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	Obsolete
Core Processor	RS08
Core Size	8-Bit
Speed	10MHz
Connectivity	-
Peripherals	LVD, POR, WDT
Number of I/O	2
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	0°C ~ 70°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/product-detail/nxp-semiconductors/pc9rs08ka2fpe

MC9RS08KA2
MC9RS08KA1

Data Sheet

RS08
Microcontrollers

MC9RS08KA2
Rev. 4
12/2008

freescale.com

freescale™
semiconductor

MC9RS08KA2 Series Data Sheet

Covers: MC9RS08KA2
MC9RS08KA1

MC9RS08KA2
Rev. 4
12/2008

freescale™
semiconductor

- Computer operating properly (COP) timer
- Illegal opcode detect (ILOP)
- Illegal address detect (ILAD)
- Background debug forced reset via BDC command BDC_RESET

Each of these sources, with the exception of the background debug forced reset, has an associated bit in the system reset status register (SRS).

5.4 Computer Operating Properly (COP) Watchdog

The COP watchdog is intended to force a system reset if the application software fails to execute as expected. To prevent a system reset from the COP timer (when it is enabled), application software must reset the COP counter periodically. If the application program gets lost and fails to reset the COP counter before it times out, a system reset is generated to force the system back to a known starting point.

After any reset, the COPE becomes set in SOPT, which enables the COP watchdog (see Section 5.8.2, "System Options Register (SOPT)," for additional information). If the COP watchdog is not used in an application, it can be disabled by clearing COPE. The COP counter is reset by writing any value to the address of SRS. This write does not affect the data in the read-only SRS. Instead, the act of writing to this address is decoded and sends a reset signal to the COP counter.

There is an associated short and long time-out controlled by COPT in SOPT. Table 5-1 summaries the control functions of the COPT bit. The COP watchdog operates from the 1-kHz clock source and defaults to the associated long time-out (2^8 cycles).

Table 5-1. COP Configuration Options

COPT	COP Overflow Count[1]
0	2^5 cycles (32 ms)
1	2^8 cycles (256 ms)

1 Values shown in this column are based on $t_{RTI} \approx 1$ ms. See t_{RTI} in the Section A.9.1, "Control Timing," for the tolerance of this value.

Even if the application will use the reset default settings of COPE and COPT, the user should write to the write-once SOPT registers during reset initialization to lock in the settings. That way, they cannot be changed accidentally if the application program gets lost. The initial write to SOPT will reset the COP counter.

In background debug mode, the COP counter will not increment.

When the MCU enters stop mode, the COP counter is re-initialized to zero upon entry to stop mode. The COP counter begins from zero as soon as the MCU exits stop mode.

5.5 Interrupts

The MC9RS08KA2 Series does not include an interrupt controller with vector table lookup mechanism as used on the HC08 and HCS08 devices. However, the interrupt sources from modules such as LVD, KBI,

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

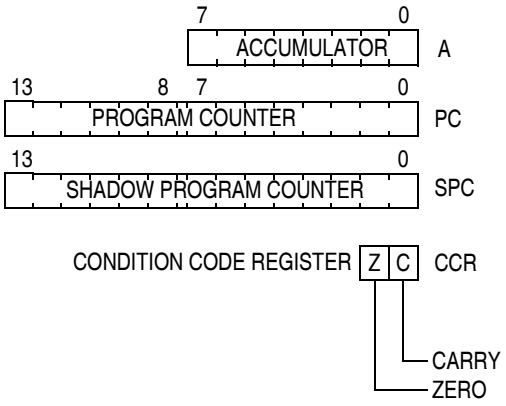


Figure 8-1. CPU Registers

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.

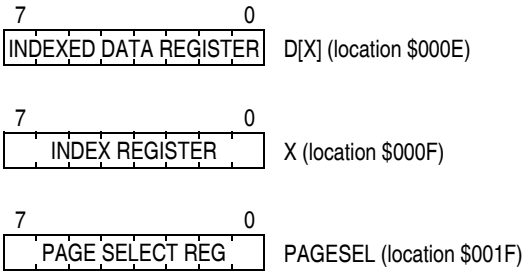


Figure 8-2. Memory Mapped Registers

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.

