Freescale Semiconductor - MC9RS08KA1CDBR Datasheet

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	2
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	6-VDFN Exposed Pad
Supplier Device Package	6-DFN-EP (3x3)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9rs08ka1cdbr

2.4.1 Power

V_{DD} and V_{SS} are the primary power supply pins for the MCU. This voltage source supplies power to all I/O buffer circuitry and to an internal voltage regulator. The internal voltage regulator provides a regulated lower-voltage source to the CPU and other internal circuitry of the MCU.

Typically, application systems have two separate capacitors across the power pins: a bulk electrolytic capacitor, such as a 10-μF tantalum capacitor, to provide bulk charge storage for the overall system, and a bypass capacitor, such as a 0.1-μF ceramic capacitor, located as near to the MCU power pins as practical to suppress high-frequency noise.

2.4.2 PTA2/KBIP2/TCLK/$\overline{\text{RESET}}$/V_{PP}

After a power-on reset (POR) into user mode, the PTA2/KBIP2/TCLK/$\overline{\text{RESET}}$/V_{PP} pin defaults to a general-purpose input port pin, PTA2. Setting RSTPE in SOPT configures the pin to be the $\overline{\text{RESET}}$ input pin. After configured as $\overline{\text{RESET}}$, the pin will remain as $\overline{\text{RESET}}$ until the next POR. The $\overline{\text{RESET}}$ pin can be used to reset the MCU from an external source when the pin is driven low. When enabled as the $\overline{\text{RESET}}$ pin (RSTPE = 1), the internal pullup device is automatically enabled.

External V_{PP} voltage (typically 12 V, see Section A.10, “FLASH Specifications”) is required on this pin when performing Flash programming or erasing. The V_{PP} connection is always connected to the internal Flash module regardless of the pin function. To avoid over stressing the Flash, external V_{PP} voltage must be removed and voltage higher than V_{DD} must be avoided when Flash programming or erasing is not taking place.

NOTE

> This pin does not contain a clamp diode to V_{DD} and should not be driven above V_{DD} when Flash programming or erasing is not taking place.

2.4.3 PTA3/ACMPO/BKGD/MS

The background / mode select function is shared with an output-only pin on PTA3 pin and the optional analog comparator output. While in reset, the pin functions as a mode select pin. Immediately after reset rises, the pin functions as the background pin and can be used for background debug communication. While functioning as a background / mode select pin, this pin has an internal pullup device enabled. To use as an output-only port, BKGDPE in SOPT must be cleared.

If nothing is connected to this pin, the MCU will enter normal operating mode at the rising edge of reset. If a debug system is connected to the 6-pin standard background debug header, it can hold BKGD/MS low during the power-on-reset, which forces the MCU to active background mode.

The BKGD pin is used primarily for background debug controller (BDC) communications using a custom protocol that uses 16 clock cycles of the target MCU’s BDC clock per bit time. The target MCU’s BDC clock equals the bus clock rate; therefore, no significant capacitance should connected to the BKGD/MS pin that could interfere with background serial communications.

Although the BKGD pin is a pseudo open-drain pin, the background debug communication protocol provides brief, actively driven, high speedup pulses to ensure fast rise times. Small capacitances from

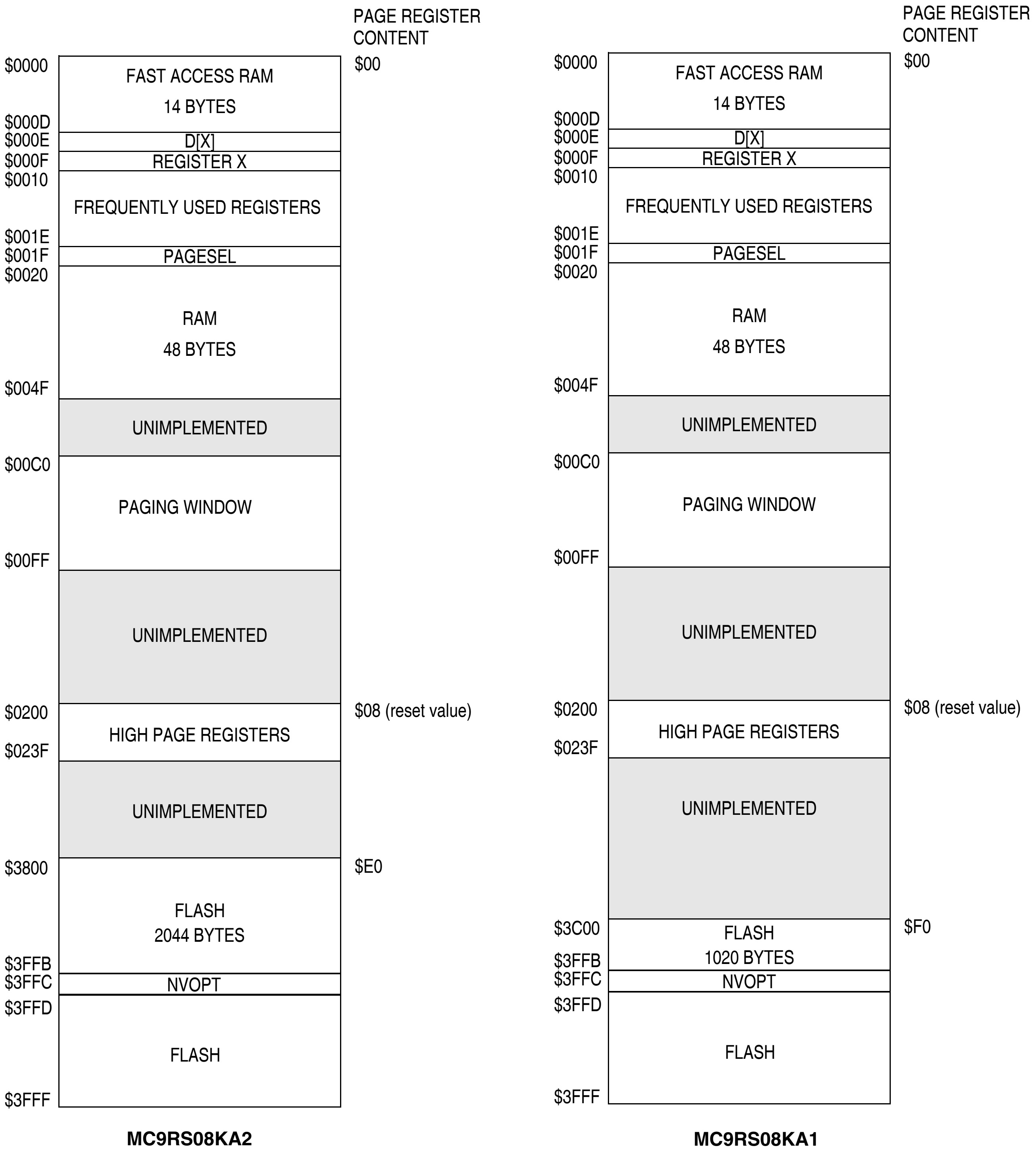


Figure 4-1. MC9RS08KA2 Series Memory Maps

Frequently used registers can make use of the short addressing mode instructions for faster load, store, and clear operations. For short addressing mode instructions, the operand is encoded along with the opcode to a single byte.

