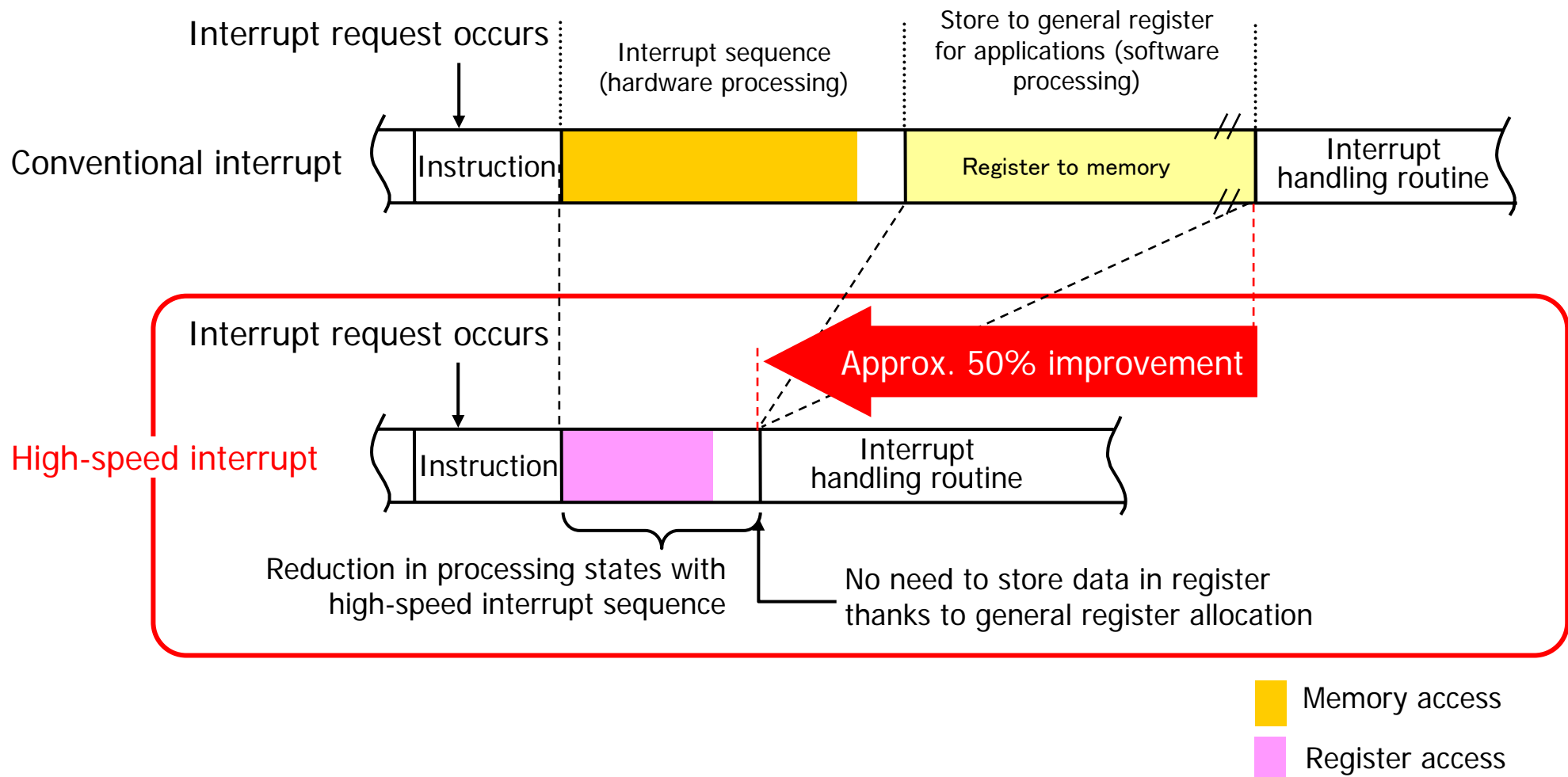


Flexibility for the programmer is increased and an optimal ratio of registers can be allocated to the application and to interrupts.

* to be supported by tools

Improved Interrupt Responsiveness

- High-speed interrupt sequences and allocation of general register boost performance by about 50%.



