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	2KB (2K 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/mc9rs08ka2cscr

4.6.4 Security 31
4.7 Flash Registers and Control Bits 32
4.7.1 Flash Options Register (FOPT and NVOPT) 32
4.7.2 Flash Control Register (FLCR) 33
4.8 Page Select Register (PAGESEL) 33

Chapter 5
Resets, Interrupts, and General System Control

5.1 Introduction 35
5.2 Features 35
5.3 MCU Reset 35
5.4 Computer Operating Properly (COP) Watchdog 36
5.5 Interrupts 36
5.6 Low-Voltage Detect (LVD) System 37
5.6.1 Power-On Reset Operation 37
5.6.2 LVD Reset Operation 37
5.6.3 LVD Interrupt Operation 37
5.7 Real-Time Interrupt (RTI) 37
5.8 Reset, Interrupt, and System Control Registers and Control Bits 38
5.8.1 System Reset Status Register (SRS) 38
5.8.2 System Options Register (SOPT) 39
5.8.3 System Device Identification Register (SDIDH, SDIDL) 40
5.8.4 System Real-Time Interrupt Status and Control Register (SRTISC) 41
5.8.5 System Power Management Status and Control 1 Register (SPMSC1) 43
5.8.6 System Interrupt Pending Register (SIP1) 44

Chapter 6
Parallel Input/Output Control

6.1 Pin Behavior in Low-Power Modes 46
6.2 Parallel I/O Registers 46
6.2.1 Port A Registers 46
6.3 Pin Control Registers 47
6.3.1 Port A Pin Control Registers 47
6.3.1.1 Internal Pulling Device Enable 47
6.3.1.2 Pullup/Pulldown Control 48
6.3.1.3 Output Slew Rate Control Enable 48

Chapter 7
Keyboard Interrupt (RS08KBIV1)

7.1 Introduction 51
7.1.1 Features 51
7.1.2 Modes of Operation 52
7.1.2.1 Operation in Wait Mode 52
7.1.2.2 Operation in Stop Mode 52
7.1.2.3 Operation in Active Background Mode 52

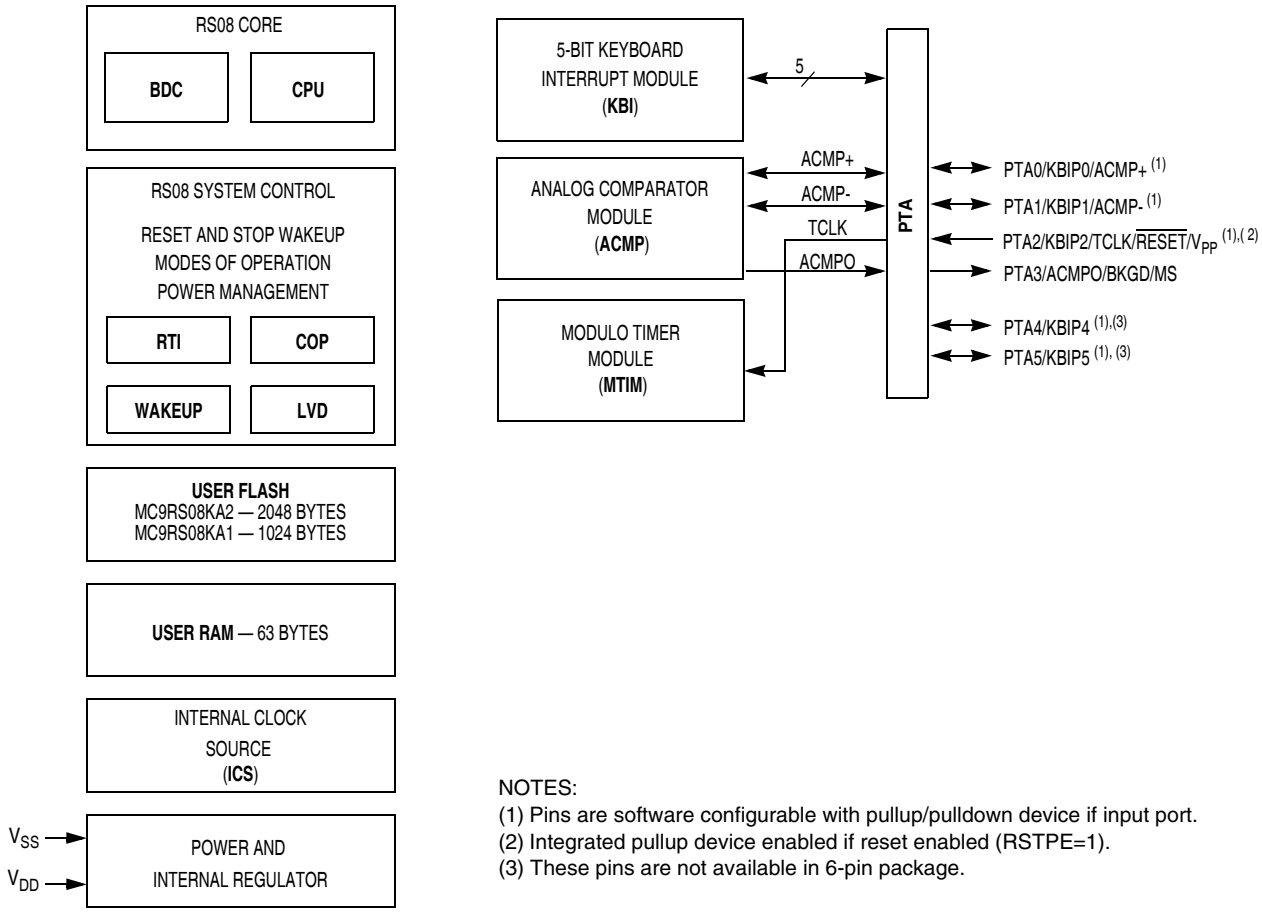
Chapter 1
MC9RS08KA2 Series Device Overview

1.1 Overview

The MC9RS08KA2 Series microcontroller unit (MCU) is an extremely low-cost, small pin count device for home appliances, toys, and small geometry applications. This device is composed of standard on-chip modules including, a very small and highly efficient RS08 CPU core, 63 bytes RAM, 2K bytes Flash, an 8-bit modulo timer, keyboard interrupt, and analog comparator. The device is available in small 6- and 8-pin packages.

1.2 MCU Block Diagram

The block diagram, Figure 1-1, shows the structure of the MC9RS08KA2 Series MCU.



NOTES:
(1) Pins are software configurable with pullup/pulldown device if input port.
(2) Integrated pullup device enabled if reset enabled (RSTPE=1).
(3) These pins are not available in 6-pin package.

Figure 1-1. MC9RS08KA2 Series Block Diagram

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.

