Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	RS08
Core Size	8-Bit
Speed	10MHz
Connectivity	-
Peripherals	LVD, POR, WDT
Number of I/O	4
Program Memory Size	1KB (1K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	63 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	8-SOIC (0.154", 3.90mm Width)
Supplier Device Package	8-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9rs08ka1csc

Appendix A
Electrical Characteristics

Appendix B
Ordering Information and Mechanical Drawings

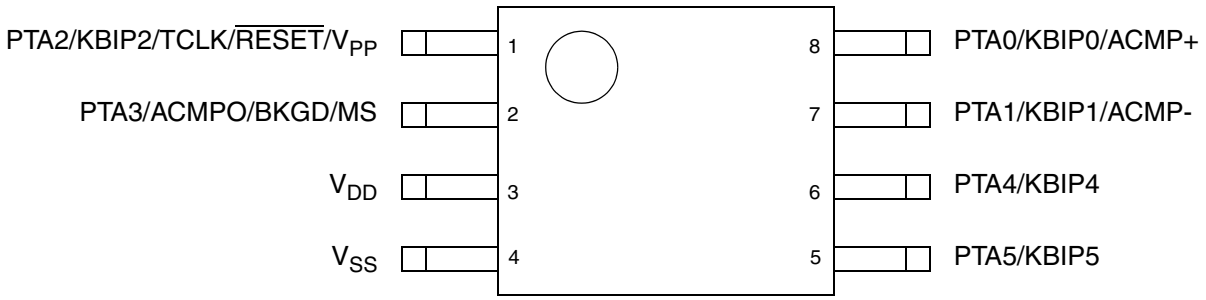


Figure 2-3. MC9RS08KA2 Series in 8-Pin Narrow Body SOIC

2.3 Recommended System Connections

Figure 2-4 shows reference connection for background debug and Flash programming.

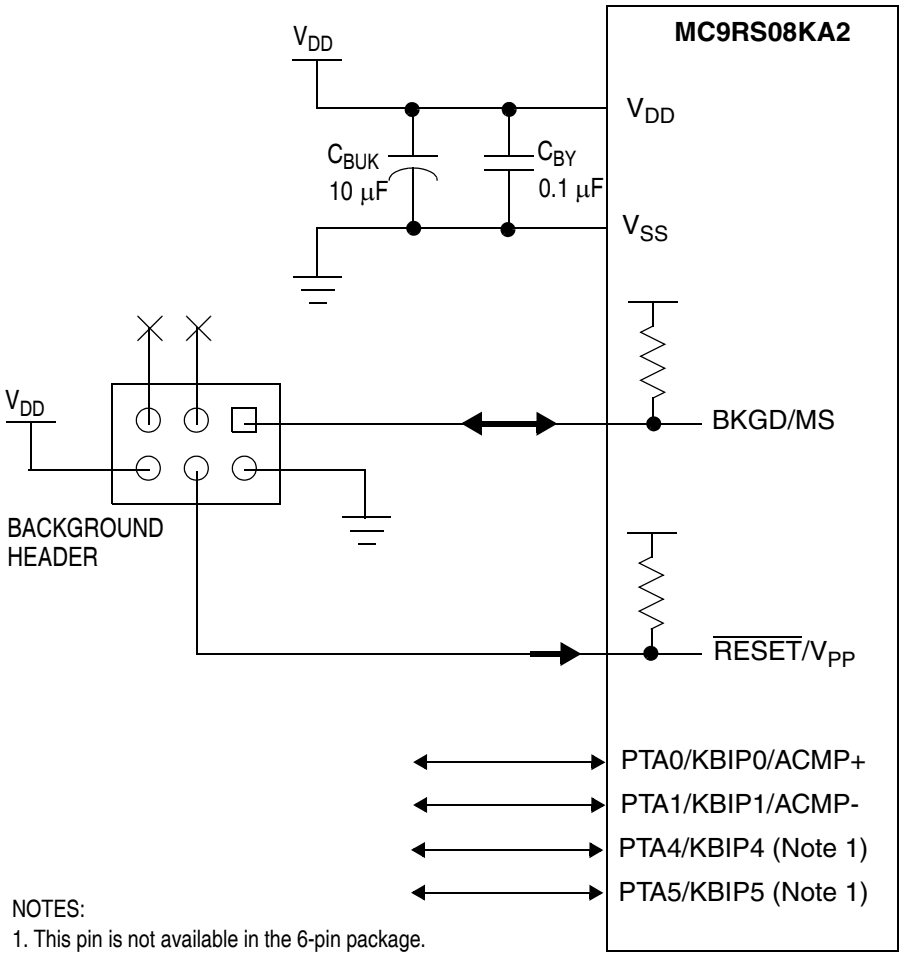


Figure 2-4. Reference System Connection Diagram

2.4 Pin Detail

This section provides a detailed description of system connections.

4.2 Unimplemented Memory

Attempting to access either data or an instruction at an unimplemented memory address will cause reset.

4.3 Indexed/Indirect Addressing

Register D[X] and register X together perform the indirect data access. Register D[X] is mapped to address $000E. Register X is located in address $000F. The 8-bit register X contains the address that is used when register D[X] is accessed. Register X is cleared to zero upon reset. By programming register X, any location on the first page ($0000–$00FF) can be read/written via register D[X]. Figure 4-2 shows the relationship between D[X] and register X. For example, in HC08/S08 syntax *lda ,x* is comparable to *lda D[X]* in RS08 coding when register X has been programmed with the index value.