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.

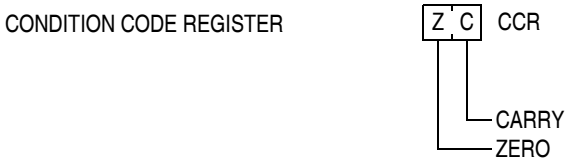


Figure 8-3. Condition Code Register (CCR)

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:
```

8.3.5 Short Addressing Mode (SRT)

SRT addressing mode is capable of addressing only the first 32 bytes in the address map, from $0000 to $001F. This addressing mode is available for CLR, LDA, and STA instructions. A system can be optimized by placing the most computation-intensive data in this area of memory.

Because the 5-bit address is embedded in the opcode, only the least significant five bits of the address must be included in the instruction; this saves program space and execution time. During execution, the CPU adds nine high-order 0s to the 5-bit operand address and uses the combined 14-bit address ($000x or $001x) to access the intended operand.

8.3.6 Direct Addressing Mode (DIR)

DIR addressing mode is used to access operands located in direct address space ($0000 through $00FF).

During execution, the CPU adds six high-order 0s to the low byte of the direct address operand that follows the opcode. The CPU uses the combined 14-bit address ($00xx) to access the intended operand.

8.3.7 Extended Addressing Mode (EXT)

In the extended addressing mode, the 14-bit address of the operand is included in the object code in the low-order 14 bits of the next two bytes after the opcode. This addressing mode is only used in JSR and JMP instructions for jump destination address in RS08 MCUs.

8.3.8 Indexed Addressing Mode (IX, Implemented by Pseudo Instructions)

Indexed addressing mode is sometimes called indirect addressing mode because an index register is used as a reference to access the intended operand.

An important feature of indexed addressing mode is that the operand address is computed during execution based on the current contents of the X index register located in $000F of the memory map rather than being a constant address location that was determined during program assembly. This allows writing of a program that accesses different operand locations depending on the results of earlier program instructions (rather than accessing a location that was determined when the program was written).

The index addressing mode supported by the RS08 Family uses the register X located at $000F as an index and D[X] register located at $000E as the indexed data register. By programming the index register X, any location in the direct page can be read/written via the indexed data register D[X].

These pseudo instructions can be used with all instructions supporting direct, short, and tiny addressing modes by using the D[X] as the operand.