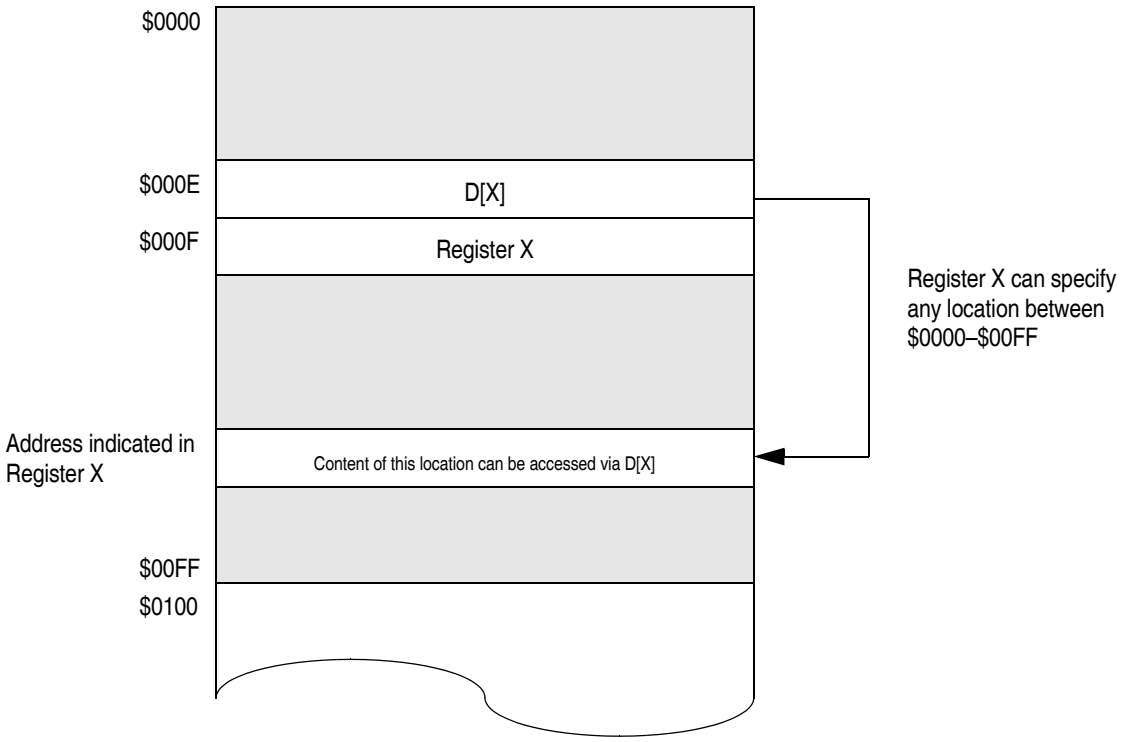


Figure 4-2. Indirect Addressing Registers

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.

3. Write any data to any Flash location, via the high page accessing window $00C0–$00FF. (Prior to the data writing operation, the PAGESEL register must be configured correctly to map the high page accessing window to the any Flash locations).
4. Wait for a time, t_{nvs}.
5. Set the HVEN bit.
6. Wait for a time t_{me}.
7. Clear the MASS bit.
8. Wait for a time, t_{nvh1}.
9. Clear the HVEN bit.
10. After time, t_{rcv}, the memory can be accessed in read mode again.
11. Remove external V_{PP}.

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 an erase or program sequence.

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

4.6.4 Security

The MC9RS08KA2 Series includes circuitry to help prevent unauthorized access to the contents of Flash memory. When security is engaged, Flash is considered a secure resource. The RAM, direct-page registers, and background debug controller are considered unsecured resources. Attempts to access a secure memory location through the background debug interface, or whenever BKGDPE is set, are blocked (reads return all 0s).

Security is engaged or disengaged based on the state of a nonvolatile register bit (SECD) in the FOPT register. During reset, the contents of the nonvolatile location NVOPT are copied from Flash into the working FOPT register in high-page register space. A user engages security by programming the NVOPT location, which can be done at the same time the Flash memory is programmed. Notice the erased state (SECD = 1) makes the MCU unsecure. When SECD in NVOPT is programmed (SECD = 0), next time the device is reset via POR, internal reset, or external reset, security is engaged. In order to disengage security, mass erase must be performed via BDM commands and followed by any reset.

The separate background debug controller can still be used for registers and RAM access. Flash mass erase is possible by writing to the Flash control register that follows the Flash mass erase procedure listed in Section 4.6.3, “Flash Mass Erase Operation,” via BDM commands.

Security can always be disengaged through the background debug interface by following these steps:

1. Mass erase Flash via background BDM commands or RAM loaded program.
2. Perform reset and the device will boot up with security disengaged.

5.8.6 System Interrupt Pending Register (SIP1)

This high page register contains status of the pending interrupt from the modules.

	7	6	5	4	3	2	1	0
R	0	0	0	KBI	ACMP	MTIM	RTI	LVD
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 5-7. System Interrupt Pending Register (SIP1)

Table 5-9. SIP1 Register Field Descriptions

Field	Description
4 KBI	**Keyboard Interrupt Pending** — This read-only bit indicates there is a pending interrupt from the KBI module. Clearing the KBF flag of the KBISC register clears this bit. Reset also clears this bit. 0 There is no pending KBI interrupt; i.e., KBF flag and/or KBIE bit is cleared. 1 There is a pending KBI interrupt; i.e., KBF flag and KBIE bit are set.
3 ACMP	**Analog Comparator Interrupt Pending** — This read-only bit indicates there is a pending interrupt from the ACMP module. Clearing the ACF flag of the ACMPSC register clears this bit. Reset also clears this bit. 0 There is no pending ACMP interrupt; i.e., ACF flag and/or ACIE bit is cleared. 1 There is a pending a ACMP interrupt; i.e., ACF flag and ACIE bit are set.
2 MTIM	**Modulo Timer Interrupt Pending** — This read-only bit indicates there is a pending interrupt from the MTIM module. Clearing the TOF flag of the MTIMSC register clears this bit. Reset also clears this bit. 0 There is no pending MTIM interrupt; i.e., TOF flag and/or TOIE bit is cleared. 1 There is a pending MTIM interrupt; i.e., TOF flag and TOIE bit are set.
1 RTI	**Real-Time Interrupt Pending** — This read-only bit indicates there is a pending interrupt from the RTI. Clearing the RTIF flag of the SRTISC register clears this bit. Reset also clears this bit. 0 There is no pending RTI interrupt; i.e., RTIF flag and/or RTIE bit is cleared. 1 There is a pending RTI interrupt; i.e., RTIF flag and RTIE bit are set.
0 LVD	**Low-Voltage Detect Interrupt Pending** — This read-only bit indicates there is a pending interrupt from the low voltage detect module. Clearing the LVDF flag of the SPMSC1 register clears this bit. Reset also clears this bit. 0 There is no pending LVD interrupt; i.e., LVDF flag and/or LVDE bit is cleared. 1 There is a pending LVD interrupt; i.e., LVDF flag, LVDIE, and LVDE bits are set.