The physical location of $000E is in RAM. Accessing the location through D[X] returns $000E RAM content when register X contains $0E. The physical location of $000F is register X, itself. Reading the location through D[X] returns register X content; writing to the location modifies register X.

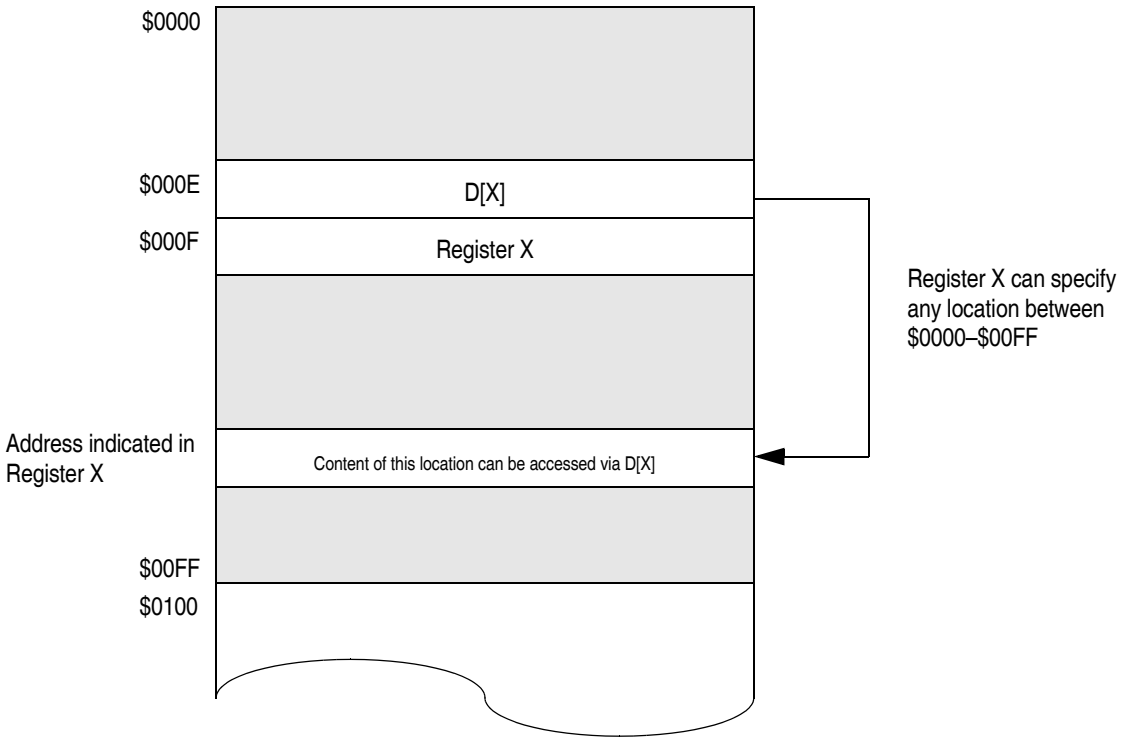


Figure 4-2. Indirect Addressing Registers

4.4 RAM and Register Addresses and Bit Assignments

The fast access RAM area can be accessed by instructions using tiny, short, and direct addressing mode instructions. For tiny addressing mode instructions, the operand is encoded along with the opcode to a single byte.

- Up to 1000 program/erase cycles at typical voltage and temperature
- Security feature for Flash

4.6.2 Flash Programming Procedure

Programming of Flash memory is done on a row basis. A row consists of 64 consecutive bytes starting from addresses $3X00, $3X40, $3X80, or $3XC0. Use the following procedure to program a row of Flash memory:

1. Apply external V_{PP}.
2. Set the PGM bit. This configures the memory for program operation and enables the latching of address and data for programming.
3. Write any data to any Flash location, via the high page accessing window $00C0–$00FF, within the address range of the row to be programmed. (Prior to the data writing operation, the PAGESEL register must be configured correctly to map the high page accessing window to the corresponding Flash row).
4. Wait for a time, t_{nvs}.
5. Set the HVEN bit.
6. Wait for a time, t_{pgs}.
7. Write data to the Flash location to be programmed.
8. Wait for a time, t_{prog}.
9. Repeat steps 7 and 8 until all bytes within the row are programmed.
10. Clear the PGM bit.
11. Wait for a time, t_{nvh}.
12. Clear the HVEN bit.
13. After time, t_{rcv}, the memory can be accessed in read mode again.
14. Remove external V_{PP}.

This program sequence is repeated throughout the memory until all data is programmed.

NOTE

Flash memory cannot be programmed or erased by software code executed from Flash locations. To program or erase Flash, commands must be executed from RAM or BDC commands. User code should not enter wait or stop during erase or program sequence.

These operations must be performed in the order shown; other unrelated operations may occur between the steps.

4.6.3 Flash Mass Erase Operation

Use the following procedure to mass erase the entire Flash memory:

1. Apply external V_{PP}.
2. Set the MASS bit in the Flash control register.

and ACMP are still available to wake the CPU from wait or stop mode. It is the responsibility of the user application to poll the corresponding module to determine the source of wakeup.

