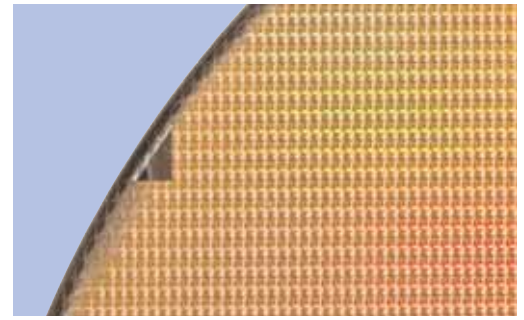


# RS08 Core



## Overview

The RS08 core is a reduced version of the very popular S08 central processing unit (CPU) that has been specifically crafted to be more efficient and cost effective for small memory size microcontrollers. This is an ideal solution for such emerging applications as simple electro-mechanical devices that are migrating to fully solid-state electronic operation or portable devices that have evolved into smaller or even disposable versions.

To create the RS08 core, the engineers at Freescale have removed from the S08 core those functions that are not required for such applications and have enhanced the remaining operations to improve the efficiency of extremely small controllers with a maximum of 16K addressable space. Thirty percent smaller than the S08 core, the RS08 is optimized for cost-effective, space efficient operation in ultra-small 3 x 3 mm 6-pin QFN packaging, with fewer interconnects for improved board level reliability.

The RS08 employs a von Neumann architecture with shared program and data bus, and although it is not 100 percent compatible with the S08 core, the instruction set is very similar to help make a transition from the S08 core to the RS08 core as simple as possible. The RS08 core uses the same bus structure to make memory and peripheral module reuse possible, and it has compatible debug interface hardware. Short and Tiny addressing modes allow for more efficient access and manipulation of the most commonly used variables and registers.

The RS08 core deepens the Freescale 8-bit portfolio, and it will offer a starting point for a performance roadmap that will provide pin-for-pin compatible 8-bit and 32-bit devices that share peripherals and a common set of development tools. It will allow mechanical and electronic product developers to come up with the totally unexpected, creating new, cost-effective electronic applications that have no precedent and disposable products with surprisingly sophisticated functionality.

## Major RS08 Core Changes from S08 Core

- > Modified working register set
  - Eliminated H and SP
  - Reduced the number of bits in the CCR
  - Single level subroutine call with software support for nesting interrupts
- > Instruction set changes
  - Eliminated less frequently used instructions such as:
    - › Multiply and Divide
    - › Nibble swap, arithmetic shift and some branches
- > Introduced
  - Tiny and Short addressing mode
  - Shadow register for program counter (SPC) to allow subroutine calls
  - Shadow PC transfer instructions: SHA, SLA
- > Interrupt function changes
  - RESET sequence begins executing from \$3FFD (highest memory location -3)
  - One interrupt pending register exists for polling
  - Keyboard and Real Time interrupt wake-up capability from STOP or WAIT mode
- > Paged memory scheme
  - Allows access to entire memory map with efficient direct page addressing
  - Program execution and RAM access do not require paging
- > Enhanced Background Debug commands to support RS08 architecture

## RS08 Instructions with Reduced Addressing as Compared to S08

- > The Short addressing is available with LDA, STA and CLR
  - Single byte instructions for lower 32-byte addressable space
- > The Tiny addressing is available with ADD, SUB, INC and DEC
  - Single byte instruction for lower 16-byte addressable space
- > Benefits of Tiny and Short addressing
  - Reduces code size by 1 byte (50 percent less)\*
  - Some instructions execute in 1 less cycle (up to 33 percent less)\*

\*Improvements in code efficiency and execution time will be application dependent; applications using limited RAM and peripherals will frequently benefit most.

## Sample Applications

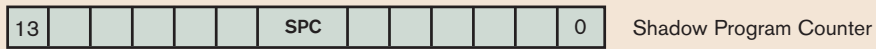
- > Ultra-bright LED
- > Low-end thermal controlled brushless DC (BLDC) fan
- > Low-end remote control in light dimmer, light switch or electric fan
- > Toaster oven
- > Low-end microwave oven
- > AC voltage line monitor
- > External watchdog



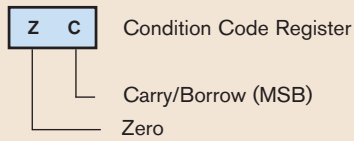
Accumulator



Program Counter



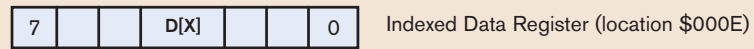
Shadow Program Counter



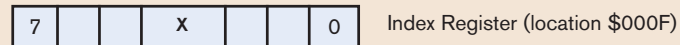
Condition Code Register

Carry/Borrow (MSB)

Zero



Indexed Data Register (location \$000E)



Index Register (location \$000F)



PAGESEL Register (location \$001F)

**Learn More:** For more information about Freescale products, please visit [www.freescale.com](http://www.freescale.com).