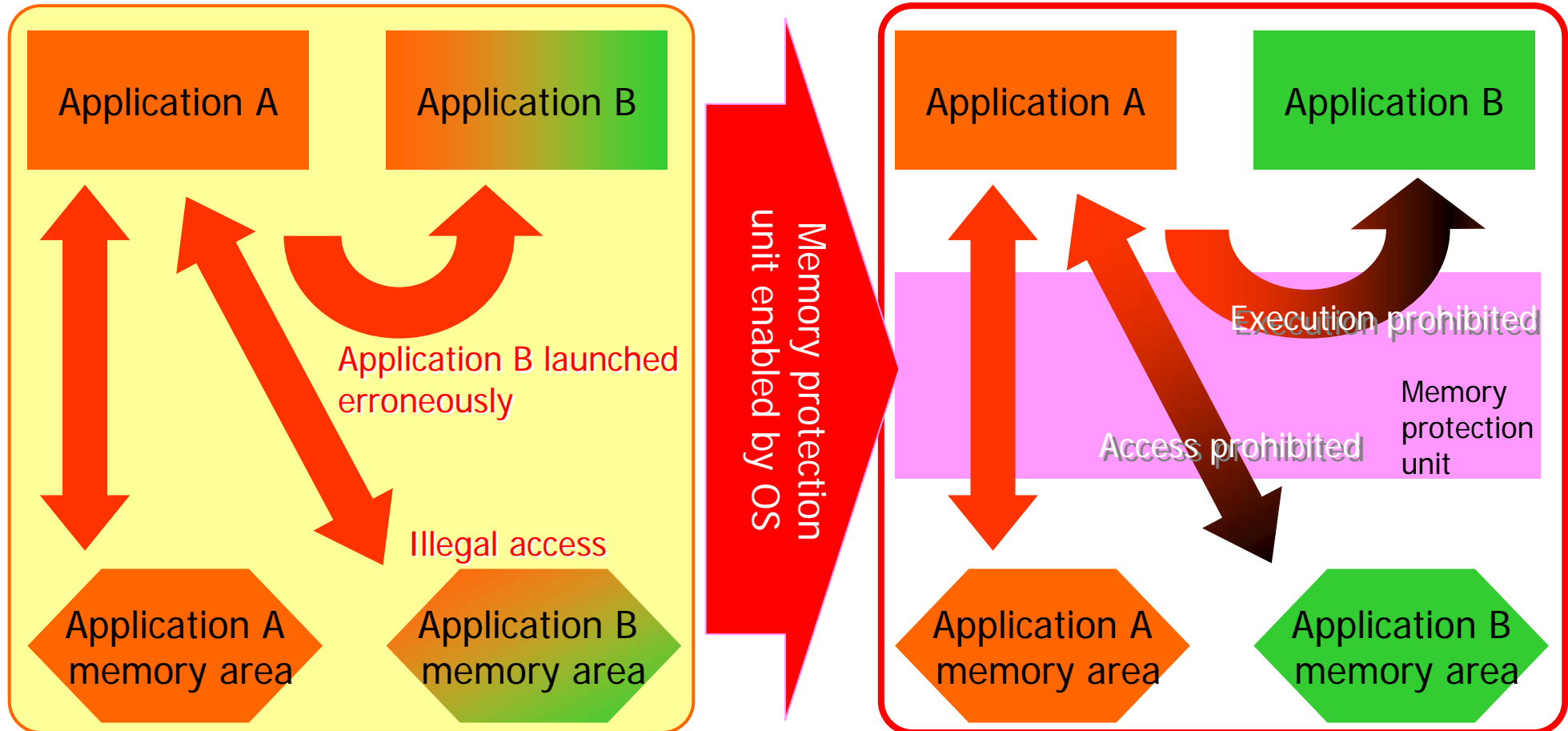
Memory Protection Unit

* Option

- The memory protection unit * increases system reliability.

If application A misbehaves, application B is affected as well.

If application A misbehaves, application B is not affected.

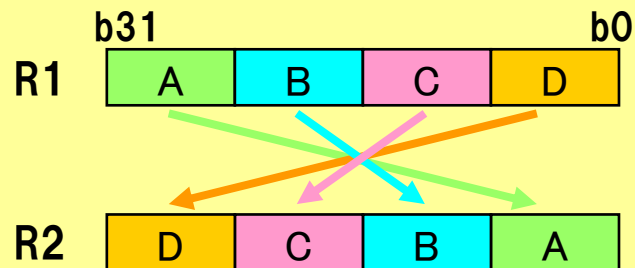


Bi-Endian Support

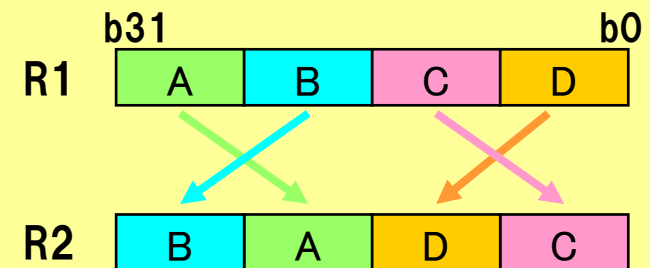


- Support for endian conversion instructions to maintain compatibility with data format of existing CPUs

Longword endian conversion instruction
REVL R1,R2



Word endian conversion instruction
REWV R1,R2



Contents



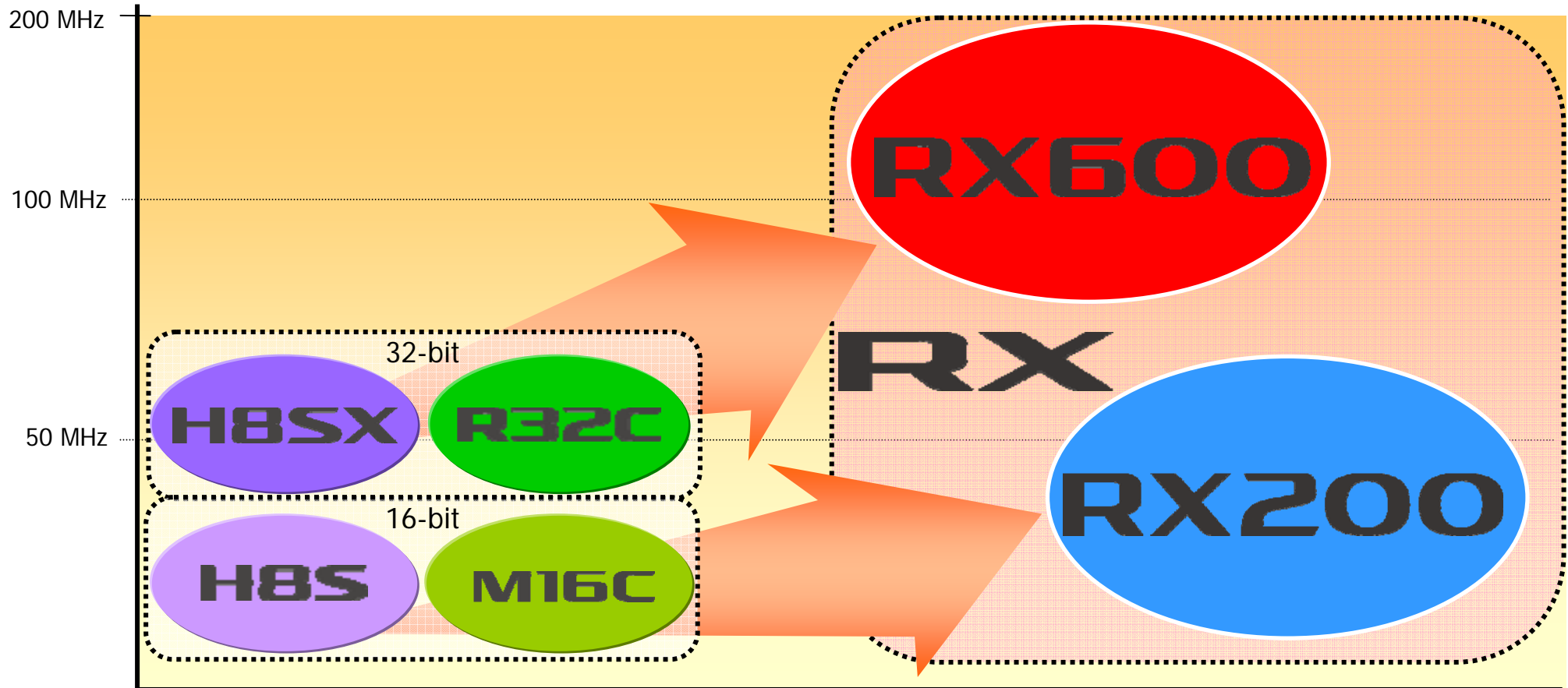
1. Concept of RX Family
2. Features of RX CPU
 - 2-1 Improved Performance
 - 2-2 Improved Code Efficiency
 - 2-3 Enhanced Flexibility
- 3. RX Family Product Evolution**
 - 3-1 Lower Power Consumption
 - 3-2 Large-Capacity, High-Speed Flash Memory
 - 3-3 MCU Development Platform
4. Development Environment
5. Summary

RX Family Series Evolution



- RX600 Series: 32-bit MCUs designed for high speed and excellent performance
- RX200 Series: 16-bit MCUs designed for low power consumption

Maximum operating frequency



RX600 Series



- This product series is designed for high-end applications demanding quick and high-performance execution of large-scale programs.