Each wakeup source of the module is associated with a corresponding interrupt enable bit. If the bit is disabled, the interrupt source is gated, and that particular source cannot wake the CPU from wait or stop mode. However, the corresponding interrupt flag will still be set to indicate that an external wakeup event has occurred.

The system interrupt pending register (SIP1) indicates the status of the system pending interrupt. When the read-only bit of the SIP1 is enabled, it shows there is a pending interrupt to be serviced from the indicated module. Writing to the register bit has no effect. The pending interrupt flag will be cleared automatically when the all corresponding interrupt flags from the indicated module are cleared.

5.6 Low-Voltage Detect (LVD) System

The MC9RS08KA2 Series includes a system to protect against low voltage conditions in order to protect memory contents and control MCU system states during supply voltage variations. The system is comprised of a power-on reset (POR) circuit and an LVD circuit with a predefined trip voltage. The LVD circuit is enabled with LVDE in SPMSC1. The LVD is disabled upon entering stop mode unless LVDSE is set in SPMSC1. If LVDSE and LVDE are both set, the current consumption in stop with the LVD enabled will be greater.

5.6.1 Power-On Reset Operation

When power is initially applied to the MCU, or when the supply voltage drops below the V_{POR} level, the POR circuit will cause a reset condition. As the supply voltage rises, the LVD circuit will hold the MCU in reset until the supply has risen above the V_{LVD} level. Both the POR bit and the LVD bit in SRS are set following a POR.

5.6.2 LVD Reset Operation

The LVD can be configured to generate a reset upon detection of a low voltage condition by setting LVDRE to 1. After an LVD reset has occurred, the LVD system will hold the MCU in reset until the supply voltage has risen above the level V_{LVD}. The LVD bit in the SRS register is set following either an LVD reset or POR.

5.6.3 LVD Interrupt Operation

When a low voltage condition is detected and the LVD circuit is configured using SPMSC1 for interrupt operation (LVDE set, LVDIE set, and LVDRE clear), LVDF in SPMSC1 will be set and an LVD interrupt request will occur.

5.7 Real-Time Interrupt (RTI)

The real-time interrupt function can be used to generate periodic interrupts. The RTI is driven from either the 1-kHz internal clock reference or the trimmed 32-kHz internal clock reference from the ICS module. The 32-kHz internal clock reference is divided by 32 by the RTI logic to produce a trimmed 1-kHz clock

5.8.5 System Power Management Status and Control 1 Register (SPMSC1)

This high page register contains status and control bits to support the low voltage detect function, and to enable the bandgap voltage reference for use by the ACMP and the LVD module.

	7	6	5	4	3	2	1	0
R	LVDF	0	LVDIE	LVDRE[1]	LVDSE	LVDE[1]	0	BGBE
W		LVDACK						
Reset:	0	0	0	1	1	1	0	0

= Unimplemented or Reserved

1 This bit can be written only one time after reset. Additional writes are ignored.

Figure 5-6. System Power Management Status and Control 1 Register (SPMSC1)

Table 5-8. SPMSC1 Register Field Descriptions

Field	Description
7 LVDF	**Low-Voltage Detect Flag** — Provided LVDE = 1, this read-only status bit indicates a low-voltage detect event.
6 LVDACK	**Low-Voltage Detect Acknowledge** — This write-only bit is used to acknowledge low voltage detection errors (write 1 to clear LVDF). Reads always return 0.
5 LVDIE	**Low-Voltage Detect Interrupt Enable** — This bit enables hardware interrupt requests for LVDF. 0 Hardware interrupt disabled (use polling). 1 Request a hardware interrupt when LVDF = 1.
4 LVDRE	**Low-Voltage Detect Reset Enable** — This write-once bit enables low-voltage detect events to generate a hardware reset (provided LVDE = 1). 0 LVDF does not generate hardware resets. 1 Force an MCU reset when LVDF = 1.
3 LVDSE	**Low-Voltage Detect Stop Enable** — Provided LVDE = 1, this read/write bit determines whether the low-voltage detect function operates when the MCU is in stop mode. 0 Low-voltage detect disabled during stop mode. 1 Low-voltage detect enabled during stop mode.
2 LVDE	**Low-Voltage Detect Enable** — This write-once bit enables low-voltage detect logic and qualifies the operation of other bits in this register. 0 LVD logic disabled. 1 LVD logic enabled.
0 BGBE	**Bandgap Buffer Enable** — This bit enables an internal buffer for the bandgap voltage reference for use by the ACMP module on one of its internal channels. 0 Bandgap buffer disabled. 1 Bandgap buffer enabled.

7.4 Functional Description

This on-chip peripheral module is called a keyboard interrupt (KBI) module because it was originally designed to simplify the connection and use of row-column matrices of keyboard switches. However, these inputs are also useful as extra external interrupt inputs and as an external means of waking the MCU from stop or wait low-power modes.

The KBI module allows its pins to act as additional interrupt sources. Writing to the KBIPEn bits in the keyboard interrupt pin enable register (KBIPE) independently enables or disables each KBI pin. Each KBI pin can be configured as edge sensitive or edge and level sensitive based on the KBMOD bit in the keyboard interrupt status and control register (KBISC). Edge sensitive can be software programmed to be either falling or rising; the level can be either low or high. The polarity of the edge or edge and level sensitivity is selected using the KBEDGn bits in the keyboard interrupt edge select register (KBIES).