8.2.2 Program Counter (PC)

The program counter is a 14-bit register that contains the address of the next instruction or operand to be fetched.

During normal execution, the program counter automatically increments to the next sequential memory location each time an instruction or operand is fetched. Jump, branch, and return operations load the program counter with an address other than that of the next sequential location. This is called a change-of-flow.

During reset, the program counter is loaded with $3FFD and the program will start execution from this specific location.

8.2.3 Shadow Program Counter (SPC)

The shadow program counter is a 14-bit register. During a subroutine call using either a JSR or a BSR instruction, the return address will be saved into the SPC. Upon completion of the subroutine, the RTS instruction will restore the content of the program counter from the shadow program counter.

During reset, the shadow program counter is loaded with $3FFD.

8.2.4 Condition Code Register (CCR)

The 2-bit condition code register contains two status flags. The content of the CCR in the RS08 is not directly readable. The CCR bits can be tested using conditional branch instructions such as BCC and BEQ. These two register bits are directly accessible through the BDC interface. The following paragraphs provide detailed information about the CCR bits and how they are used. Figure 8-3 identifies the CCR bits and their bit positions.

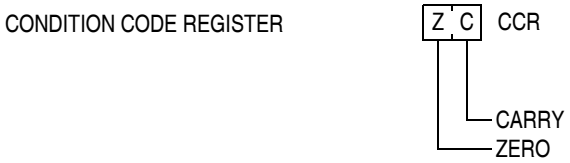


Figure 8-3. Condition Code Register (CCR)

The status bits (Z and C) are cleared to 0 after reset.

The two status bits indicate the results of arithmetic and other instructions. Conditional branch instructions will either branch to a new program location or allow the program to continue to the next instruction after the branch, depending on the values in the CCR status bit. Conditional branch instructions, such as BCC, BCS, and BNE, cause a branch depending on the state of a single CCR bit.

Often, the conditional branch immediately follows the instruction that caused the CCR bit(s) to be updated, as in this sequence:

```
        cmp     #5        ;compare accumulator A to 5
        blo     lower     ;branch if A smaller 5
more:   deca              ;do this if A not higher than or same as 5
lower:
```

8.2.7 Page Select Register (PAGESEL)

This 8-bit page select register allows the user to access all memory locations in the entire 16K-byte address space through a page window located from $00C0 to $00FF. This register resides at the memory mapped location $001F. For details on the PAGESEL register, please refer to the RS08 Core Reference Manual.

8.3 Addressing Modes

Whenever the MCU reads information from memory or writes information into memory, an addressing mode is used to determine the exact address where the information is read from or written to. This section explains several addressing modes and how each is useful in different programming situations.

Every opcode tells the CPU to perform a certain operation in a certain way. Many instructions, such as load accumulator (LDA), allow several different ways to specify the memory location to be operated on, and each addressing mode variation requires a separate opcode. All of these variations use the same instruction mnemonic, and the assembler knows which opcode to use based on the syntax and location of the operand field. In some cases, special characters are used to indicate a specific addressing mode (such as the # [pound] symbol, which indicates immediate addressing mode). In other cases, the value of the operand tells the assembler which addressing mode to use. For example, the assembler chooses short addressing mode instead of direct addressing mode if the operand address is from $0000 to $001F. Besides allowing the assembler to choose the addressing mode based on the operand address, assembler directives can also be used to force direct or tiny/short addressing mode by using the “>” or “<” prefix before the operand, respectively.

Some instructions use more than one addressing mode. For example, the move instructions use one addressing mode to access the source value from memory and a second addressing mode to access the destination memory location. For these move instructions, both addressing modes are listed in the documentation. All branch instructions use relative (REL) addressing mode to determine the destination for the branch, but BRCLR, BRSET, CBEQ, and DBNZ also must access a memory operand. These instructions are classified by the addressing mode used for the memory operand, and the relative addressing mode for the branch offset is assumed.

The discussion in the following paragraphs includes how each addressing mode works and the syntax clues that instruct the assembler to use a specific addressing mode.

8.3.1 Inherent Addressing Mode (INH)

This addressing mode is used when the CPU inherently knows everything it needs to complete the instruction and no addressing information is supplied in the source code. Usually, the operands that the CPU needs are located in the CPU’s internal registers, as in LSLA, CLRA, INCA, SLA, RTS, and others. A few inherent instructions, including no operation (NOP) and background (BGND), have no operands.

8.3.2 Relative Addressing Mode (REL)

Relative addressing mode is used to specify the offset address for branch instructions relative to the program counter. Typically, the programmer specifies the destination with a program label or an

8.5 Summary Instruction Table

Instruction Set Summary Nomenclature

The nomenclature listed here is used in the instruction descriptions in Table 8-1 through Table 8-2.

Operators

( )	=	Contents of register or memory location shown inside parentheses
←	=	Is loaded with (read: “gets”)
⇔	=	Exchange with
&	=	Boolean AND
|	=	Boolean OR
⊕	=	Boolean exclusive-OR
:	=	Concatenate
+	=	Add

CPU registers

A	=	Accumulator
CCR	=	Condition code register
PC	=	Program counter
PCH	=	Program counter, higher order (most significant) six bits
PCL	=	Program counter, lower order (least significant) eight bits
SPC	=	Shadow program counter
SPCH	=	Shadow program counter, higher order (most significant) six bits
SPCL	=	Shadow program counter, lower order (least significant) eight bits

Memory and addressing

M	=	A memory location or absolute data, depending on addressing mode
rel	=	The relative offset, which is the two’s complement number stored in the last byte of machine code corresponding to a branch instruction
X	=	Pseudo index register, memory location $000F
,X or D[X]	=	Memory location $000E pointing to the memory location defined by the pseudo index register (location $000F)

Condition code register (CCR) bits

Z	=	Zero indicator
C	=	Carry/borrow

CCR activity notation

–	=	Bit not affected
0	=	Bit forced to 0
1	=	Bit forced to 1
↕	=	Bit set or cleared according to results of operation
U	=	Undefined after the operation

Machine coding notation

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

Table 8-1. Instruction Set Summary (Sheet 2 of 6)