Item	Basic Configuration of RX600 Series
CPU core	RX CPU
Max. operating frequency	200 MHz
Power supply voltage	3.0 V/3.3 V/5.0 V
General registers	32-bit × 16
Floating-point unit	Single-precision floating-point unit (add, subtract, compare, multiply, divide, etc.)
Memory protection unit	Yes (option)
DSP	Yes (option)
Multiplier	32-bit multiplier
Divider	Yes
Flash ROM (for program storage)	256 KB to 4 MB
Flash ROM (for data storage)	Yes
RAM	64 KB to 256 KB
On-chip peripheral functions	Timers
	Serial interface
	A/D converter
	D/A converter
On-chip debugging functions	Yes
Performance	1.25 MIPS/MHz

Contents



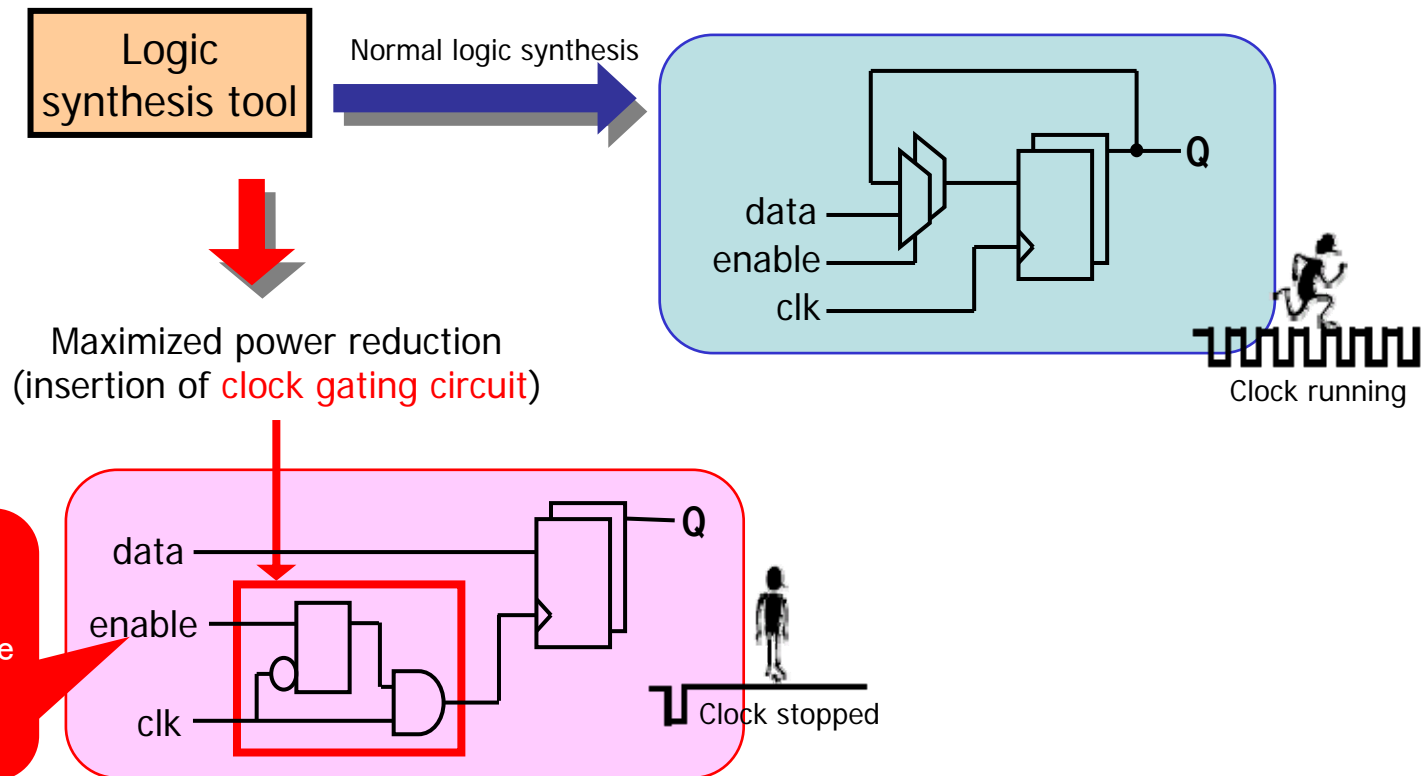
1. Concept of RX Family
2. Features of RX CPU
 - 2-1 Improved Performance
 - 2-2 Improved Code Efficiency
 - 2-3 Enhanced Flexibility
3. RX Family Product Evolution
 - 3-1 Lower Power Consumption
 - 3-2 Large-Capacity, High-Speed Flash Memory
 - 3-3 MCU Development Platform
4. Development Environment
5. Summary

Efforts to Achieve Lower Power Consumption



■ Clock gating design methodology

- Clock is cut off as appropriate to unused logic blocks.
 - Current consumption is reduced during normal operation.
- Ultrafine 90 nm process reduces load capacitance (of gates and wires).
 - High-speed operation is achieved and current consumption is reduced during normal operation.

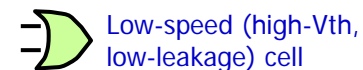
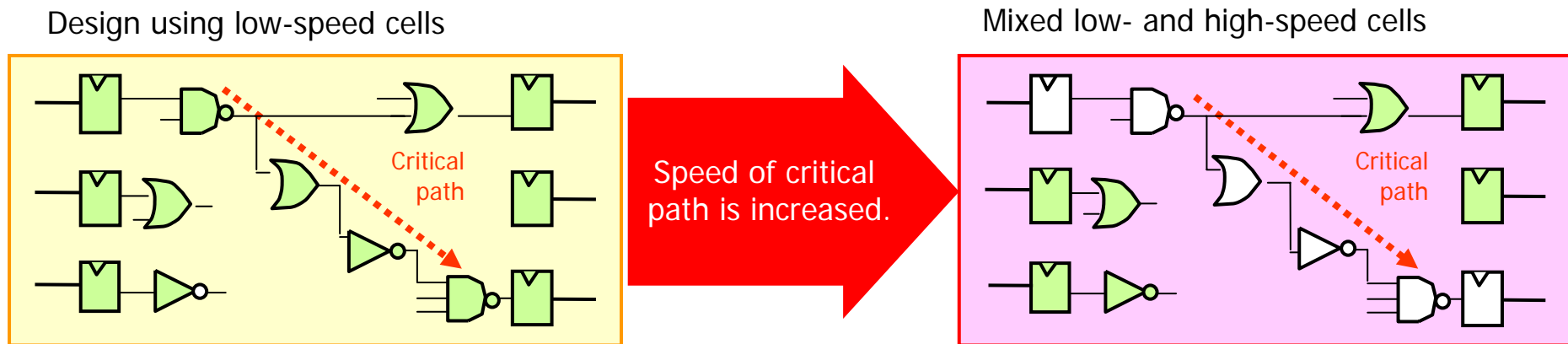


Efforts to Achieve Lower Power Consumption (Reduced Current Leakage)



Multi-threshold/leak current minimization design methodology

- The basic design uses low-speed (high-V_{th}, low-leakage) cells.
→ This reduces current leakage during normal operation.
- Critical paths are replaced with high-speed (low-V_{th}) cells.
→ Both high-speed operation and low power consumption are achieved.