Synchronous logic is used to detect edges. Prior to detecting an edge, enabled keyboard inputs must be at the deasserted logic level. A falling edge is detected when an enabled keyboard input signal is seen as a logic 1 (the deasserted level) during one bus cycle and then a logic 0 (the asserted level) during the next cycle. A rising edge is detected when the input signal is seen as a logic 0 during one bus cycle and then a logic 1 during the next cycle.

7.4.1 Edge Only Sensitivity

A valid edge on an enabled KBI pin will set KBF in KBISC. If KBIE in KBISC is set, an interrupt request will be presented to the CPU. Clearing of KBF is accomplished by writing a 1 to KBACK in KBISC.

7.4.2 Edge and Level Sensitivity

A valid edge or level on an enabled KBI pin will set KBF in KBISC. If KBIE in KBISC is set, an interrupt request will be presented to the CPU. Clearing of KBF is accomplished by writing a 1 to KBACK in KBISC, provided all enabled keyboard inputs are at their deasserted levels. KBF will remain set if any enabled KBI pin is asserted while attempting to clear by writing a 1 to KBACK.

7.4.3 KBI Pullup/Pulldown Device

The KBI pins does not automatically configure an internal pullup/pulldown device when a KBI pin is enabled. An internal pull device can be used by configuring the associated I/O port pull device enable register (PTAPE) and pullup/pulldown control register (PTAPUD).

7.4.4 KBI Initialization

When a keyboard interrupt pin is first enabled, it is possible to get a false keyboard interrupt flag. To prevent a false interrupt request during keyboard initialization, the user should do the following:

1. Mask keyboard interrupts by clearing KBIE in KBISC.
2. If using internal pullup/pulldown device, configure the associated I/O port pullup/pulldown device.
3. Enable the KBI polarity by setting the appropriate KBEDGn bits in KBIES.
4. Enable the KBI pins by setting the appropriate KBIPEn bits in KBIPE.

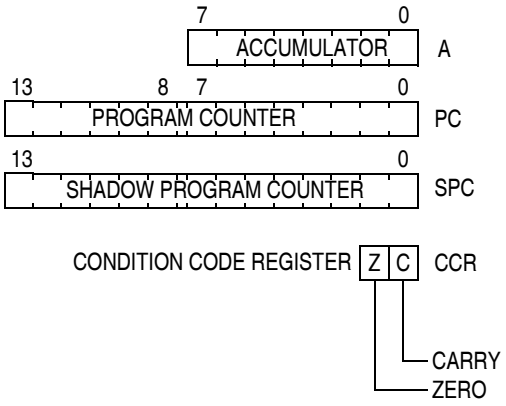


Figure 8-1. CPU Registers

In addition to the CPU registers, there are three memory mapped registers that are tightly coupled with the core address generation during data read and write operations. They are the indexed data register (D[X]), the index register (X), and the page select register (PAGESEL). These registers are located at $000E, $000F, and $001F, respectively.

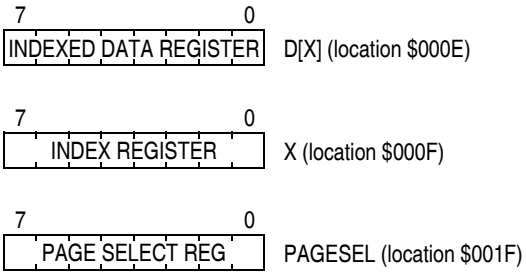


Figure 8-2. Memory Mapped Registers

8.2.1 Accumulator (A)

This general-purpose 8-bit register is the primary data register for RS08 MCUs. Data can be read from memory into A with a load accumulator (LDA) instruction. The data in A can be written into memory with a store accumulator (STA) instruction. Various addressing mode variations allow a great deal of flexibility in specifying the memory location involved in a load or store instruction. Exchange instructions allow values to be exchanged between A and SPC high (SHA) and also between A and SPC low (SLA).

Arithmetic, shift, and logical operations can be performed on the value in A as in ADD, SUB, RORA, INCA, DECA, AND, ORA, EOR, etc. In some of these instructions, such as INCA and LSLA, the value in A is the only input operand and the result replaces the value in A. In other cases, such as ADD and AND, there are two operands: the value in A and a second value from memory. The result of the arithmetic or logical operation replaces the value in A.

Some instructions, such as memory-to-memory move instructions (MOV), do not use the accumulator. DBNZ also relieves A because it allows a loop counter to be implemented in a memory variable rather than the accumulator.

During reset, the accumulator is loaded with $00.

dd	=	Low-order eight bits of a direct address $0000–$00FF (high byte assumed to be $00)
ii	=	One byte of immediate data
hh	=	High-order 6-bit of 14-bit extended address prefixed with 2-bit of 0
ll	=	Low-order byte of 14-bit extended address
rr	=	Relative offset

Source form

Everything in the source forms columns, *except expressions in italic characters,* is literal information which must appear in the assembly source file exactly as shown. The initial 3- to 5-letter mnemonic is always a literal expression. All commas, pound signs (#), parentheses, and plus signs (+) are literal characters.