8.4 Special Operations

Most of what the CPU does is described by the instruction set, but a few special operations must be considered, such as how the CPU starts at the beginning of an application program after power is first applied. After the program begins running, the current instruction normally determines what the CPU will do next. Two exceptional events can cause the CPU to temporarily suspend normal program execution:

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

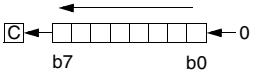
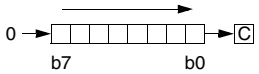
Table 8-1. Instruction Set Summary (Sheet 4 of 6)

Source Form	Description	Operation	Effect on CCR		Address Mode	Opcode	Operand	Cycles
			Z	C				
BSR *rel*	Branch Subroutine	PC ← (PC) + 2 Push PC to shadow PC PC ← (PC) + *rel*	—	—	REL	AD	rr	3
CBEQA #*opr8i*,*rel* CBEQ *opr8a*,*rel* CBEQ ,X,*rel* (1),(2) CBEQ X,*rel* (1)	Compare and Branch if Equal	PC ← (PC) + $0003 + *rel*, if (A) – (M) = $00 PC ← (PC) + $0003 + *rel*, if (A) – (M) = $00 PC ← (PC) + $0003 + *rel*, if (A) – (X) = $00	—	—	IMM DIR IX DIR	41 31 31 31	ii rr dd rr 0E rr 0F rr	4 5 5 5
CLC	Clear Carry Bit	C ← 0	—	0	INH	38		1
CLR *opr8a* CLR *opr5a* CLR ,X (1) CLRA CLRX (1)	Clear	M ← $00 A ← $00 X ← $00	1	—	DIR SRT IX INH INH	3F 8*x* / 9*x* 8E 4F 8F	dd	3 2 2 1 2
CMP #*opr8i* CMP *opr8a* CMP ,X (1) CMP X (1)	Compare Accumulator with Memory	(A) – (M) (A) – (X)	↕	↕	IMM DIR IX INH	A1 B1 B1 B1	ii dd 0E 0F	2 3 3 3
COMA	Complement (One's Complement)	A ← ($\overline{A}$)	↕	1	INH	43		1
DBNZ *opr8a*,*rel* DBNZ ,X,rel (1) DBNZA *rel* DBNZX rel (1)	Decrement and Branch if Not Zero	A ← (A) – $01 or M ← (M) - $01 PC ← (PC) + $0003 + *rel* if (result) ≠ 0 for DBNZ direct PC ← (PC) + $0002 + *rel* if (result) ≠ 0 for DBNZA X ← (X) – $01 PC ← (PC) + $0003 + *rel* if (result) ≠ 0	—	—	DIR IX INH INH	3B 3B 4B 3B	dd rr 0E rr rr 0F rr	7 7 4 7
DEC *opr8a* DEC *opr4a* DEC ,X (1) DECA DEC X	Decrement	M ← (M) – $01 A ← (A) – $01 X ← (X) – $01	↕	—	DIR TNY IX INH DIR	3A 5*x* 5E 4A 5F	dd	5 4 4 1 4
EOR #*opr8i* EOR *opr8a* EOR ,X (1) EOR X	Exclusive OR Memory with Accumulator	A ← (A ⊕ M) A ← (A ⊕ X)	↕	—	IMM DIR IX DIR	A8 B8 B8 B8	ii dd 0E 0F	2 3 3 3
INC *opr8a* INC *opr4a* INC ,X (1) INCA INCX (1)	Increment	M ← (M) + $01 A ← (A) + $01 X ← (X) + $01	↕	—	DIR TNY IX INH INH	3C 2*x* 2E 4C 2F	dd	5 4 4 1 4
JMP *opr16a*	Jump	PC ← Effective Address	—	—	EXT	BC	hh ll	4
JSR *opr16a*	Jump to Subroutine	PC ← (PC) + 3 Push PC to shadow PC PC ← Effective Address	—	—	EXT	BD	hh ll	4
LDA #*opr8i* LDA *opr8a* LDA *opr5a* LDA ,X (1)	Load Accumulator from Memory	A ← (M)	↕	—	IMM DIR SRT IX	A6 B6 C*x*/D*x* CE	ii dd	2 3 3 3

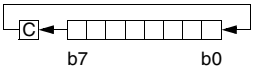
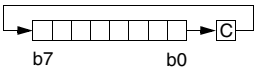