Low-speed (high-V_{th},
low-leakage) cell



High-speed (low-V_{th})
cell

Efforts to Achieve Lower Power Consumption (Reduced Current Leakage)

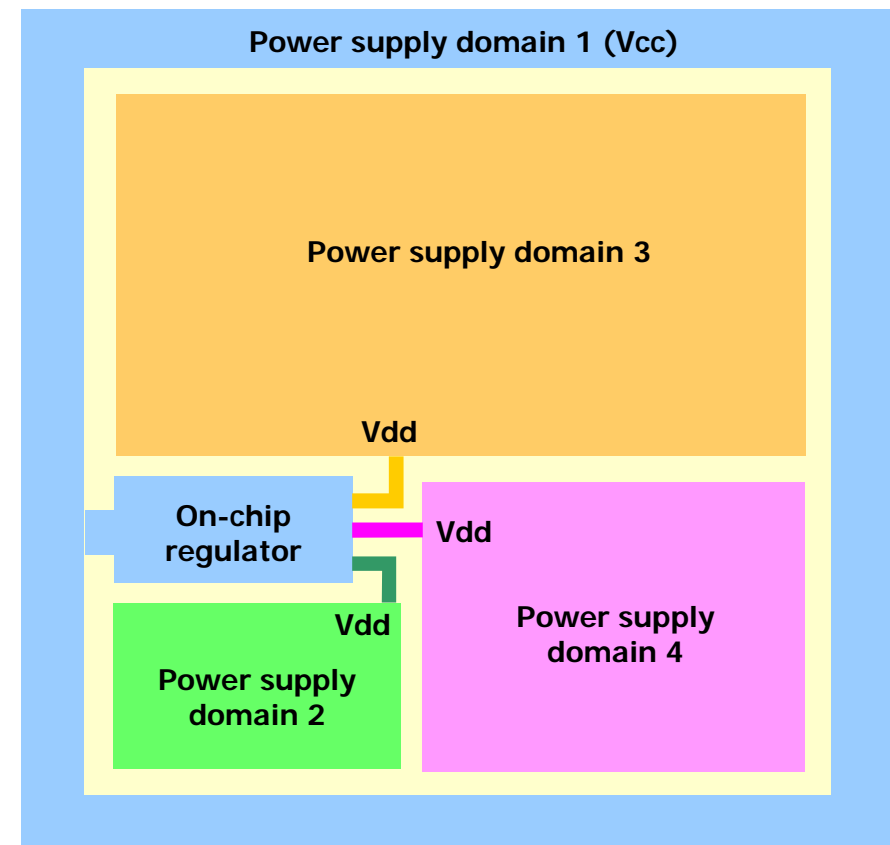
■ Power gating (power cutoff) design methodology

- Separate power supply domains are used, and cut off in power-down (low-power) modes.
 - Current leakage in power-down modes is reduced.
 - Multiple power-down modes are provided for maximum flexibility.

	Power supply domain 1	Power supply domain 2	Power supply domain 3	Power supply domain 4
Normal operation	○	○	○	○
Power-down mode 1	○	○	○	●
Power-down mode 2	○	○	●	●
Power-down mode 3	○	●	●	●

○:Power on

●:Power off



Contents



1. Concept of RX Family
2. Features of RX CPU
 - 2-1 Improved Performance
 - 2-2 Improved Code Efficiency
 - 2-3 Enhanced Flexibility
3. RX Family Product Evolution
 - 3-1 Lower Power Consumption
 - 3-2 Large-Capacity, High-Speed Flash Memory
 - 3-3 MCU Development Platform
4. Development Environment
5. Summary

High Integration and Large-Capacity Flash Memory from Renesas

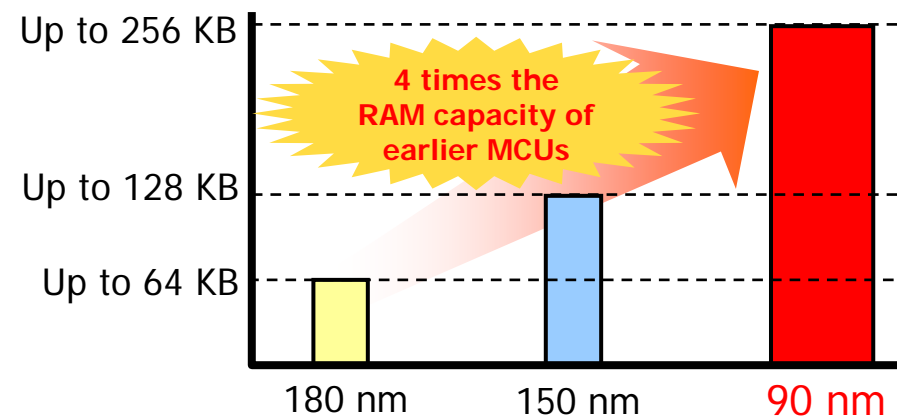
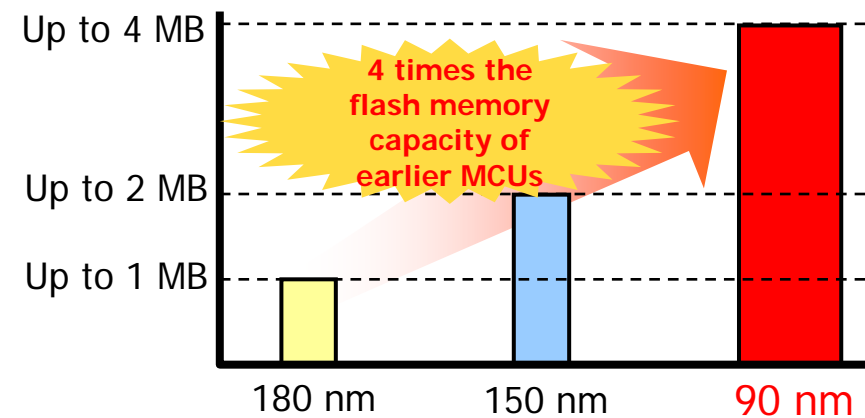


- We are using a cutting-edge **90 nm process** to create a lineup of products with large-capacity flash memory and abundant peripheral functions on-chip.