n	—	Any label or expression that evaluates to a single integer in the range 0–7.
x	—	Any label or expression that evaluates to a single hexadecimal integer in the range $0–$F.
opr8i	—	Any label or expression that evaluates to an 8-bit immediate value.
opr4a	—	Any label or expression that evaluates to a Tiny address (4-bit value). The instruction treats this 4-bit value as the low order four bits of an address in the 16-Kbyte address space ($0000–$000F). This 4-bit value is embedded in the low order four bits in the opcode.
opr5a	—	Any label or expression that evaluates to a Short address (5-bit value). The instruction treats this 5-bit value as the low order five bits of an address in the 16-Kbyte address space ($0000–$001F). This 5-bit value is embedded in the low order 5 bits in the opcode.
opr8a	—	Any label or expression that evaluates to an 8-bit value. The instruction treats this 8-bit value as the low order eight bits of an address in the 16-Kbyte address space ($0000–$00FF).
opr16a	—	Any label or expression that evaluates to a 14-bit value. On the RS08 core, the upper two bits are always 0s. The instruction treats this value as an address in the 16-Kbyte address space.
rel	—	Any label or expression that refers to an address that is within –128 to +127 locations from the next address after the last byte of object code for the current instruction. The assembler will calculate the 8-bit signed offset and include it in the object code for this instruction.

Address modes

INH	=	Inherent (no operands)
IMD	=	Immediate to Direct (in MOV instruction)
IMM	=	Immediate
DD	=	Direct to Direct (in MOV instruction)
DIR	=	Direct
SRT	=	Short
TNY	=	Tiny
EXT	=	Extended
REL	=	8-bit relative offset

Chapter 9
Internal Clock Source (RS08ICSV1)

9.1 Introduction

The internal clock source (ICS) module provides clock source choices for the MCU. The module contains a frequency-locked loop (FLL) as a clock source that is controllable by an internal reference clock. The module can provide this FLL clock or the internal reference clock as a source for the MCU system clock, ICSOUT.

Whichever clock source is chosen, ICSOUT is passed through a bus clock divider (BDIV), which allows a lower final output clock frequency to be derived. ICSOUT is two times the bus frequency.

Figure 9-1 shows the MC9RS08KA2 Series block diagram with the ICS highlighted.

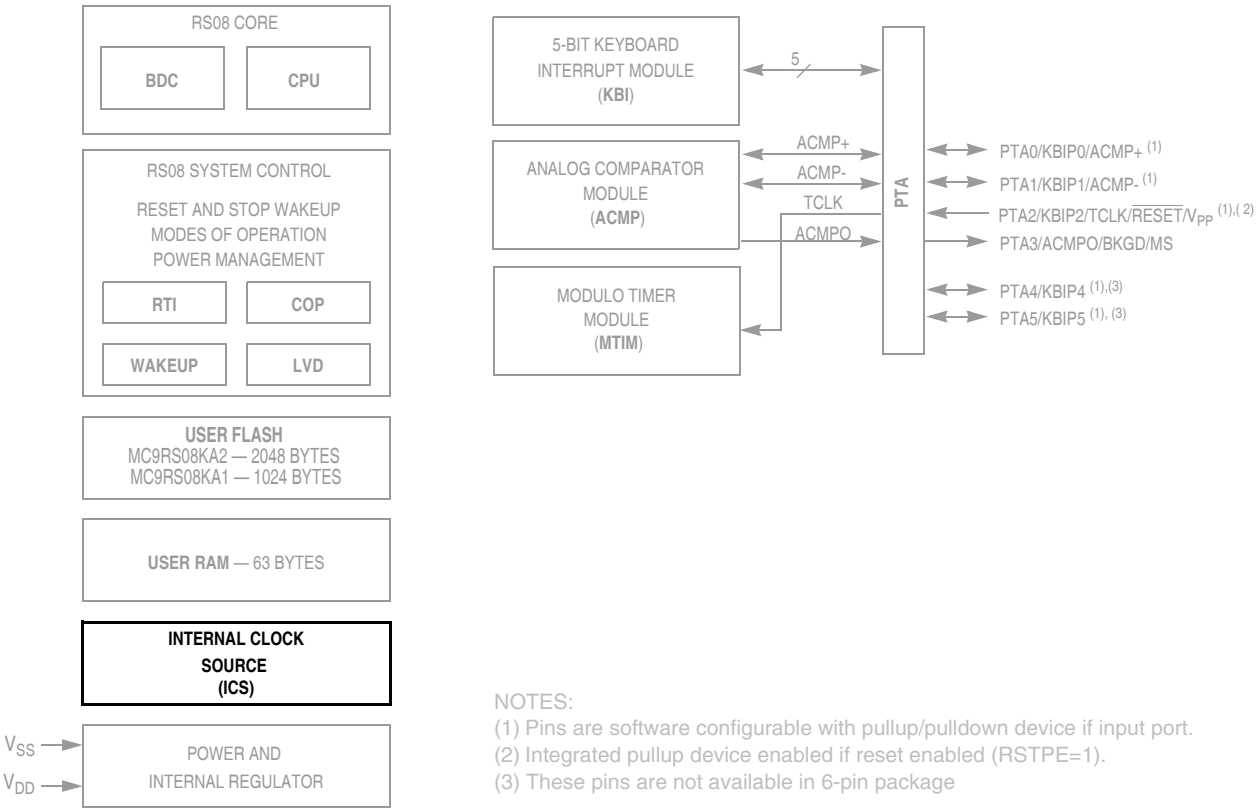


Figure 9-1. MC9RS08KA2 Series Block Diagram Highlighting ICS Block

9.4.6 Fixed Frequency Clock

The ICS provides the ICSFFCLK output which can be used as an additional clock source to a peripheral such as a timer, when the ICS is in FEI. ICSFFCLK is not a valid clock source for a peripheral when in either FBI or FBILP modes. ICSFFCLK is ICSRCLK divided by two.