Table 4-1. Register Summary

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
$0000–$000D		Fast Access RAM							
$000E	D[X][1]	Bit 7	6	5	4	3	2	1	Bit 0
$000F	X	Bit 7	6	5	4	3	2	1	Bit 0
$0010	PTAD	0	0	PTAD5	PTAD4	PTAD3	PTAD2	PTAD1	PTAD0
$0011	PTADD	0	0	PTADD5	PTADD4	0	0	PTADD1	PTADD0
$0012	Unimplemented	—	—	—	—	—	—	—	—
$0013	ACMPSC	ACME	ACBGS	ACF	ACIE	ACO	ACOPE	ACMOD	
$0014	ICSC1	0	CLKS	0	0	0	0	0	IREFSTEN
$0015	ICSC2	BDIV		0	0	LP	0	0	0
$0016	ICSTRM	TRIM							
$0017	ICSSC	0	0	0	0	0	CLKST	0	FTRIM
$0018	MTIMSC	TOF	TOIE	TRST	TSTP	0	0	0	0
$0019	MTIMCLK	0	0	CLKS		PS			
$001A	MTIMCNT	COUNT							
$001B	MTIMMOD	MOD							
$001C	KBISC	0	0	0	0	KBF	KBACK	KBIE	KBIMOD
$001D	KBIPE	—	—	KBIPE5	KBIPE4	—	KBIPE2	KBIPE1	KBIPE0
$001E	KBIES	—	—	KBEDG5	KBEDG4	—	KBEDG2	KBEDG1	KBEDG0
$001F	PAGESEL	AD13	AD12	AD11	AD10	AD9	AD8	AD7	AD6
$0020–$004F		RAM							
$0050–$00BF	Unimplemented	—	—	—	—	—	—	—	—
$00C0–$00FF		Paging Window							
$0100–$01FF	Unimplemented	—	—	—	—	—	—	—	—
$0200	SRS	POR	PIN	COP	ILOP	ILAD	0	LVD	0
$0201	SOPT	COPE	COPT	STOPE	0	0	0	BKGDPE	RSTPE
$0202	SIP1	—	—	—	KBI	ACMP	MTIM	RTI	LVD
$0203	Unimplemented	—	—	—	—	—	—	—	—
$0204	Reserved	—	—	—	—	—	—	—	—
$0205	Unimplemented	—	—	—	—	—	—	—	—
$0206	SDIDH	REV3	REV2	REV1	REV0	ID			
$0207	SDIDL	ID							
$0208	SRTISC	RTIF	RTIACK	RTICLKS	RTIE	0	RTIS		
$0209	SPMSC1	LVDF	LVDACK	LVDIE	LVDRE	LVDSE	LVDE	0	BGBE
$020A	Reserved	—	—	—	—	—	—	—	—
$020B	Reserved	—	—	—	—	—	—	—	—

▨ = Unimplemented or Reserved

- 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.

Other instructions may be executed between the test and the conditional branch as long as the only instructions used are those which do not disturb the CCR bits that affect the conditional branch. For instance, a test is performed in a subroutine or function and the conditional branch is not executed until the subroutine has returned to the main program. This is a form of parameter passing (that is, information is returned to the calling program in the condition code bits).

Z — Zero Flag

The Z bit is set to indicate the result of an operation was $00.

Branch if equal (BEQ) and branch if not equal (BNE) are simple branches that branch based solely on the value in the Z bit. All load, store, move, arithmetic, logical, shift, and rotate instructions cause the Z bit to be updated.

C — Carry

After an addition operation, the C bit is set if the source operands were both greater than or equal to $80 or if one of the operands was greater than or equal to $80 and the result was less than $80. This is equivalent to an unsigned overflow. A subtract or compare performs a subtraction of a memory operand from the contents of a CPU register so after a subtract operation, the C bit is set if the unsigned value of the memory operand was greater than the unsigned value of the CPU register. This is equivalent to an unsigned borrow or underflow.

Branch if carry clear (BCC) and branch if carry set (BCS) are branches that branch based solely on the value in the C bit. The C bit is also used by the unsigned branches BLO and BHS. Add, subtract, shift, and rotate instructions cause the C bit to be updated. The branch if bit set (BRSET) and branch if bit clear (BRCLR) instructions copy the tested bit into the C bit to facilitate efficient serial-to-parallel conversion algorithms. Set carry (SEC) and clear carry (CLC) allow the carry bit to be set or cleared directly. This is useful in combination with the shift and rotate instructions and for routines that pass status information back to a main program, from a subroutine, in the C bit.

The C bit is included in shift and rotate operations so those operations can easily be extended to multi-byte operands. The shift and rotate operations can be considered 9-bit shifts that include an 8-bit operand or CPU register and the carry bit of the CCR. After a logical shift, C holds the bit that was shifted out of the 8-bit operand. If a rotate instruction is used next, this C bit is shifted into the operand for the rotate, and the bit that gets shifted out the other end of the operand replaces the value in C so it can be used in subsequent rotate instructions.

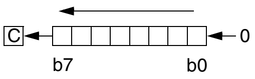
8.2.5 Indexed Data Register (D[X])

This 8-bit indexed data register allows the user to access the data in the direct page address space indexed by X. This register resides at the memory mapped location $000E. For details on the D[X] register, please refer to Section 8.3.8, “Indexed Addressing Mode (IX, Implemented by Pseudo Instructions).”

8.2.6 Index Register (X)

This 8-bit index register allows the user to index or address any location in the direct page address space. This register resides at the memory mapped location $000F. For details on the X register, please refer to Section 8.3.8, “Indexed Addressing Mode (IX, Implemented by Pseudo Instructions).”

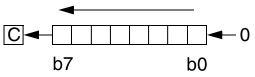
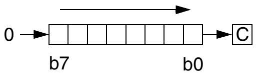
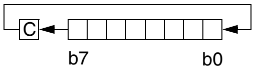
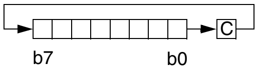
Table 8-1. Instruction Set Summary (Sheet 1 of 6)

Source Form	Description	Operation	Effect on CCR		Address Mode	Opcode	Operand	Cycles
			Z	C				
ADC #*opr8i* ADC *opr8a* ADC ,X [1] ADC X	Add with Carry	A ← (A) + (M) + (C) A ← (A) + (X) + (C)	↕	↕	IMM DIR IX DIR	A9 B9 B9 B9	ii dd 0E 0F	2 3 3 3
ADD #*opr8i* ADD *opr8a* ADD *opr4a* ADD ,X [1] ADD X	Add without Carry	A ← (A) + (M)	↕	↕	IMM DIR TNY IX DIR	AB BB 6*x* 6E 6F	ii dd	2 3 3 3 3
AND #*opr8i* AND *opr8a* AND ,X [1] AND X	Logical AND	A ← (A) & (M) A ← (A) & (X)	↕	—	IMM DIR IX DIR	A4 B4 B4 B4	ii dd 0E 0F	2 3 3 3
ASLA[1]	Arithmetic Shift Left		↕	↕	INH	48		1
BCC *rel*	Branch if Carry Bit Clear	PC ← (PC) + $0002 + *rel*, if (C) = 0	—	—	REL	34	rr	3
BCLR *n*,*opr8a* BCLR *n*,D[X] BCLR *n*,X	Clear Bit n in Memory	Mn ← 0	—	—	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7) IX (b0) IX (b1) IX (b2) IX (b3) IX (b4) IX (b5) IX (b6) IX (b7) DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	11 13 15 17 19 1B 1D 1F 11 13 15 17 19 1B 1D 1F 11 13 15 17 19 1B 1D 1F	dd dd dd dd dd dd dd dd 0E 0E 0E 0E 0E 0E 0E 0E 0F 0F 0F 0F 0F 0F 0F 0F	5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
BCS *rel*	Branch if Carry Bit Set (Same as BLO)	PC ← (PC) + $0002 + *rel*, if (C) = 1	—	—	REL	35	rr	3
BEQ *rel*	Branch if Equal	PC ← (PC) + $0002 + *rel*, if (Z) = 1	—	—	REL	37	rr	3
BGND	Background	Enter Background Debug Mode	—	—	INH	BF		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.