90 nm process MCU
2.5 Mbytes of on-chip memory

- CPU : SH-2A **200MHz**
- FPU : Single/Double
- **Flash : 2.5MB**
- **Data Flash : 128KB**
- RAM : 128KB
- A/D : 12bit x 37ch
- PKG : BGA2121-272

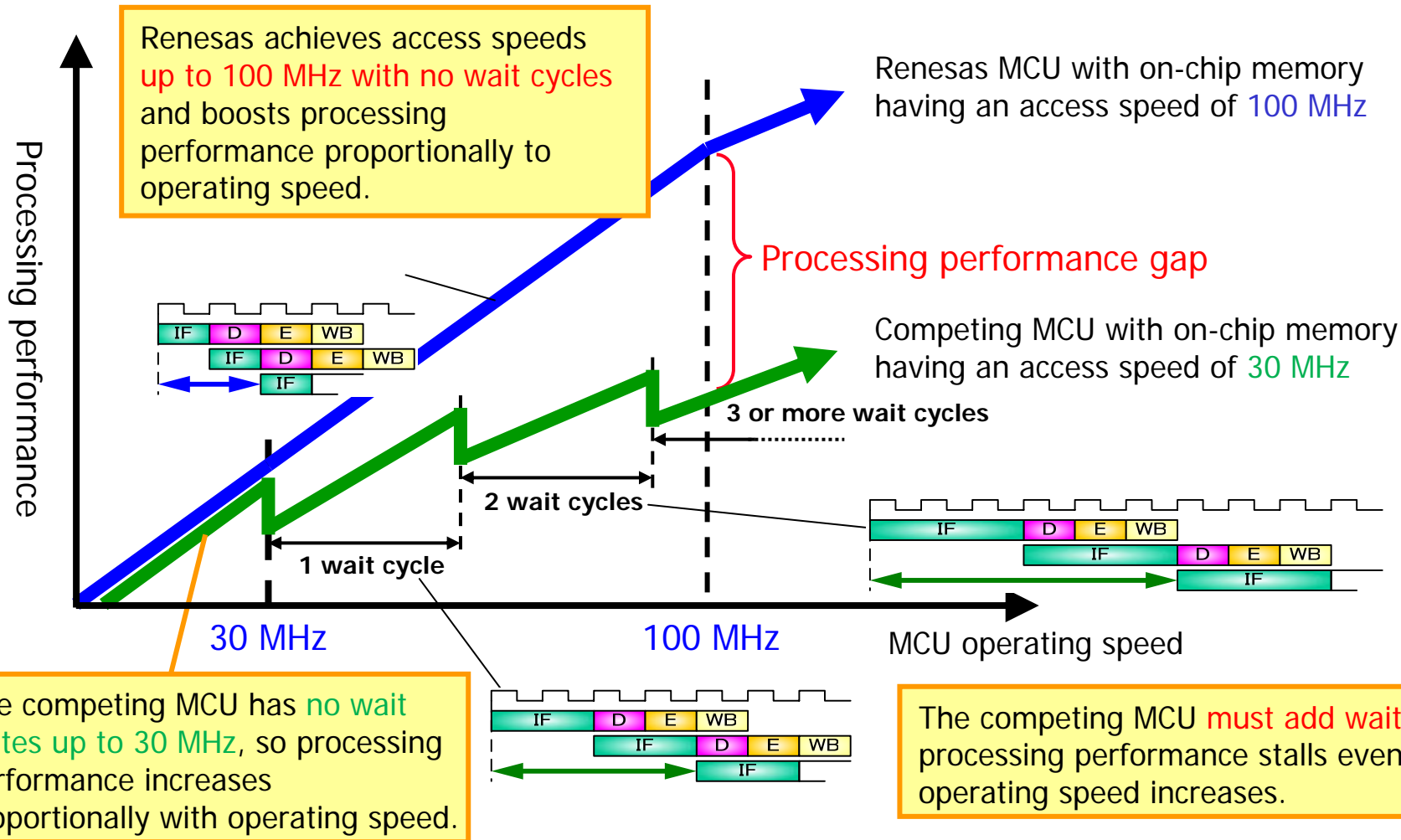
The engineering sample of the inaugural product was released in May 2007.



High-Speed Flash Memory from Renesas



- With speeds up to 100 MHz with no wait cycles, processing performance increases proportionally with operating speed.



Note: This example illustrates a case in which one wait cycle imposes overhead of 30%.