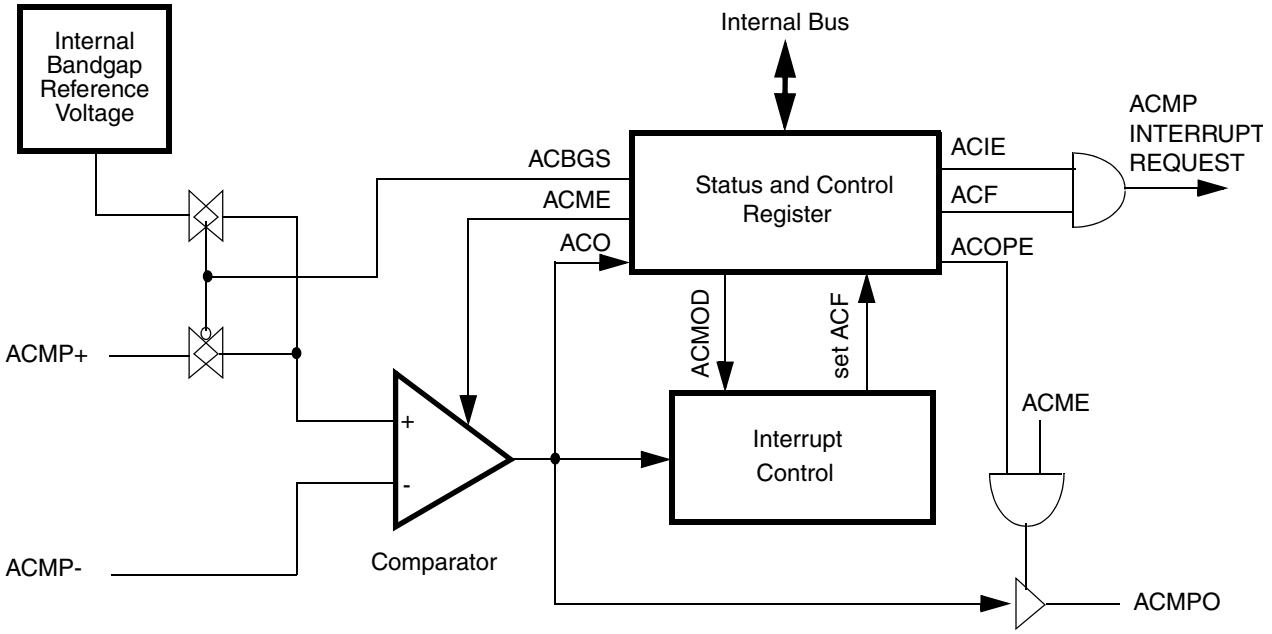


Figure 10-2. Analog Comparator (ACMP) Block Diagram

NOTE

Comparator inputs are high impedence analog pins which are sensitive to noise. Noisy VDD and/or pin toggling adjacent to the analog inputs may cause the comparator offset/hysteresis performance to exceed the specified values. Maximum source impedence is restricted to the value specified in Table A-6. To achieve maximum performance device is recommended to enter WAIT/STOP mode for ACMP measurement and adjacent pin toggling must be avoided.

11.1.3 Block Diagram

The block diagram for the modulo timer module is shown Figure 11-2.

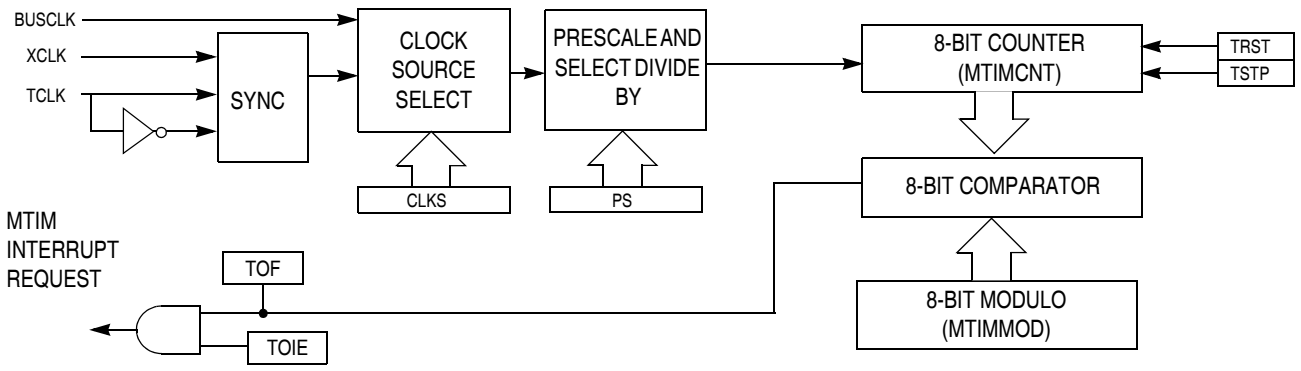


Figure 11-2. Modulo Timer (MTIM) Block Diagram

11.2 External Signal Description

The MTIM includes one external signal, TCLK, used to input an external clock when selected as the MTIM clock source. The signal properties of TCLK are shown in Table 11-1.