Source Form	Description	Operation	Effect on CCR		Address Mode	Opcode	Operand	Cycles
			Z	C				
BHS *rel* (1)	Branch if Higher or Same (Same as BCC)	PC ← (PC) + $0002 + *rel*, if (C) = 0	—	—	REL	34	rr	3
BLO *rel* (1)	Branch if Lower (Same as BCS)	PC ← (PC) + $0002 + *rel*, if (C) = 1	—	—	REL	35	rr	3
BNE *rel*	Branch if Not Equal	PC ← (PC) + $0002 + *rel*, if (Z) = 0	—	—	REL	36	rr	3
BRA *rel*	Branch Always	PC ← (PC) + $0002 + *rel*	—	—	REL	30	rr	3
BRN *rel* (1)	Branch Never	PC ← (PC) + $0002	—	—	REL	30	00	3
BRCLR *n*,*opr8a*,*rel*	Branch if Bit *n* in Memory Clear	PC ← (PC) + $0003 + *rel*, if (Mn) = 0	—	↕	DIR (b0)	01	dd rr	5
					DIR (b1)	03	dd rr	5
					DIR (b2)	05	dd rr	5
					DIR (b3)	07	dd rr	5
					DIR (b4)	09	dd rr	5
					DIR (b5)	0B	dd rr	5
					DIR (b6)	0D	dd rr	5
					DIR (b7)	0F	dd rr	5
BRCLR *n*,D[X],*rel*					IX (b0)	01	0E rr	5
					IX (b1)	03	0E rr	5
					IX (b2)	05	0E rr	5
					IX (b3)	07	0E rr	5
					IX (b4)	09	0E rr	5
					IX (b5)	0B	0E rr	5
					IX (b6)	0D	0E rr	5
					IX (b7)	0F	0E rr	5
BRCLR *n*,X,*rel*					DIR (b0)	01	0F rr	5
					DIR (b1)	03	0F rr	5
					DIR (b2)	05	0F rr	5
					DIR (b3)	07	0F rr	5
					DIR (b4)	09	0F rr	5
					DIR (b5)	0B	0F rr	5
					DIR (b6)	0D	0F rr	5
					DIR (b7)	0F	0F rr	5

1. This is a pseudo instruction supported by the normal RS08 instruction set.
2. This instruction is different from that of the HC08 and HCS08 in that the RS08 does not auto-increment the index register.

9.4.1.4 Stop

ICS stop mode is entered whenever the MCU enters stop. In this mode, all ICS clocks are stopped except ICSIRCLK which will remaining running if IREFSTEN is written to a 1.

When the MCU is interrupted from stop, the ICS will go back to the operating mode that was running when the MCU entered stop. If the internal reference was not running in stop (IREFSTEN = 0), the ICS will take some time, t_{ir_wu}, for the internal reference to wakeup. If the internal reference was already running in stop (IREFSTEN = 1), entering into FEI will take some time, t_{fll_wu}, for the FLL to return its previous acquired frequency.

9.4.2 Mode Switching

When changing from FBILP to either FEI or FBI, or anytime the trim value is written, the user should wait the FLL acquisition time, $t_{acquire}$, before FLL will be guaranteed to be at desired frequency.

9.4.3 Bus Frequency Divider

The BDIV bits can be changed at anytime and the actual switch to the new frequency will occur immediately.

9.4.4 Low Power Bit Usage

The low power bit (LP) is provided to allow the FLL to be disabled and thus conserve power when it is not being used. However, in some applications it may be desirable to enable the FLL and allow it to lock for maximum accuracy before switching to an FLL engaged mode. The FLL is disabled in bypass mode when LP = 1.

9.4.5 Internal Reference Clock

The ICSIRCLK frequency can be re-targeted by trimming the period of the internal reference clock. This can be done by writing a new value to the TRIM bits in the ICSTRM register. Writing a larger value will slow down the ICSIRCLK frequency, and writing a smaller value to the ICSTRM register will speed up the ICSIRCLK frequency. The TRIM bits will affect the ICSOUT frequency if the ICS is in FLL engaged internal (FEI), FLL bypassed internal (FBI), or FLL bypassed internal low power (FBILP) mode. The TRIM and FTRIM values will not be affected by a reset. For the ICS to run in stop, the LVDE and LVDSE bits in the SPMSC1 must both be set before entering stop.

Until ICSIRCLK is trimmed, ICSOUT frequencies may exceed the maximum chip-level frequency and violate the chip-level clock timing specifications (see the Device Overview chapter). The BDIV is reset to a divide by 2 to prevent the bus frequency from exceeding the maximum. The user should trim the device to an allowable frequency before changing BDIV to a divide by 1 operation.

Chapter 10
Analog Comparator (RS08ACMPV1)

10.1 Introduction

The analog comparator module (ACMP) provides a circuit for comparing two analog input voltages or for comparing one analog input voltage to an internal reference voltage. The comparator circuit is designed to operate across the full range of the supply voltage (rail to rail operation).

Figure 10-1 shows the MC9RS08KA2 Series block diagram with the ACMP highlighted.