Contents

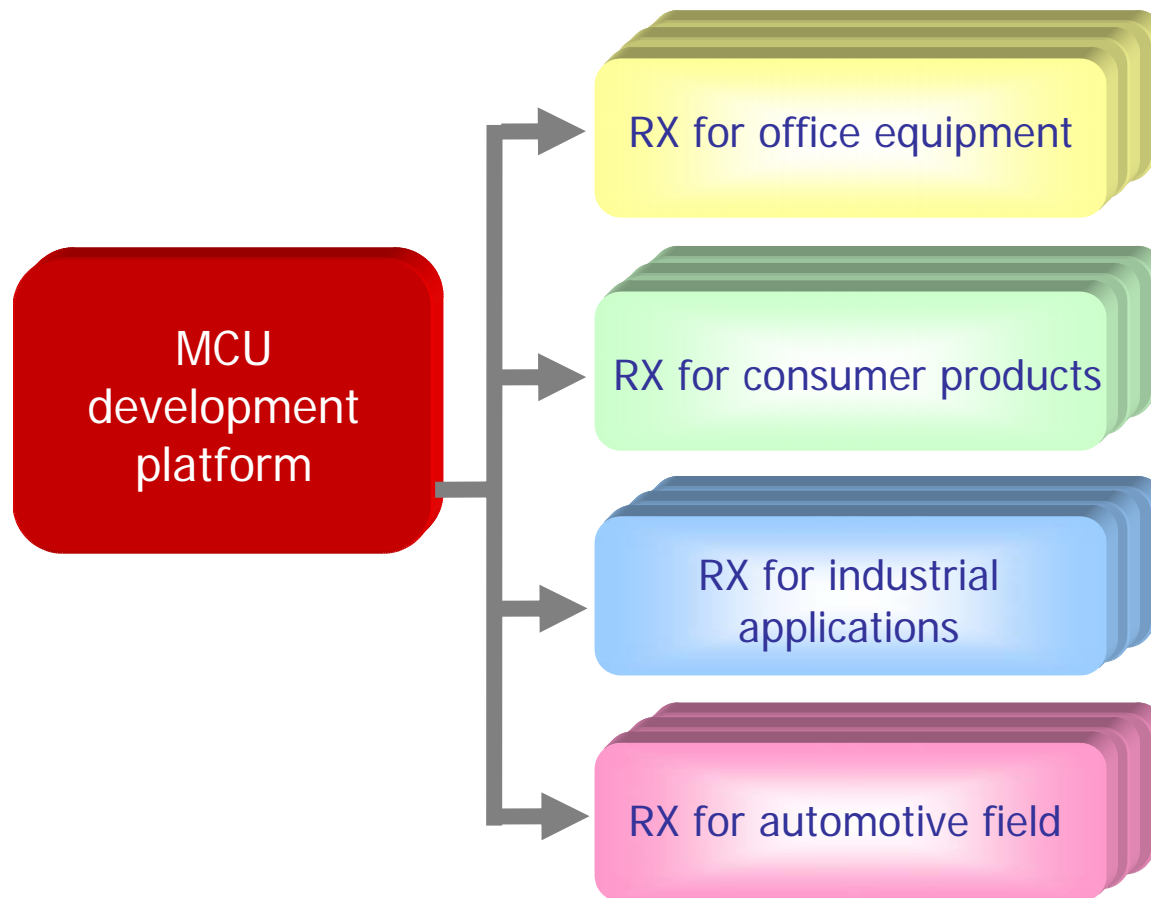


1. Concept of RX Family
2. Features of RX CPU
 - 2-1 Improved Performance
 - 2-2 Improved Code Efficiency
 - 2-3 Enhanced Flexibility
- 3. RX Family Product Evolution**
 - 3-1 Lower Power Consumption
 - 3-2 Large-Capacity, High-Speed Flash Memory
 - 3-3 MCU Development Platform**
4. Development Environment
5. Summary

Product Evolution Based on MCU Development Platform



- The RX family will include competitive products for many different fields.



Operating frequency

- 20 MHz to 200 MHz

Low power consumption

- Low-leakage process for reduced current consumption
- Precise power control for reduced power consumption during standby

On-chip peripheral functions

- Continuation of peripheral functions from earlier products
- Addition of new peripheral functions offering more flexibility
- ASSP functions (Ethernet, USB, CAN, LIN, FlexRay, etc.)

Many on-chip flash memory configurations

- High reliability and fast access
- Capacities from 128 KB to 4 MB

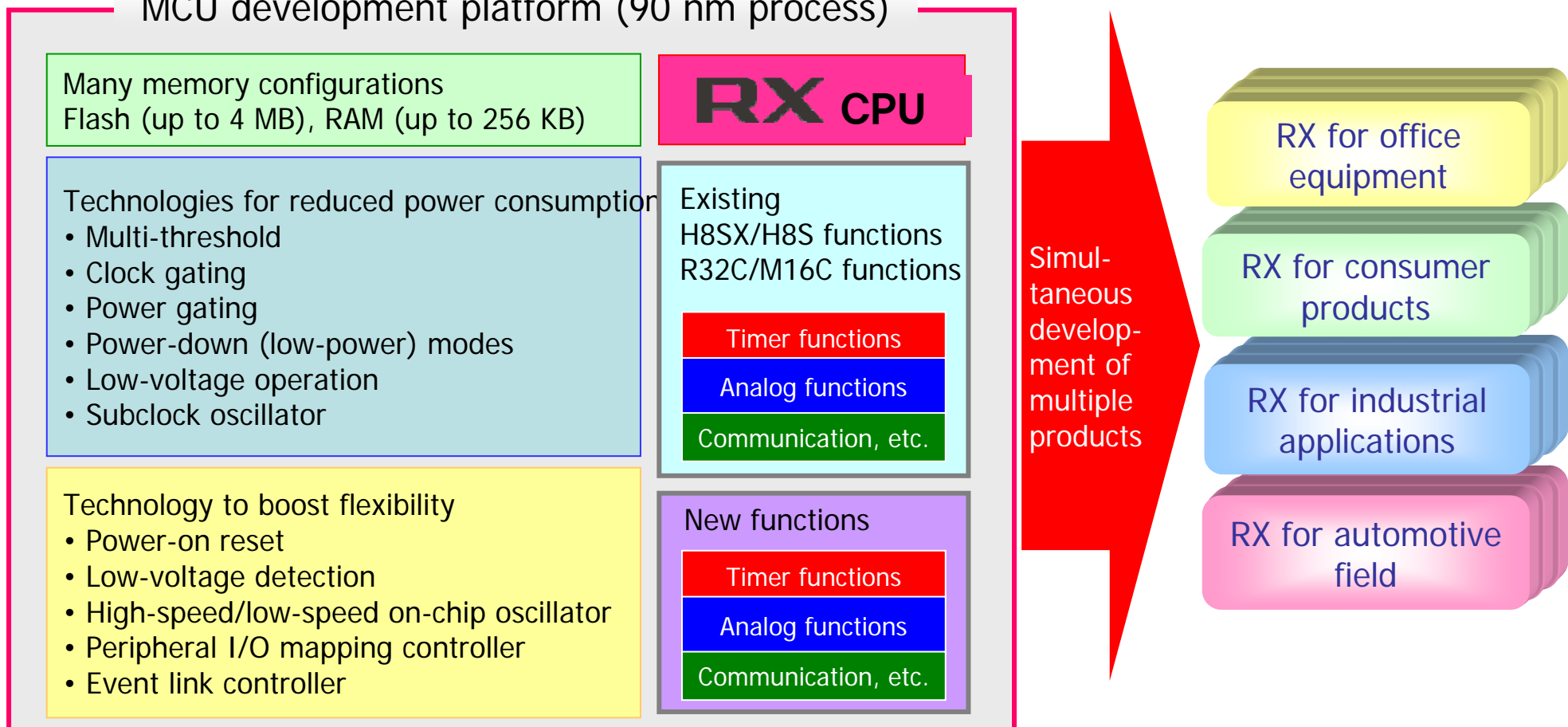
Many package configurations

- LQFP packages with low to high pin counts
- Lineup of smaller and higher density LGA and FBGA packages