1. This is a pseudo instruction supported by the normal RS08 instruction set.

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 *oprr8a,oprr8a*
MOV #*opr8i,oprr8a*
MOV D[X],*oprr8a*
MOV *oprr8a*,D[X]
MOV #*opr8i*,D[X]
	Move	$(M)_{destination} ← (M)_{source}$	↕	—	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 E*x* / F*x* 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.

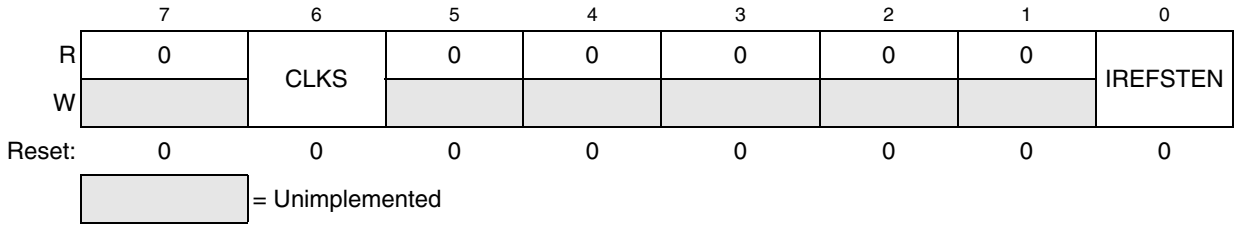


Figure 9-3. ICS Control Register 1 (ICSC1)

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)

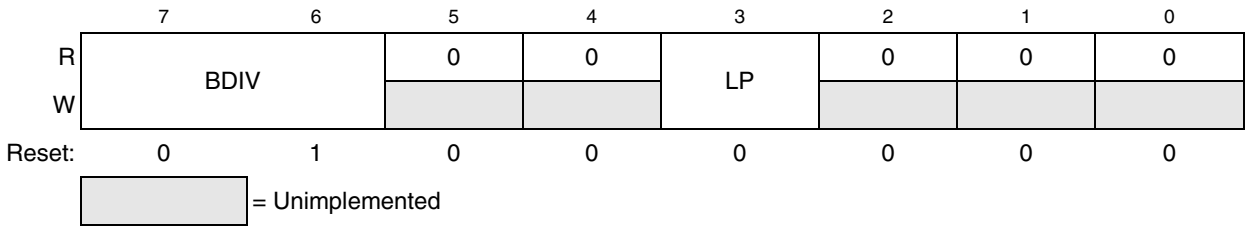


Figure 9-4. ICS Control Register 2 (ICSC2)

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

9.3.3 ICS Trim Register (ICSTRM)

	7	6	5	4	3	2	1	0
R W	TRIM							
POR:	1	0	0	0	0	0	0	0
Reset:	U	U	U	U	U	U	U	U

Figure 9-5. ICS Trim Register (ICSTRM)

Table 9-4. ICSTRM Field Descriptions

Field	Description
7:0 TRIM	**ICS Trim Setting** — The TRIM bits control the internal reference clock frequency by controlling the internal reference clock period. The bits' effect are binary weighted (i.e., bit 1 will adjust twice as much as bit 0). Increasing the binary value in TRIM will increase the period, and decreasing the value will decrease the period. An additional fine trim bit is available in ICSSC as the FTRIM bit.

9.3.4 ICS Status and Control (ICSSC)

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	CLKST	0	FTRIM
W								
POR:	0	0	0	0	0	0	0	0
Reset:	0	0	0	0	0	0	0	U

= Unimplemented

Figure 9-6. ICS Status and Control Register (ICSSC)

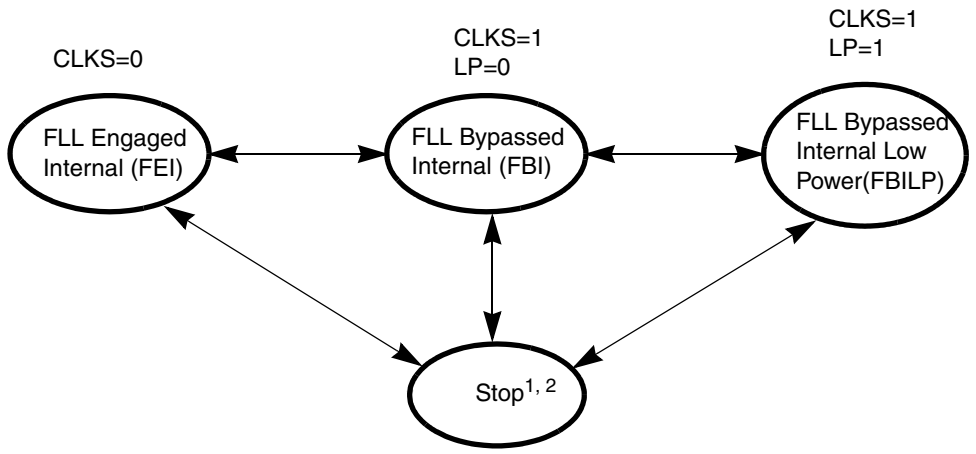
Table 9-5. ICSSC Field Descriptions

Field	Description
2 CLKST	**Clock Mode Status** — The CLKST read-only bit indicate the current clock mode. The CLKST bit does not update immediately after a write to the CLKS bit due to internal synchronization between clock domains. 0 Output of FLL is selected 1 Internal reference clock is selected
0 FTRIM	**ICS Fine Trim** — The FTRIM bit controls the smallest adjustment of the internal reference clock frequency. Setting FTRIM will increase the period and clearing FTRIM will decrease the period by the smallest amount possible.

9.4 Functional Description

9.4.1 Operational Modes

The states of the ICS are shown as a state diagram and are described in this section. The arrows indicate the allowed movements between the states.



[1] ICS enters its Stop state when MCU enters stop, FLL is always disabled. ICS returns to the state that was active before MCU entered stop, unless a reset occurs while in stop.
[2] If IREFSTEN is set when MCU enters stop, the ICSIRCLK remains running.

Figure 9-7. Clock Switching Modes

9.4.1.1 FLL Engaged Internal (FEI)