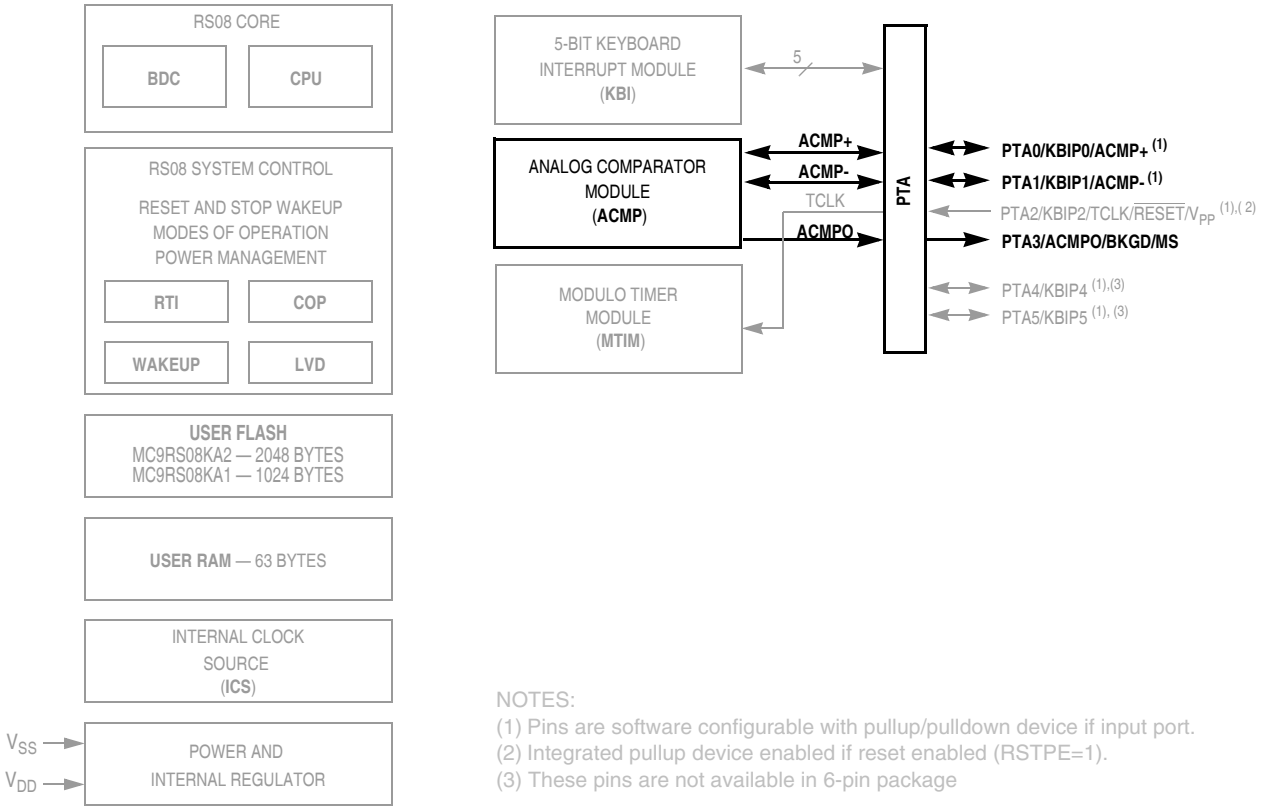


Figure 10-1. MC9RS08KA2 Series Block Diagram Highlighting ACMP Block and Pins

11.4.1 MTIM Operation Example

This section shows an example of the MTIM operation as the counter reaches a matching value from the modulo register.

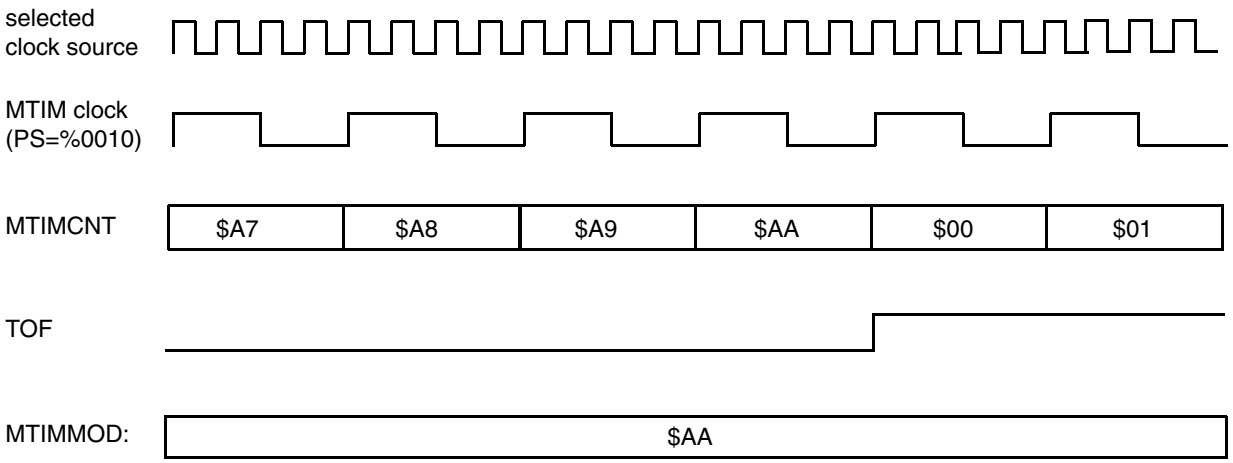


Figure 11-7. MTIM Counter Overflow Example

In the example of Figure 11-7, the selected clock source could be any of the four possible choices. The prescaler is set to PS = %0010 or divide-by-4. The modulo value in the MTIMMOD register is set to $AA. When the counter, MTIMCNT, reaches the modulo value of $AA, the counter overflows to $00 and continues counting. The timer overflow flag, TOF, sets when the counter value changes from $AA to $00. An MTIM overflow interrupt is generated when TOF is set, if TOIE = 1.

The BDC serial communication protocol requires the host to know the target BDC clock speed. Commands and data are sent most significant bit first (MSB-first) at 16 BDC clock cycles per bit. The interface times out if 512 BDC clock cycles occur between falling edges from the host. Any BDC command that was in progress when this timeout occurs is aborted without affecting the memory or operating mode of the target MCU system.

Figure 12-3 shows an external host transmitting a logic 1 or 0 to the BKGD pin of a target MCU. The host is asynchronous to the target so there is a 0-to-1 cycle delay from the host-generated falling edge to where the target perceives the beginning of the bit time. Ten target BDC clock cycles later, the target senses the bit level on the BKGD pin. Typically, the host actively drives the pseudo-open-drain BKGD pin during host-to-target transmissions to speed up rising edges. Because the target does not drive the BKGD pin during the host-to-target period, there is no need to treat the line as an open-drain signal during this period.

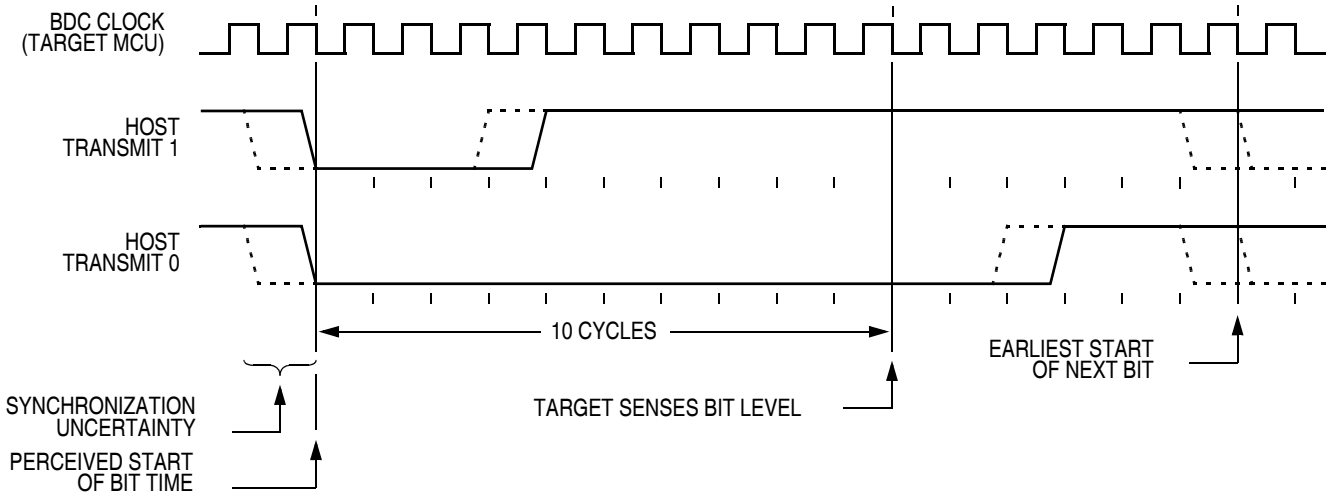


Figure 12-3. BDC Host-to-Target Serial Bit Timing

Figure 12-4 shows the host receiving a logic 1 from the target MCU. Because the host is asynchronous to the target, there is a 0-to-1 cycle delay from the host-generated falling edge on BKGD to the perceived start of the bit time in the target. The host holds the BKGD pin low long enough for the target to recognize it (at least two target BDC cycles). The host must release the low drive before the target drives a brief active-high speedup pulse seven cycles after the perceived start of the bit time. The host must sample the bit level approximately 10 cycles after it started the bit time.