Features of MCU Development Platform

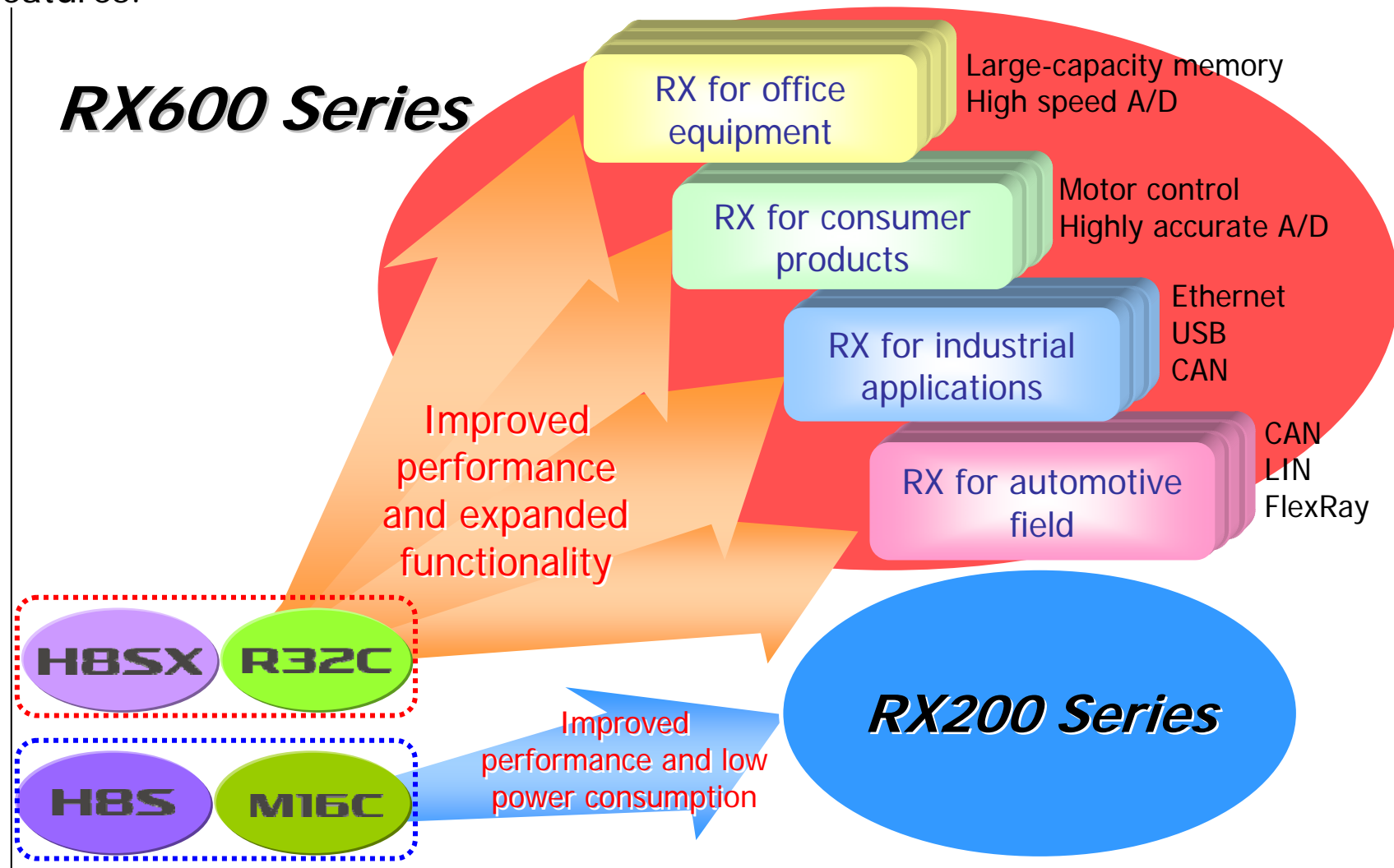
- Ability to develop products with optimal combinations of memory configuration, peripheral functions, low power consumption technology, and improved flexibility
- Improved design efficiency for faster development and easier rollout of new product versions

MCU development platform (90 nm process)



Planned Evolution of RX Family

- Retain the usability of current products while boosting performance and expanding features.



RX600 Series Product Plan



- Plans call for samples of the first products in the RX600 Series, for office equipment, consumer, and industrial applications, to be released in the 2nd quarter of 2009.
- Versions with peripheral functions such as USB module (host/function), CAN interface, Ethernet module, and timers for motor control will be released one after another thereafter.

1st products currently under development

RX CPU: Multiplier, divider, multiply-accumulate unit, single-precision floating-point unit

Fast, large-capacity memory

High-speed interrupts

General timers

DMAC

Serial I/F

Watchdog timer

A/D converter

On-chip debugger

Contents



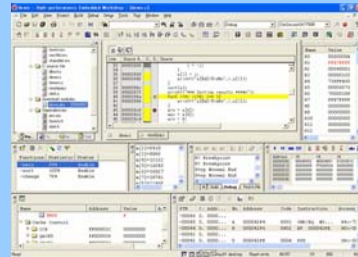
1. Concept of RX Family
2. Features of RX CPU
 - 2-1 Improved Performance
 - 2-2 Improved Code Efficiency
 - 2-3 Enhanced Flexibility
3. RX Family Product Evolution
 - 3-1 Lower Power Consumption
 - 3-2 Large-Capacity, High-Speed Flash Memory
 - 3-3 MCU Development Platform
4. Development Environment
5. Summary

RX Development Environment



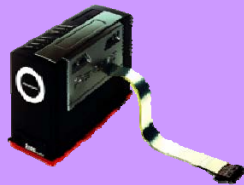
- We will support a development environment similar to that for existing products so that customers can use it with confidence.

Renesas integrated development environment



Allows program development with same flexibility as existing MCUs.

Full-spec emulator



On-chip emulators



Customers can choose an emulator to match the scale of the project.

Starter kit (RSK)



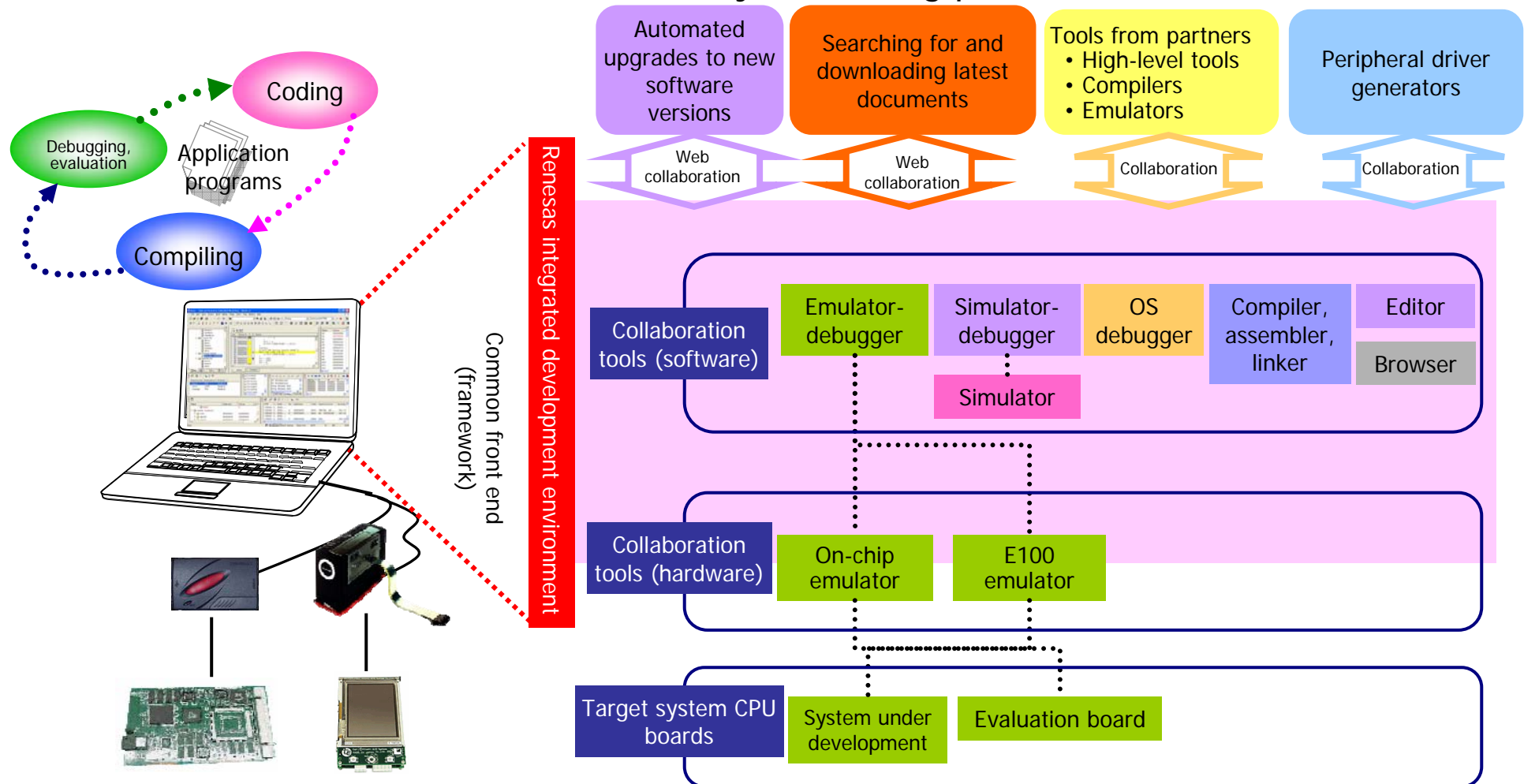
Ideal for learning about MCUs, these kits include a CPU board and a development environment in a single package.