FLL engaged internal (FEI) is the default mode of operation out of any reset and is entered when CLKS is written to 0.

In FLL engaged internal mode, the ICSOUT clock is derived from the FLL clock, which is controlled by the internal reference clock. The FLL loop will lock the frequency to 512 times the filter frequency.

9.4.1.2 FLL Bypassed Internal (FBI)

The FLL bypassed internal (FBI) mode is entered when CLKS is written to 1 and LP bit is a 0.

In FLL bypassed internal mode, the ICSOUT clock is derived from the internal reference clock. The FLL clock is controlled by the internal reference clock, and the FLL loop will lock the FLL frequency to 512 times the filter frequency.

9.4.1.3 FLL Bypassed Internal Low Power (FBILP)

The FLL bypassed internal low power (FBILP) mode is entered when CLKS is written to 1 and LP = 1.

In FLL bypassed internal low power mode, the ICSOUT clock is derived from the internal reference clock and the FLL is disabled.

Table 10-2. ACMPSC Field Descriptions

Field	Description
7 ACME	**Analog Comparator Module Enable** — ACME enables the ACMP module. 0 ACMP not enabled. 1 ACMP is enabled.
6 ACBGS	**Analog Comparator Bandgap Select** — ACBGS is used to select between the internal bandgap reference voltage or the ACMP+ pin as the non-inverting input of the analog comparator. 0 External pin ACMP+ selected as non-inverting input to comparator. 1 Internal bandgap reference voltage selected as non-inverting input to comparator.
5 ACF	**Analog Comparator Flag** — ACF is set when a compare event occurs. Compare events are defined by ACMOD. ACF is cleared by writing a one to ACF. 0 Compare event has not occurred. 1 Compare event has occurred.
4 ACIE	**Analog Comparator Interrupt Enable** — ACIE enables the interrupt for the ACMP. When ACIE is set, an interrupt will be asserted when ACF is set. 0 Interrupt disabled. 1 Interrupt enabled.
3 ACO	**Analog Comparator Output** — Reading ACO will return the current value of the analog comparator output. ACO is reset to a 0 and will read as a 0 when the ACMP is disabled (ACME = 0).
2 ACOPE	**Analog Comparator Output Pin Enable** — ACOPE is used to enable the comparator output to be placed onto the external pin, ACMPO. ACOPE will only control the pin if the ACMP is active (ACME=1). 0 Analog comparator output not available on ACMPO. 1 Analog comparator output is driven out on ACMPO.
1:0 ACMOD	**Analog Comparator Mode** — ACMOD selects the type of compare event which sets ACF. 00 Encoding 0 — Comparator output falling edge. 01 Encoding 1 — Comparator output rising edge. 10 Encoding 2 — Comparator output falling edge. 11 Encoding 3 — Comparator output rising or falling edge.

10.4 Functional Description