Table 12-1. BDCSCR Register Field Descriptions (continued)

Field	Description
5 BKPTEN	**BDC Breakpoint Enable** — If this bit is clear, the BDC breakpoint is disabled and the FTS (force tag select) control bit and BDCBKPT match register are ignored 0 BDC breakpoint disabled. 1 BDC breakpoint enabled.
4 FTS	**Force/Tag Select** — When FTS = 1, a breakpoint is requested whenever the CPU address bus matches the BDCBKPT match register. When FTS = 0, a match between the CPU address bus and the BDCBKPT register causes the fetched opcode to be tagged. If this tagged opcode ever reaches the end of the instruction queue, the CPU enters active background mode rather than executing the tagged opcode. 0 Tag opcode at breakpoint address and enter active background mode if CPU attempts to execute that instruction. 1 Breakpoint match forces active background mode at next instruction boundary (address need not be an opcode).
2 WS	**Wait or Stop Status** — When the target CPU is in wait or stop mode, most BDC commands cannot function. However, the BACKGROUND command can be used to force the target CPU out of wait or stop and into active background mode where all BDC commands work. Whenever the host forces the target MCU into active background mode, the host must issue a READ_STATUS command to check that BDMACT = 1 before attempting other BDC commands. 0 Target CPU is running user application code or in active background mode (was not in wait or stop mode when background became active). 1 Target CPU is in wait or stop mode, or a BACKGROUND command was used to change from wait or stop to active background mode.
1 WSF	**Wait or Stop Failure Status** — This status bit is set if a memory access command failed due to the target CPU executing a wait or stop instruction at or about the same time. The usual recovery strategy is to issue a BACKGROUND command to get out of wait or stop mode into active background mode, repeat the command that failed, then return to the user program. (Typically, the host would restore CPU registers and stack values and re-execute the wait or stop instruction.) 0 Memory access did not conflict with a wait or stop instruction. 1 Memory access command failed because the CPU entered wait or stop mode.

12.4.2 BDC Breakpoint Match Register

This 16-bit register holds the 14-bit address for the hardware breakpoint in the BDC. The BKPTEN and FTS control bits in BDCSCR are used to enable and configure the breakpoint logic. Dedicated serial BDC commands (READ_BKPT and WRITE_BKPT) are used to read and write the BDCBKPT register. Breakpoints are normally set while the target MCU is in active background mode before running the user application program. However, because READ_BKPT and WRITE_BKPT are non-intrusive commands, they could be executed even while the user program is running. For additional information about setup and use of the hardware breakpoint logic in the BDC, refer to the RS08 Family Reference Manual.”

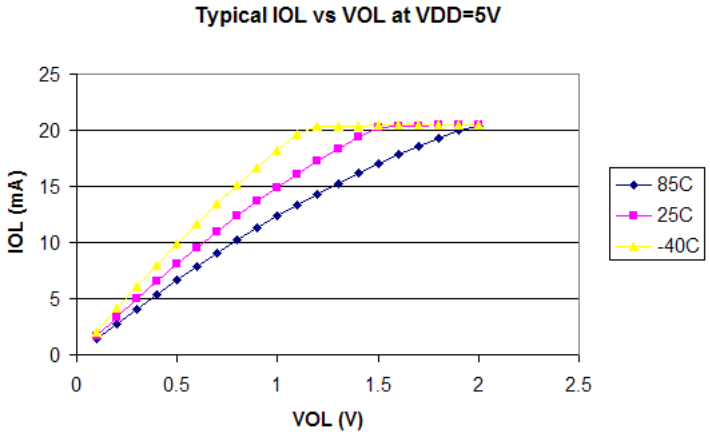


Figure 12-11. Typical I_{OL} vs. V_{OL} VDD = 5 V

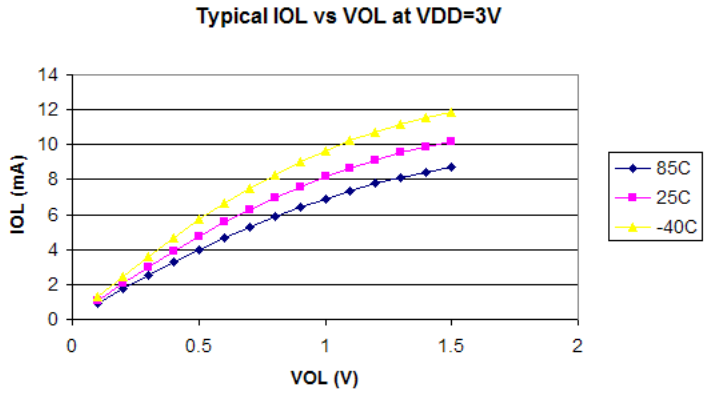


Figure 12-12. Typical I_{OL} vs. V_{OL} VDD = 3 V

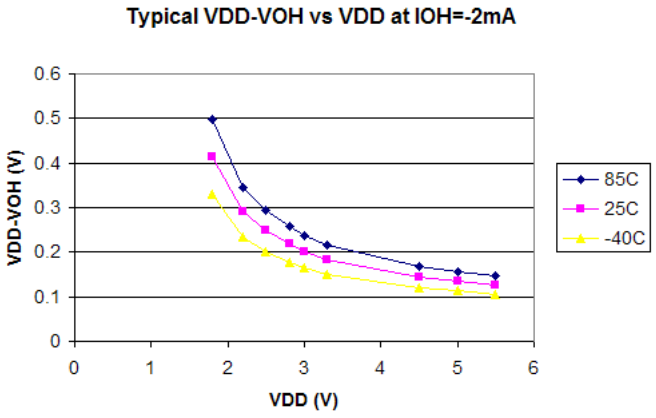


Figure 12-13. Typical V_{DD}-V_{OH} vs. V_{DD} at I_{OH}=2mA

Table A-7. Internal Clock Source Specifications

Characteristic	Symbol	Min	Typ[1]	Max	Unit
Stop recovery time (FLL wakeup to previous acquired frequency) IREFSTEN=0 IREFSTEN=1	t_wakeup t_{ir_wu} t_{fll_wu}	— —	 100 86	 —	µs

1 Data in typical column was characterized at 3.0 V and 5.0 V, 25°C or is typical recommended value.

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

3 This specification applies to any time the FLL reference source or reference divider is changed, trim value changed or changing from FLL disabled (FBILP) to FLL enabled (FEI, FBI).

A.9 AC Characteristics

This section describes ac timing characteristics for each peripheral system.