Tools from Renesas partners

Third-party tools for the M16C, M32C, R32C, H8S, and H8SX will continue to be available.

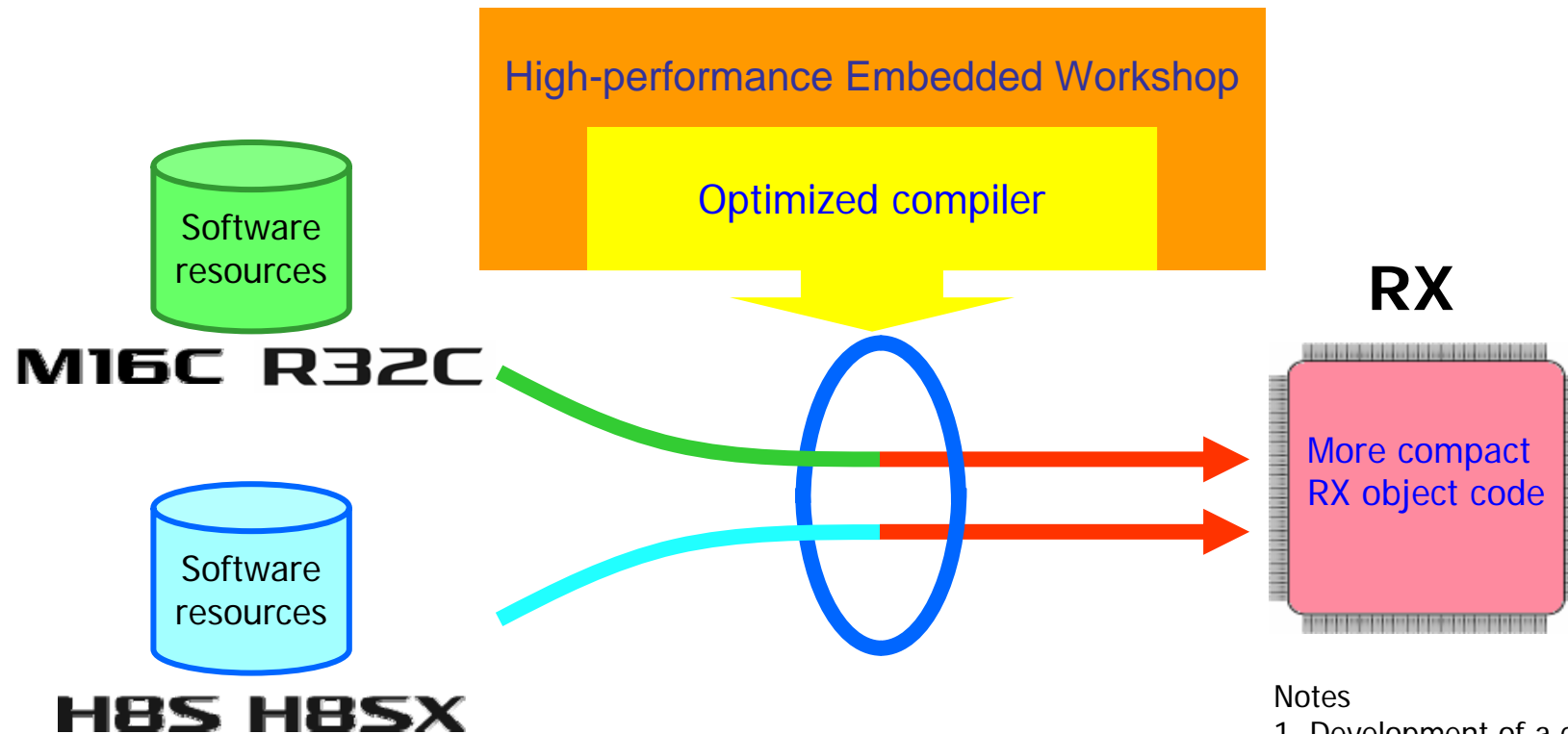
Continuation of Renesas Tool System

- Existing tools from partners can continue to be used along with the Renesas integrated development environment.
- Internet-based services, driver generation tools for peripheral functions, etc., will continue to be available in the same way as existing products.



Software Continuity

- Ability to reuse program code written in the C language
- Ability to reuse existing resources written in assembly language¹
- Retention of existing OS API²



Notes

1. Development of a compiler is planned.
2. Application Program Interface

Contents



1. Concept of RX Family
2. Features of RX CPU
 - 2-1 Improved Performance
 - 2-2 Improved Code Efficiency
 - 2-3 Enhanced Flexibility
3. RX Family Product Evolution
 - 3-1 Lower Power Consumption
 - 3-2 Large-Capacity, High-Speed Flash Memory
 - 3-3 MCU Development Platform
4. Development Environment
5. Summary

Summary



- RX CPU targets
 - Maximum operating frequency: 200 MHz
 - Processing performance: 1.25 MIPS/MHz
 - Code efficiency: 30% better than current products
 - Power consumption: CPU current consumption 0.03 mA/MHz
 - By meeting these targets we will create a world-top-class CPU.

- Samples of the first products built around the RX, the RX600 Series, will be released in the 2nd quarter of 2009.

- A development environment comprising tools for program development will be released at the same time as the samples.



Renesas Technology Corp.

©2008. Renesas Technology Corp., All rights reserved.