Table 8-1. Instruction Set Summary (Sheet 5 of 6)

Source Form	Description	Operation	Effect on CCR		Address Mode	Opcode	Operand	Cycles
			Z	C				
LDX #*opr8i* [1] LDX *opr8a* [1] LDX ,X [1]	Load Index Register from Memory	$0F ← (M)	↕	—	IMD DIR IX	3E 4E 4E	ii 0F dd 0F 0E 0E	4 5 5
LSLA	Logical Shift Left		↕	↕	INH	48		1
LSRA	Logical Shift Right		↕	↕	INH	44		1
MOV *opr8a,opr8a* MOV #*opr8i,opr8a* MOV D[X],*opr8a* MOV *opr8a*,D[X] MOV #*opr8i*,D[X]	Move	$(M)_{destination} \leftarrow (M)_{source}$	↕	—	DD IMD IX/DIR DIR/IX IMM/IX	4E 3E 4E 4E 3E	dd dd ii dd 0E dd dd 0E ii 0E	5 4 5 5 4
NOP	No Operation	None	—	—	INH	AC		1
ORA #*opr8i* ORA *opr8a* ORA ,X [1] ORA X	Inclusive OR Accumulator and Memory	A ← (A) | (M) A ← (A) | (X)	↕	—	IMM DIR IX DIR	AA BA BA BA	ii dd 0E 0F	2 3 3 3
ROLA	Rotate Left through Carry		↕	↕	INH	49		1
RORA	Rotate Right through Carry		↕	↕	INH	46		1
RTS	Return from Subroutine	Pull PC from shadow PC	—	—	INH	BE		3
SBC #*opr8i* SBC *opr8a* SBC ,X [1] SBC X	Subtract with Carry	A ← (A) – (M) – (C) A ← (A) – (X) – (C)	↕	↕	IMM DIR IX DIR	A2 B2 B2 B2	ii dd 0E 0F	2 3 3 3
SEC	Set Carry Bit	C ← 1	—	1	INH	39		1
SHA	Swap Shadow PC High with A	A ⇔ SPCH	—	—	INH	45		1
SLA	Swap Shadow PC Low with A	A ⇔ SPCL	—	—	INH	42		1
STA *opr8a* STA *opr5a* STA ,X [1] STA X	Store Accumulator in Memory	M ← (A)	↕	—	DIR SRT IX SRT	B7 Ex / Fx EE EF	dd	3 2 2 2
STX *opr8a* [1]	Store Index Register in Memory	M ← (X)	↕	—	DIR	4E	0F dd	5
STOP	Put MCU into stop mode		—	—	INH	AE		2+

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.

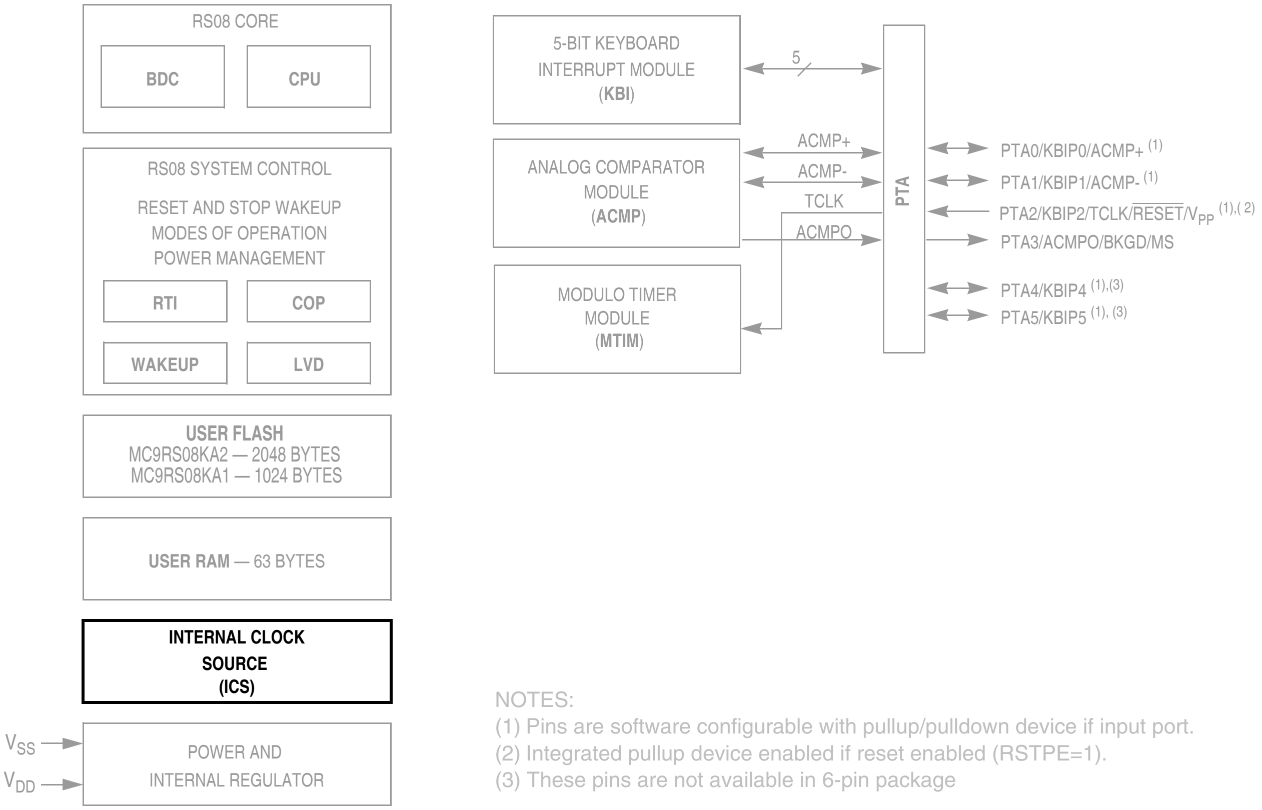
Chapter 9
Internal Clock Source (RS08ICSV1)

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.