A.9.1 Control Timing

Table A-8. Control Timing

Parameter	Symbol	Min	Typical	Max	Unit
Bus frequency (t_{cyc} = 1/f_{Bus})	f_{Bus}	dc	—	10	MHz
Real time interrupt internal oscillator period	t_{RTI}	700	1000	1300	µs
External $\overline{\text{RESET}}$ pulse width[1]	t_{extrst}	150	—	—	ns
KBI pulse width[2]	t_{KBIPW}	1.5 t_{cyc}	—	—	ns
KBI pulse width in stop[1]	t_{KBIPWS}	100	—	—	ns
Port rise and fall time (load = 50 pF)[3] Slew rate control disabled (PTxSE = 0) Slew rate control enabled (PTxSE = 1)	t_{Rise}, t_{Fall}	 — —	 11 35	 — —	ns

1 This is the shortest pulse that is guaranteed to pass through the pin input filter circuitry. Shorter pulses may or may not be recognized.

2 This is the minimum pulse width that is guaranteed to pass through the pin synchronization circuitry. Shorter pulses may or may not be recognized. In stop mode, the synchronizer is bypassed so shorter pulses can be recognized in that case.

3 Timing is shown with respect to 20% V_{DD} and 80% V_{DD} levels. Temperature range –40°C to 85°C.

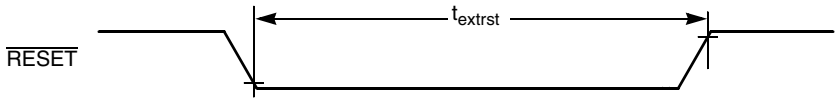
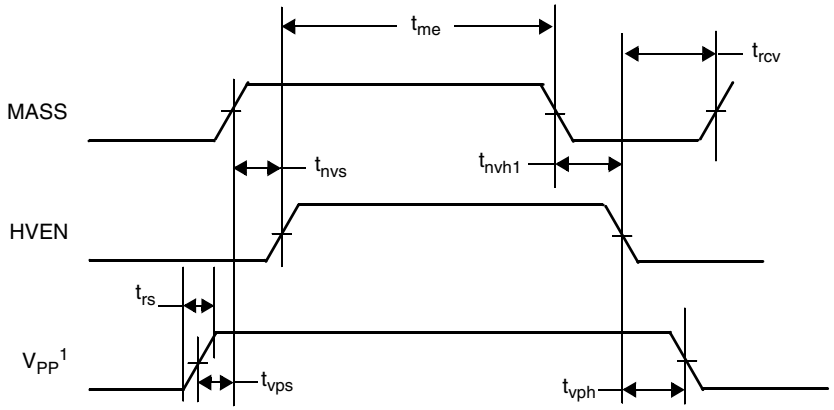
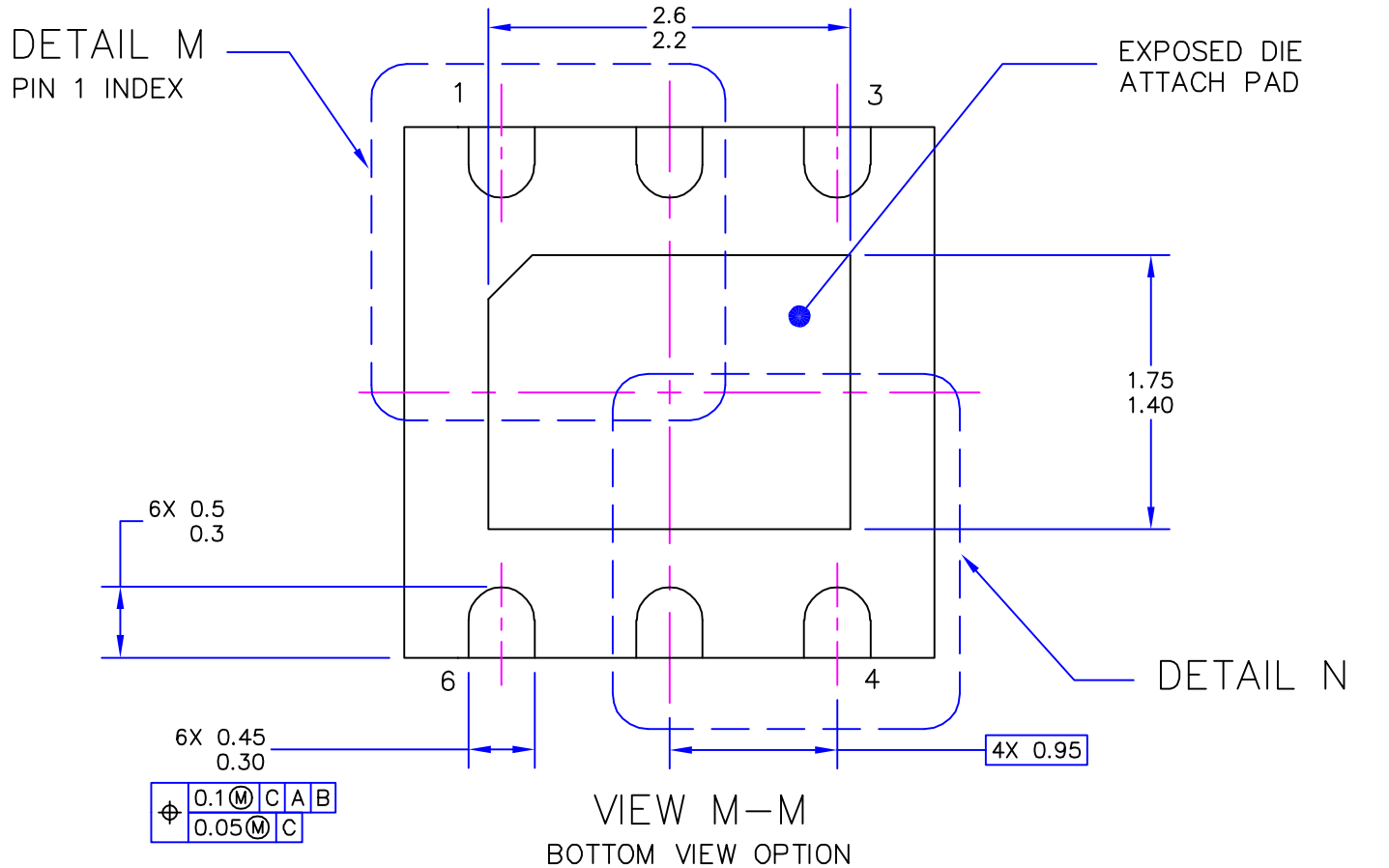