Table 11-1. Signal Properties

Signal	Function	I/O
TCLK	External clock source input into MTIM	I

The TCLK input must be synchronized by the bus clock. Also, variations in duty cycle and clock jitter must be accommodated. Therefore, the TCLK signal must be limited to one-fourth of the bus frequency.

The TCLK pin can be muxed with a general-purpose port pin. See the Pins and Connections chapter for the pin location and priority of this function.

11.3 Register Definition

Each MTIM includes four registers, which are summarized in Table 11-2:

- An 8-bit status and control register
- An 8-bit clock configuration register
- An 8-bit counter register
- An 8-bit modulo register

Refer to the direct-page register summary in the memory section of this data sheet for the absolute address assignments for all MTIM registers. This section refers to registers and control bits only by their names.

Table 11-2. MTIM Register Summary

Name		7	6	5	4	3	2	1	0
MTIMSC	R	TOF	TOIE	0	TSTP	0	0	0	0
	W			TRST					
MTIMCLK	R	0	0	CLKS		PS			
	W								
MTIMCNT	R	COUNT							
	W								
MTIMMOD	R	MOD							
	W								

11.3.1 MTIM Status and Control Register (MTIMSC)

MTIMSC contains the overflow status flag and control bits which are used to configure the interrupt enable, reset the counter, and stop the counter.

	7	6	5	4	3	2	1	0
R	TOF	TOIE	0	TSTP	0	0	0	0
W			TRST					
Reset:	0	0	0	1	0	0	0	0

Figure 11-3. MTIM Status and Control Register (MTIMSC)

Table 11-3. MTIMSC Field Descriptions

Field	Description
7 TOF	**MTIM Overflow Flag** — This read-only bit is set when the MTIM counter register overflows to $00 after reaching the value in the MTIM modulo register. Clear TOF by reading the MTIMSC register while TOF is set, then writing a 0 to TOF. TOF is also cleared when TRST is written to a 1 or when any value is written to the MTIMMOD register. 0 MTIM counter has not reached the overflow value in the MTIM modulo register. 1 MTIM counter has reached the overflow value in the MTIM modulo register.
6 TOIE	**MTIM Overflow Interrupt Enable** — This read/write bit enables MTIM overflow interrupts. If TOIE is set, then an interrupt is generated when TOF = 1. Reset clears TOIE. Do not set TOIE if TOF = 1. Clear TOF first, then set TOIE. 0 TOF interrupts are disabled. Use software polling. 1 TOF interrupts are enabled.
5 TRST	**MTIM Counter Reset** — When a 1 is written to this write-only bit, the MTIM counter register resets to $00 and TOF is cleared. Reading this bit always returns 0. 0 No effect. MTIM counter remains at current state. 1 MTIM counter is reset to $00.
4 TSTP	**MTIM Counter Stop** — When set, this read/write bit stops the MTIM counter at its current value. Counting resumes from the current value when TSTP is cleared. Reset sets TSTP to prevent the MTIM from counting. 0 MTIM counter is active. 1 MTIM counter is stopped.

Chapter 12
Development Support

12.1 Introduction

Development support systems in the RS08 family include the RS08 background debug controller (BDC).

The BDC provides a single-wire debug interface to the target MCU. This interface provides a convenient means for programming the on-chip FLASH and other nonvolatile memories. Also, the BDC is the primary debug interface for development and allows non-intrusive access to memory data and traditional debug features such as CPU register modify, breakpoint, and single-instruction trace commands.

In the RS08 Family, address and data bus signals are not available on external pins. Debug is done through commands fed into the target MCU via the single-wire background debug interface, including resetting the device without using a reset pin.

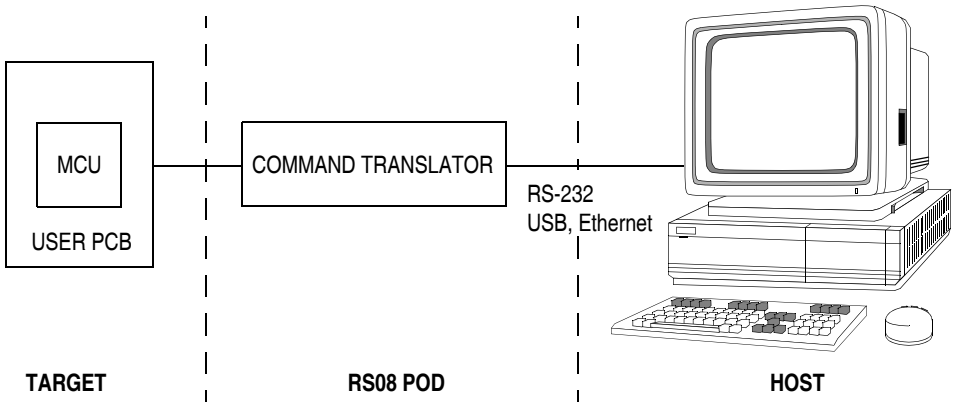


Figure 12-1. Connecting MCU to Host for Debugging

12.2 Features

Features of the RS08 background debug controller (BDC) include:

- Uses a single pin for background debug serial communications
- Non-intrusive of user memory resources; BDC registers are not located in the memory map
- SYNC command to determine target communications rate
- Non-intrusive commands allow access to memory resources while CPU is running user code without stopping applications
- Active background mode commands for CPU register access
- GO and TRACE1 commands
- BACKGROUND command can wake CPU from wait or stop modes

Table A-4. DC Characteristics (continued)
(Temperature Range = –40 to 85°C Ambient)

Parameter	Symbol	Min	Typical	Max	Unit
Input low voltage (V_{DD} > 2.3 V) (all digital inputs)	V_{IL}	—	—	0.30 × V_{DD}	V
Input low voltage (1.8 V ≤ V_{DD} ≤ 2.3 V) (all digital inputs)	V_{IL}	—	—	0.30 × V_{DD}	V
Input hysteresis (all digital inputs)	V_{hys}	0.06 × V_{DD}	—	—	V
Input leakage current (per pin) V_{In} = V_{DD} or V_{SS}, all input only pins	$\lvert I_{In} \rvert$	—	0.025	1.0	µA
High impedance (off-state) leakage current (per pin) V_{In} = V_{DD} or V_{SS}, all input/output	$\lvert I_{OZ} \rvert$	—	0.025	1.0	µA
Internal pullup/pulldown resistors[2] (all port pins)	R_{PU}	20	45	65	kΩ
Output high voltage (port A) I_{OH} = –5 mA (V_{DD} ≥ 4.5 V) I_{OH} = –3 mA (V_{DD} ≥ 3 V) I_{OH} = –2 mA (V_{DD} ≥ 1.8 V)	V_{OH}	V_{DD} – 0.8	—	— — —	V
Maximum total I_{OH} for all port pins	$\lvert I_{OHT} \rvert$	—	—	40	mA
Output low voltage (port A) I_{OL} = 5 mA (V_{DD} ≥ 4.5 V) I_{OL} = 3 mA (V_{DD} ≥ 3 V) I_{OL} = 2 mA (V_{DD} ≥ 1.8 V)	V_{OL}	—	—	0.8 0.8 0.8	V
Maximum total I_{OL} for all port pins	I_{OLT}	—	—	40	mA
dc injection current[3, 4, 5 6] V_{In} < V_{SS}, V_{In} > V_{DD} Single pin limit Total MCU limit, includes sum of all stressed pins	$\lvert I_{IC} \rvert$	—	—	 0.2 0.8	 mA mA
Input capacitance (all non-supply pins)	C_{In}	—	—	7	pF

1 This parameter is characterized and not tested on each device.

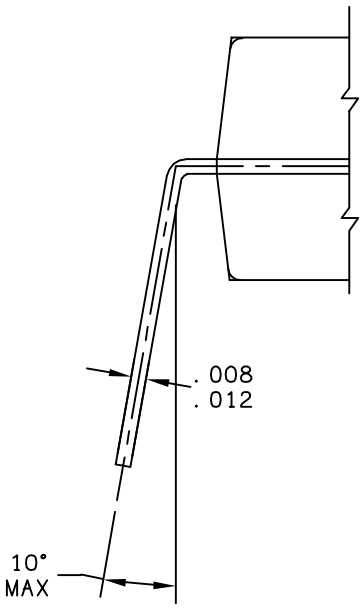
2 Measurement condition for pull resistors: V_{In} = V_{SS} for pullup and V_{In} = V_{DD} for pulldown.

3 All functional non-supply pins are internally clamped to V_{SS} and V_{DD} except the $\overline{\text{RESET}}$/V_{PP} which is internally clamped to V_{SS} only.

4 Input must be current limited to the value specified. To determine the value of the required current-limiting resistor, calculate resistance values for positive and negative clamp voltages, then use the larger of the two values.

5 Power supply must maintain regulation within operating V_{DD} range during instantaneous and operating maximum current conditions. If positive injection current (V_{In} > V_{DD}) is greater than I_{DD}, the injection current may flow out of V_{DD} and could result in external power supply going out of regulation. Ensure external V_{DD} load will shunt current greater than maximum injection current. This will be the greatest risk when the MCU is not consuming power. Examples are: if no system clock is present, or if clock rate is very low which would reduce overall power consumption.

6 This parameter is characterized and not tested on each device.



DETAIL "D"

MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE

TITLE:	DOCUMENT NO: 98ASB42420B	REV: N
8 LD PDIP	CASE NUMBER: 626-06	19 MAY 2005
	STANDARD: NON-JEDEC	

NOTES:

1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M – 1994.
2. ALL DIMENSIONS ARE IN INCHES.
3. 626-03 TO 626-06 OBSOLETE. NEW STANDARD 626-07.
4. DIMENSION TO CENTER OF LEAD WHEN FORMED PARALLEL.
5. PACKAGE CONTOUR OPTIONAL (ROUND OR SQUARE CONERS).

STYLE 1:

PIN	1.	AC IN	5.	GROUND
	2.	DC + IN	6.	OUTPUT
	3.	DC – IN	7.	AUXILIARY
	4.	AC IN	8.	VCC

MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE

TITLE:	DOCUMENT NO: 98ASB42420B	REV: N
8 LD PDIP	CASE NUMBER: 626-06	19 MAY 2005
	STANDARD: NON-JEDEC	