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 9-1. MC9RS08KA2 Series Block Diagram Highlighting ICS Block

	7	6	5	4	3	2	1	0
R	0	CLKS	0	0	0	0	0	IREFSTEN
W	▨		▨	▨	▨	▨	▨	
Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented

Figure 9-3. ICS Control Register 1 (ICSC1)

Table 9-2. ICSC1 Field Descriptions

Field	Description
6 CLKS	**Clock Source Select** — Selects the clock source that controls the bus frequency. The actual bus frequency depends on the value of the BDIV bits. 0 Output of FLL is selected 1 Internal reference clock is selected
0 IREFSTEN	**Internal Reference Stop Enable** — Controls whether the internal reference clock remains enabled when the ICS enters stop mode. 1 Internal reference clock remains enabled in stop 0 Internal reference clock is disabled in stop

9.3.2 ICS Control Register 2 (ICSC2)

	7	6	5	4	3	2	1	0
R	BDIV		0	0	LP	0	0	0
W			▨	▨		▨	▨	▨
Reset:	0	1	0	0	0	0	0	0

▨ = Unimplemented

Figure 9-4. ICS Control Register 2 (ICSC2)

Table 9-3. ICSC2 Field Descriptions

Field	Description
7:6 BDIV	**Bus Frequency Divider** — Selects the amount to divide down the clock source selected by the CLKS bit. This controls the bus frequency. 00 Encoding 0 — Divides selected clock by 1 01 Encoding 1 — Divides selected clock by 2 (reset default) 10 Encoding 2 — Divides selected clock by 4 11 Encoding 3 — Divides selected clock by 8
3 LP	**Low Power Select** — Controls whether the FLL is disabled in FLL bypassed modes. 1 FLL is disabled in bypass modes 0 FLL is not disabled in bypass mode

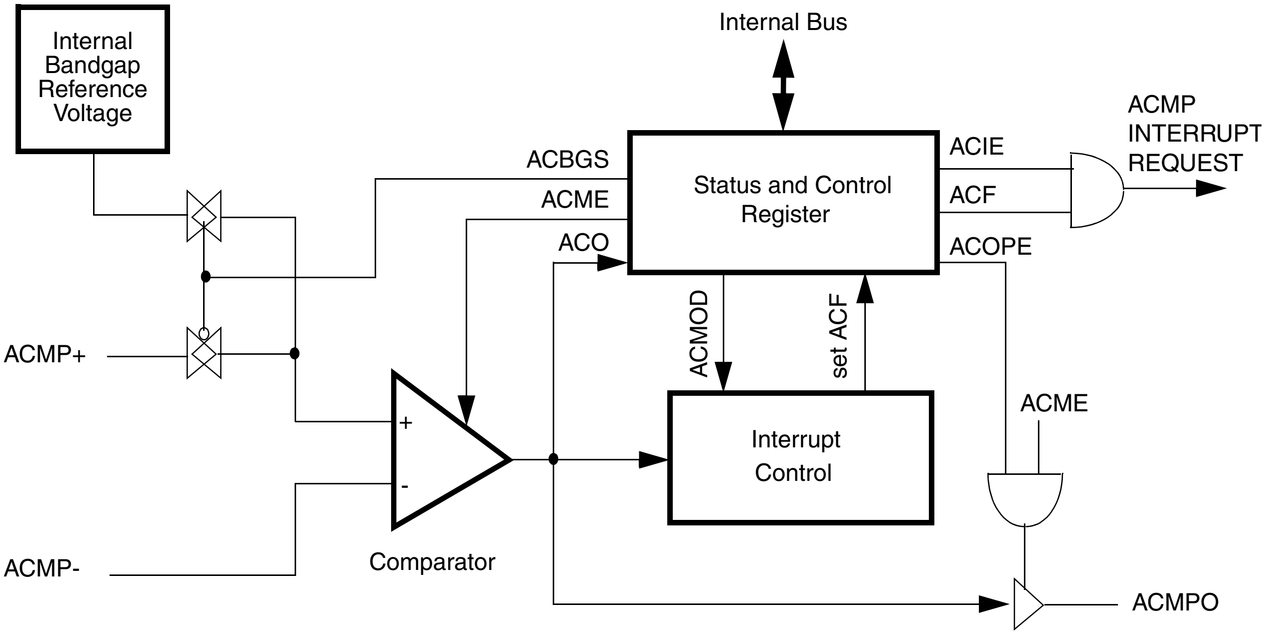


Figure 10-2. Analog Comparator (ACMP) Block Diagram

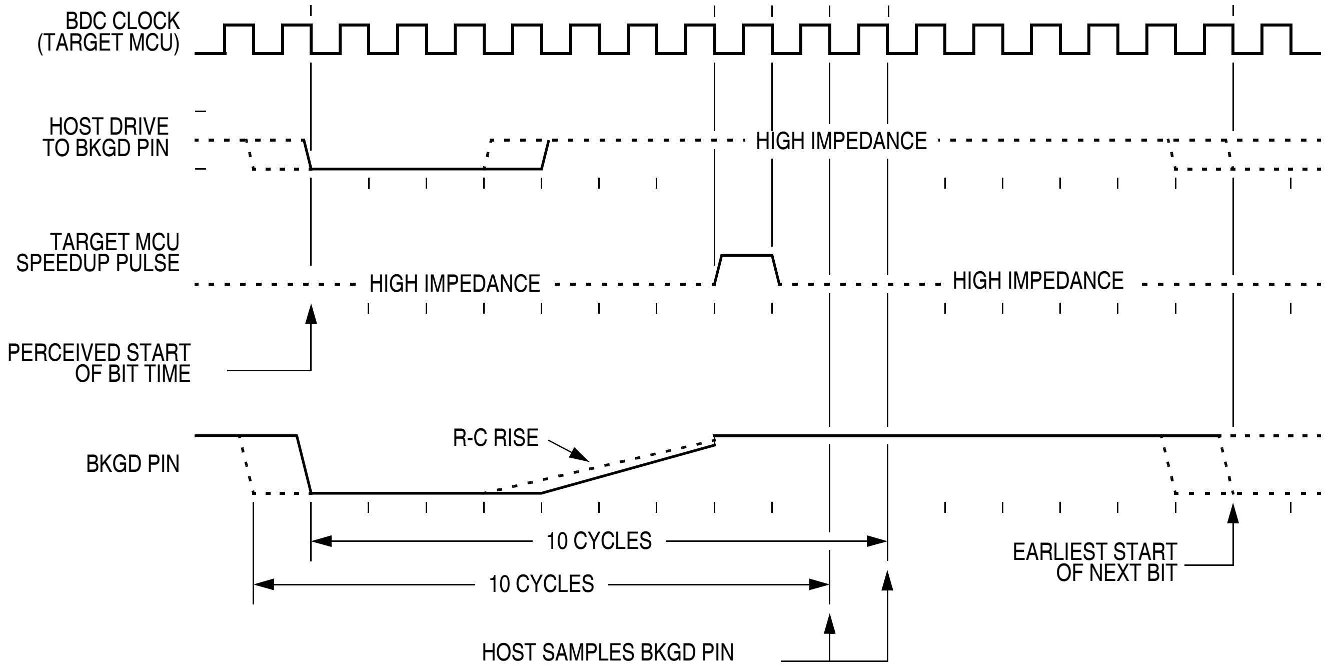


Figure 12-4. BDC Target-to-Host Serial Bit Timing (Logic 1)

Figure 12-5 shows the host receiving a logic 0 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 start of the bit time as perceived by the target. The host initiates the bit time but the target finishes it. Because the target wants the host to receive a logic 0, it drives the BKGD pin low for 13 BDC clock cycles, then briefly drives it high to speed up the rising edge. The host samples the bit level approximately 10 cycles after starting the bit time.

- Subsequent bits must occur within 512 BDC cycles of the last bit sent.

12.4 BDC Registers and Control Bits

The BDC contains two non-CPU accessible registers:

- The BDC status and control register (BDCSCR) is an 8-bit register containing control and status bits for the background debug controller.
- The BDC breakpoint register (BDCBKPT) holds a 16-bit breakpoint match address.