Figure A-1. Reset Timing

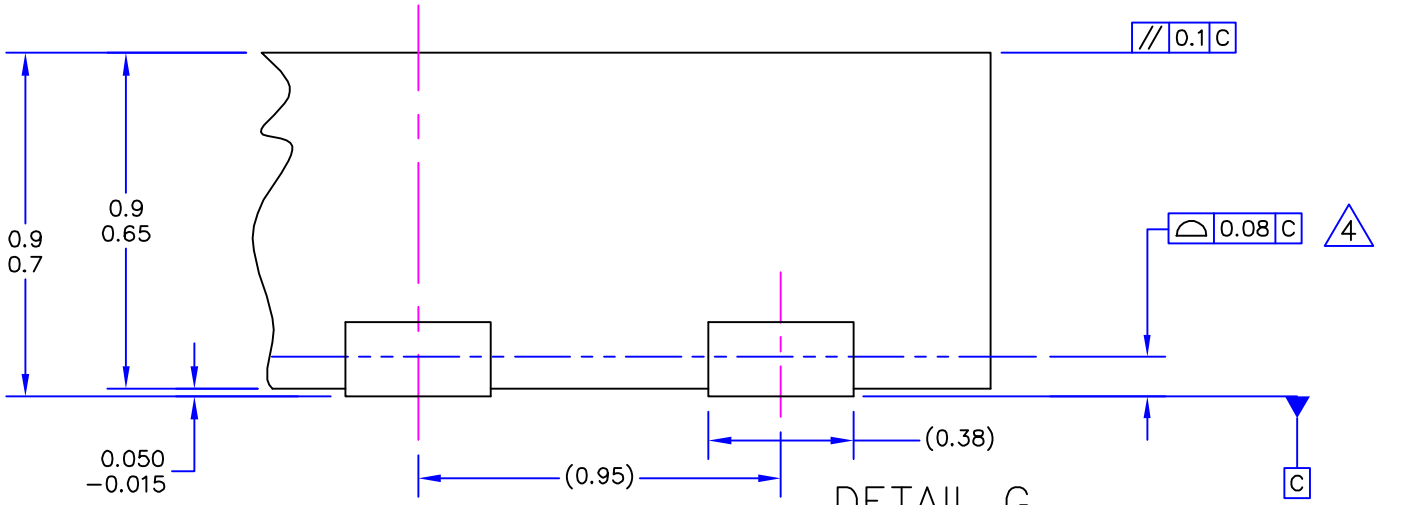


[1] V_{DD} must be at a valid operating voltage before voltage is applied or removed from the V_{PP} pin.

Figure A-5. Flash Mass Erase Timing



VIEW M–M
BOTTOM VIEW OPTION
(EXPOSED LEAD END TYPE)



DETAIL G
SIDE VIEW OPTION
(EXPOSED LEAD END TYPE)

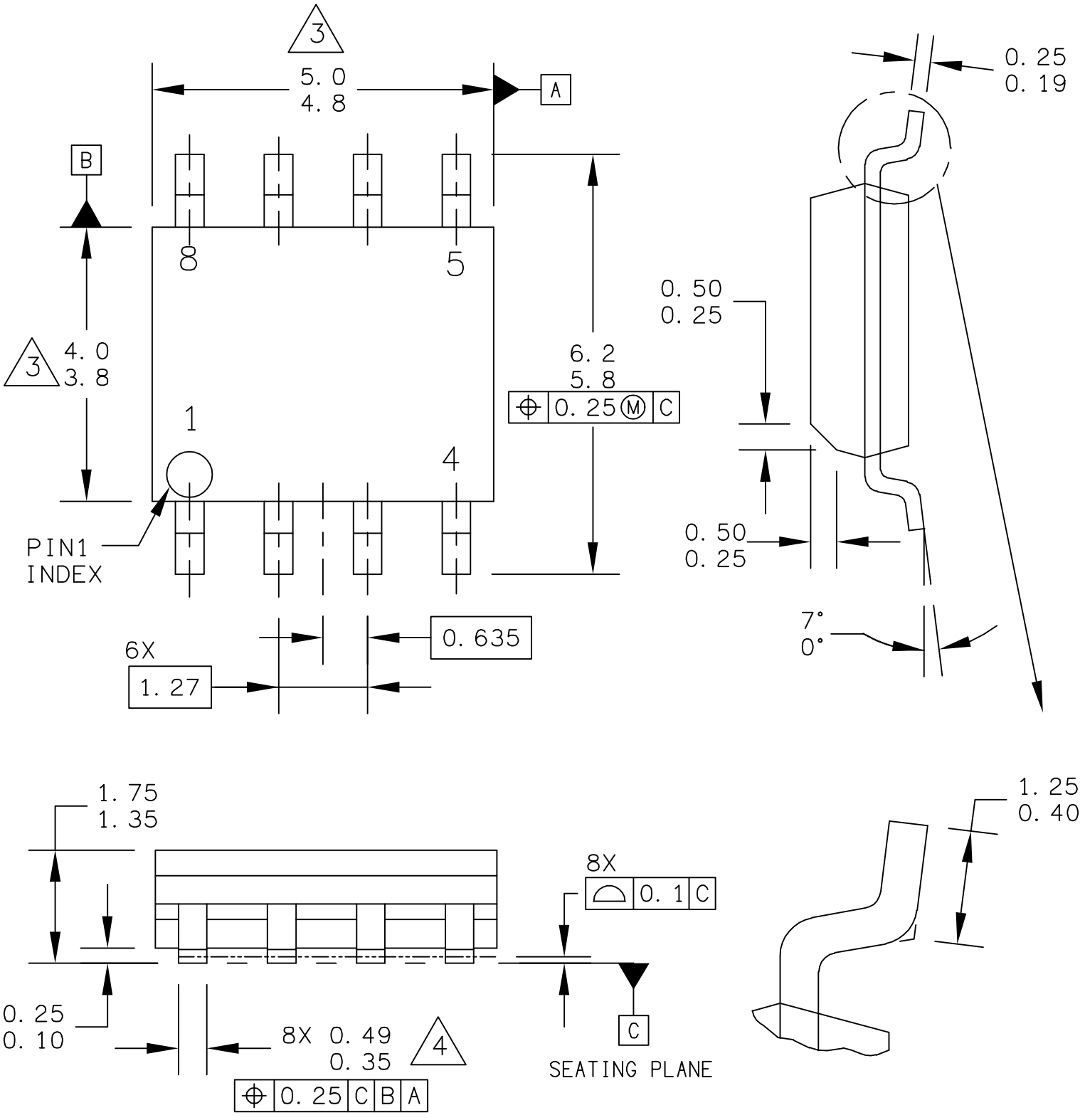
SEATING PLANE

TITLE: THERMALLY ENHANCED DUAL FLAT NON–LEADED PACKAGE (DFN) 6 TERMINAL, 0.95 PITCH (3 X 3 X 0.8)

CASE NUMBER: 1677–02

STANDARD: FREESCALE STD

PACKAGE CODE: 6197



MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE

TITLE:	DOCUMENT NO: 98ASB42564B	REV: V
8LD SOIC NARROW BODY	CASE NUMBER: 751-07	20 NOV 2007
	STANDARD: JEDEC MS-012AA	