These registers are accessed with dedicated serial BDC commands and are not located in the memory space of the target MCU (so they do not have addresses and cannot be accessed by user programs).

Some of the bits in the BDCSCR have write limitations; otherwise, these registers may be read or written at any time. For example, the ENBDM control bit may not be written while the MCU is in active background mode. This prevents the ambiguous condition of the control bit forbidding active background mode while the MCU is already in active background mode. Also, the status bits (BDMACT, WS, and WSF) are read-only status indicators and can never be written by the WRITE_CONTROL serial BDC command.

12.4.1 BDC Status and Control Register (BDCSCR)

This register can be read or written by serial BDC commands (READ_STATUS and WRITE_CONTROL) but is not accessible to user programs because it is not located in the normal memory map of the MCU.

	7	6	5	4	3	2	1	0
R	ENBDM	BDMACT	BKPTEN	FTS	0	WS	WSF	0
W		▨			▨	▨	▨	▨
Normal Reset	0	0	0	0	0	0	0	0
Reset in Active BDM:	1	1	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 12-6. BDC Status and Control Register (BDCSCR)

Table 12-1. BDCSCR Register Field Descriptions

Field	Description
7 ENBDM	**Enable BDM (Permit Active Background Mode)** — Typically, this bit is written to 1 by the debug host shortly after the beginning of a debug session or whenever the debug host resets the target and remains 1 until a normal reset clears it. If the application can go into stop mode, this bit is required to be set if debugging capabilities are required. 0 BDM cannot be made active (non-intrusive commands still allowed). 1 BDM can be made active to allow active background mode commands.
6 BDMACT	**Background Mode Active Status** — This is a read-only status bit. 0 BDM not active (user application program running). 1 BDM active and waiting for serial commands.

Table 12-2. RS08 BDC Command Summary (continued)

Command Mnemonic	Active Background Mode/ Non-Intrusive	Coding Structure	Description
WRITE_A	Active background mode	48/WD/d	Write accumulator (A)
READ_CCR_PC	Active background mode	6B/d/RD16[5]	Read the CCR bits z, c concatenated with the 14-bit program counter (PC) RD16=zc:PC
WRITE_CCR_PC	Active background mode	4B/WD16/d[6]	Write the CCR bits z, c concatenated with the 14-bit program counter (PC) WD16=zc:PC
READ_SPC	Active background mode	6F/d/RD16[7]	Read the 14-bit shadow program counter (SPC) RD16=0:0:SPC
WRITE_SPC	Active background mode	4F/WD16/d[8]	Write 14-bit shadow program counter (SPC) WD16 = x:x:SPC, the two most significant bits shown by “x” are ignored by target

[1] The SYNC command is a special operation which does not have a command code.
[2] 18 was HCS08 BDC command for TAGGO.
[3] Each RD requires a delay between host read data byte and next read, command ends when target detects a soft-reset.
[4] Each WD requires a delay between host write data byte and next byte, command ends when target detects a soft-reset.
[5] HCS08 BDC had separate READ_CCR and READ_PC commands, the RS08 BDC combined this commands.
[6] HCS08 BDC had separate WRITE_CCR and WRITE_PC commands, the RS08 BDC combined this commands.
[7] 6F is READ_SP (read stack pointer) for HCS08 BDC.
[8] 4F is WRITE_SP (write stack pointer) for HCS08 BDC.

Table A-5. Supply Current Characteristics (continued)

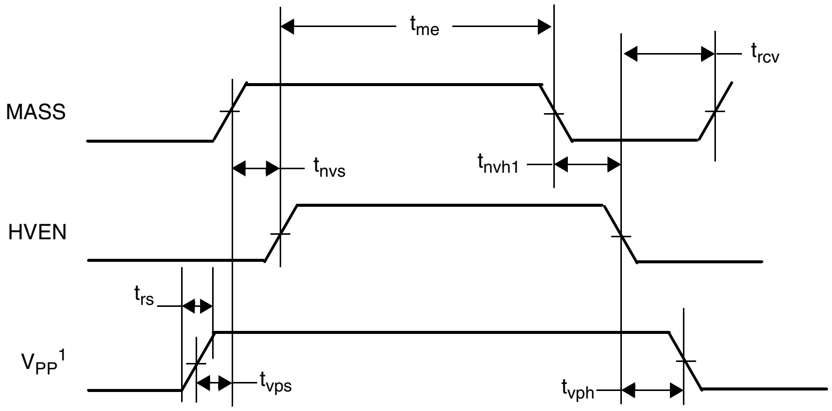
Parameter	Symbol	V_{DD} (V)	Typical[1]	Max[2]	Temp. (°C)
ACMP adder from stop (ACME = 1)	—	5	15 µA	20 µA	25
					85
		3	15 µA	20 µA	25
					85
		1.8	15 µA	20 µA	25
					85
RTI adder from stop with 1-kHz clock source enabled[4]	—	5	300 nA	500 nA	25 85
		3	300 nA	500 nA	25 85
		1.8	300 nA	500 nA	25 85
RTI adder from stop with 32-kHz ICS internal clock source reference enabled	—	5	140 µA	165 µA	25 85
		3	140 µA	165 µA	25 85
		1.8	135 µA	160 µA	25 85
LVI adder from stop (LVDE=1 and LVDSE=1)	—	5	70 µA	85 µA	25 85
		3	70 µA	85 µA	25 85
		1.8	65 µA	80 µA	25 85

[1] Typicals are measured at 25°C.

[2] Maximum value is measured at the nominal V_{DD} voltage times 10% tolerance. Values given here are preliminary estimates prior to completing characterization

[3] Does not include any dc loads on port pins

[4] Most customers are expected to find that auto-wakeup from stop can be used instead of the higher current wait mode. Wait mode typical is 560 µA at 3 V and 422 µA at 2V with f_{Bus} = 1 MHz.



[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

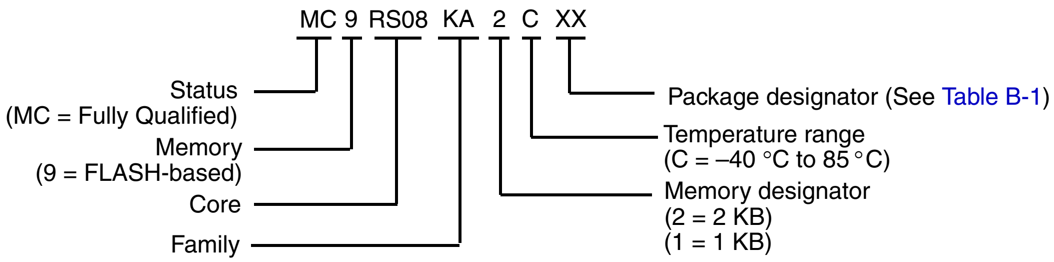
Appendix B
Ordering Information and Mechanical Drawings

B.1 Ordering Information

This section contains ordering numbers for MC9RS08KA2 Series devices. See below for an example of the device numbering system.

Table B-1. Device Numbering System

Device Number	Memory		Package		
	FLASH	RAM	Type	Designator	Document No.
MC9RS08KA2 MC9RS08KA1	2 KB 1 KB	63 bytes	6 DFN	DB	98ARL10602D
			8 PDIP	PC	98ASB42420B
			8 NB-SOIC	SC	98ASB42564B

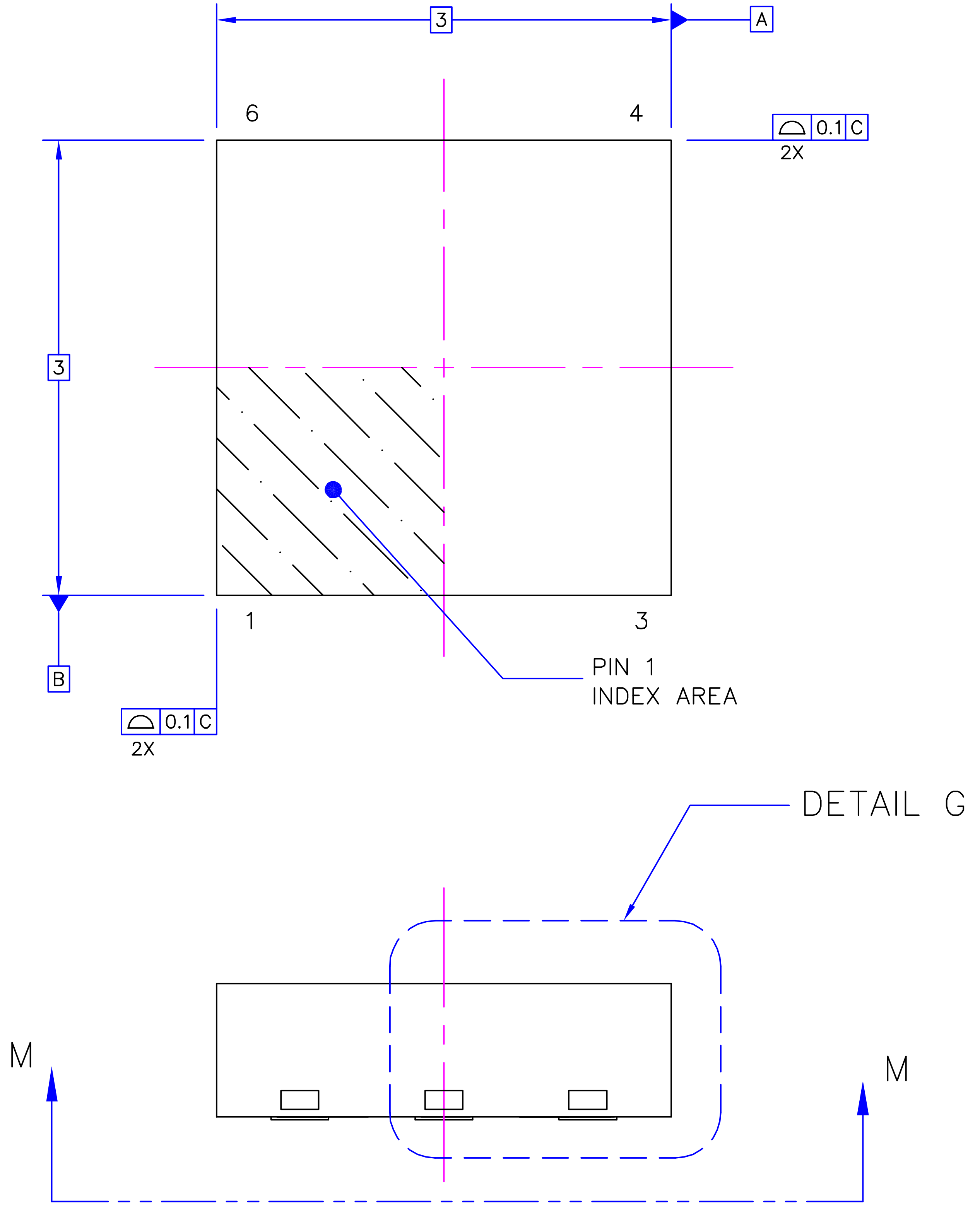


B.2 Mechanical Drawings

This following pages contain mechanical specifications for MC9RS08KA2 Series package options:

- 6-pin DFN (dual flat no-lead)
- 8-pin PDIP (plastic dual in-line pin)
- 8-pin NB-SOIC (narrow body small outline integrated circuit)

freescale semiconductor	MECHANICAL OUTLINES DICTIONARY	DOCUMENT NO: 98ARL10602D
		PAGE: 1677
	DO NOT SCALE THIS DRAWING	REV: C



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	SHEET: 1 OF 6

freescale semiconductor

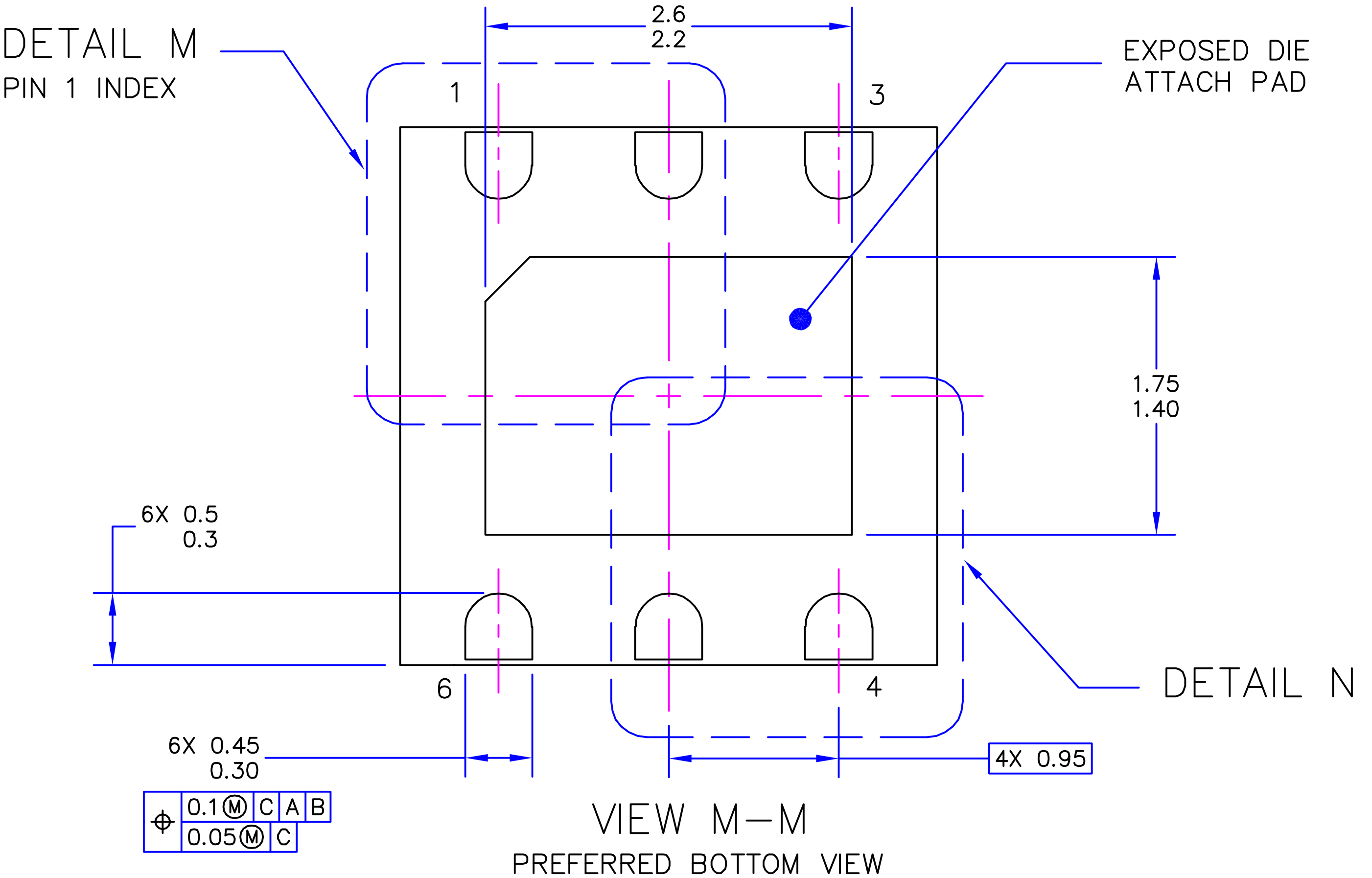
MECHANICAL OUTLINES DICTIONARY

DO NOT SCALE THIS DRAWING

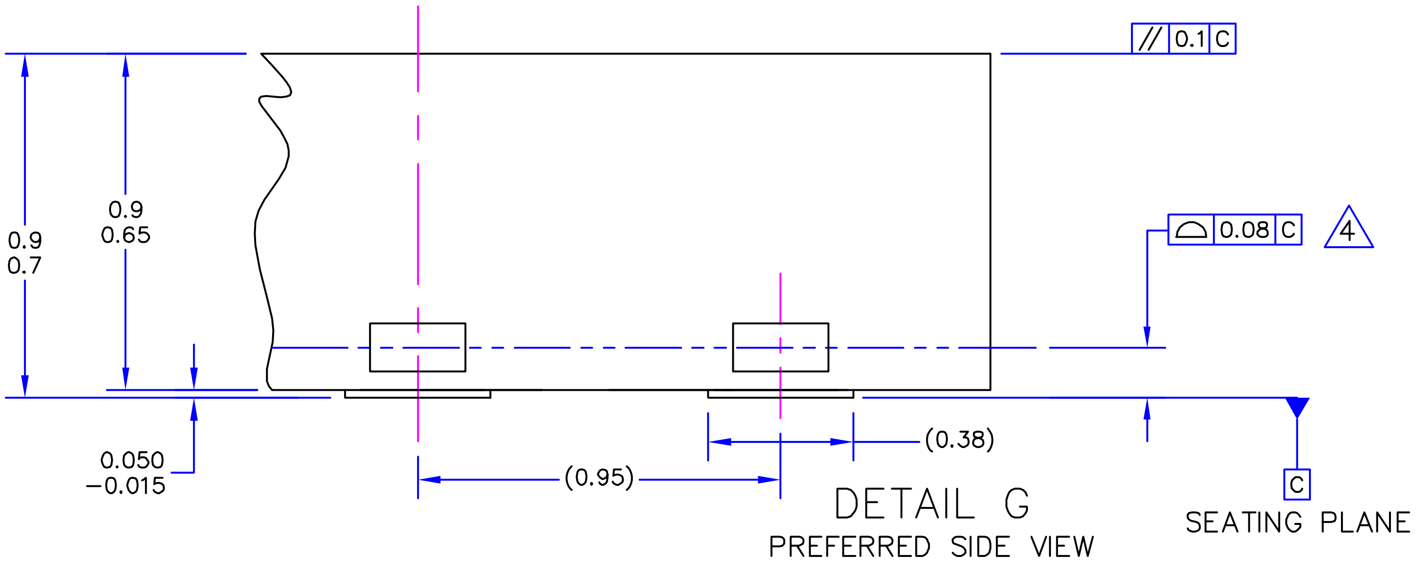
DOCUMENT NO: 98ARL10602D

PAGE: 1677

REV: C



VIEW M–M
PREFERRED BOTTOM VIEW
(NON–EXPOSED LEAD END TYPE)



DETAIL G
PREFERRED SIDE VIEW
(NON–EXPOSED LEAD END TYPE)

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

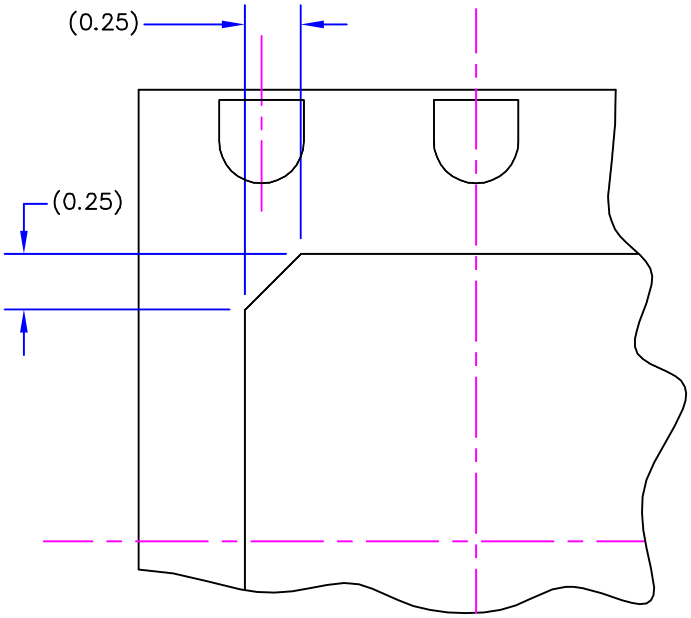
SHEET: 2 OF 6

freescale semiconductor

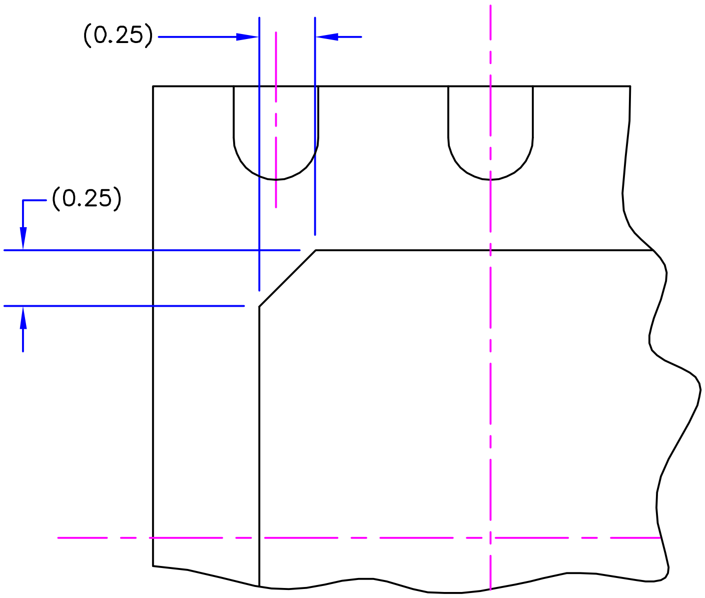
MECHANICAL OUTLINES DICTIONARY

DO NOT SCALE THIS DRAWING

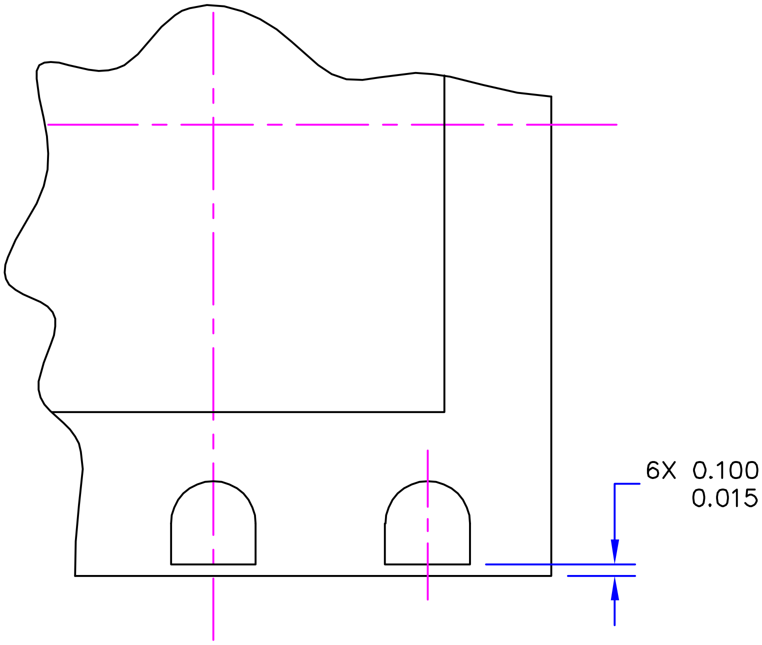
DOCUMENT NO: 98ARL10602D	
PAGE:	1677
REV:	C



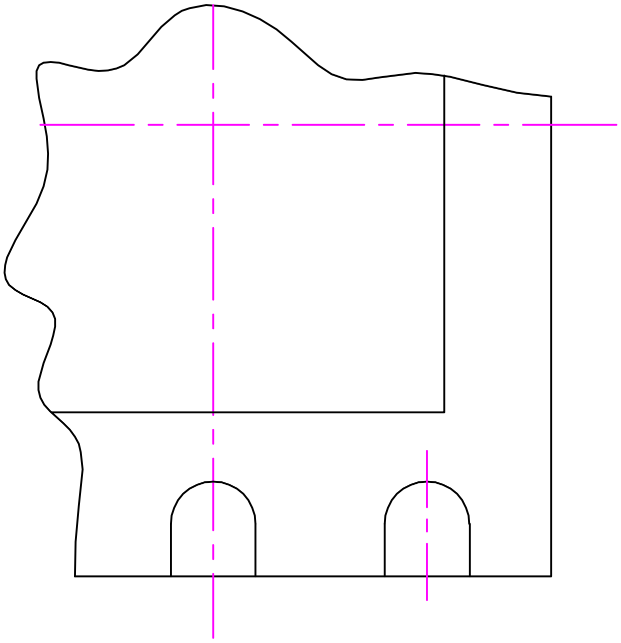
DETAIL M
PREFERRED BACKSIDE PIN 1 INDEX



DETAIL M
BACKSIDE PIN 1 INDEX OPTION



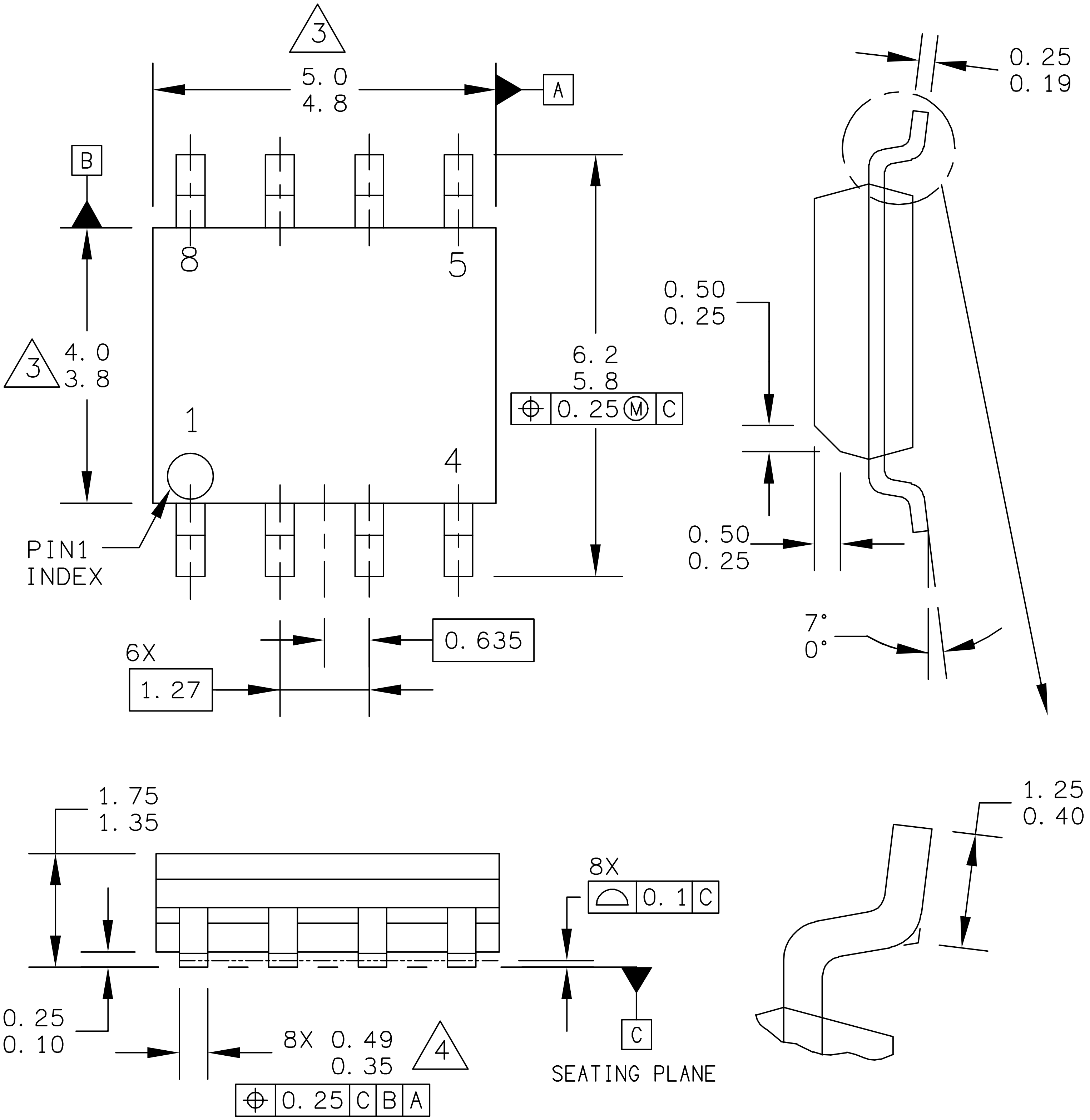
DETAIL N
PREFERRED CORNER CONFIGURATION



DETAIL N
CORNER CONFIGURATION OPTION

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	SHEET: 4 OF 6



© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	**MECHANICAL OUTLINE**	PRINT VERSION NOT TO SCALE
TITLE: 8LD SOIC NARROW BODY	DOCUMENT NO: 98ASB42564B	REV: V
	CASE NUMBER: 751–07	20 NOV 2007
	STANDARD: JEDEC MS–012AA	