The analog comparator can be used to compare two analog input voltages applied to ACMP+ and ACMP–; or it can be used to compare an analog input voltage applied to ACMP– with an internal bandgap reference voltage. ACBGS is used to select between the bandgap reference voltage or the ACMP+ pin as the input to the non-inverting input of the analog comparator.

The comparator output is high when the non-inverting input is greater than the inverting input, and it is low when the non-inverting input is less than the inverting input. ACMOD is used to select the condition which will cause ACF to be set. ACF can be set on a rising edge of the comparator output, a falling edge of the comparator output, or either a rising or a falling edge (toggle). The comparator output can be read directly through ACO. The comparator output can also be driven onto the ACMPO pin using ACOPE.

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.

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.

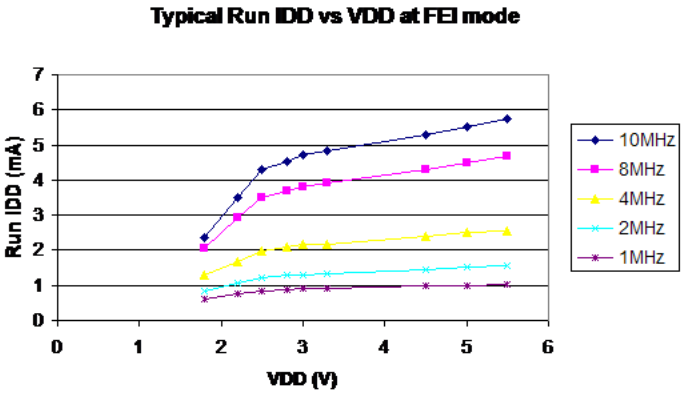


Figure 12-15. Typical Run I_{DD} vs. V_{DD} for FEI mode

A.7 Analog Comparator (ACMP) Electricals

Table A-6. Analog Comparator Electrical Specifications

Characteristic	Symbol	Min	Typical	Max	Unit
Supply voltage	V_{DD}	1.80	—	5.5	V
Analog input voltage	V_{AIN}	V_{SS} – 0.3	—	V_{DD}	V
Analog source impedance	R_{AS}	—	—	10	kΩ
Analog input offset voltage[1]	V_{AIO}	—	20	40	mV
Analog Comparator hysteresis[1]	V_H	3.0	9.0	15.0	mV
Analog Comparator bandgap reference voltage[1]	V_{BG}	1.155	1.190	1.230	V
Supply current (active)[1]	I_{DDAC}	—	20	35	µA
Analog input leakage current[1]	I_{ALKG}	—	—	1.0	µA
Analog Comparator initialization delay[1]	t_{AINIT}	—	—	1.0	µs

1 These data are characterized but not production tested. Measurements are made with the device entered STOP mode.

A.8 Internal Clock Source Characteristics

Table A-7. Internal Clock Source Specifications

Characteristic	Symbol	Min	Typ[1]	Max	Unit
Average internal reference frequency — factory trimmed at V_{DD} = 5 V and temperature = 25°C	f_{int_ft}	—	20	—	MHz
Average internal reference frequency - untrimmed	f_{int_ut}	25	31.25	41.66	kHz
Average internal reference frequency - trimmed	f_{int_t}	31.25	31.25	39.0625	kHz
DCO output frequency range - untrimmed	f_{dco_ut}	12.8	16	21.33	MHz
DCO output frequency range - trimmed	f_{dco_t}	16	16	20	MHz
Resolution of trimmed DCO output frequency at fixed voltage and temperature	$\Delta f_{dco\_res\_t}$	—	—	±0.2	$\%f_{dco}$
Total deviation of trimmed DCO output frequency over voltage and temperature	Δf_{dco_t}	—	—	±2	$\%f_{dco}$
FLL acquisition time[2,3]	$t_{acquire}$	—	—	1	ms

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.

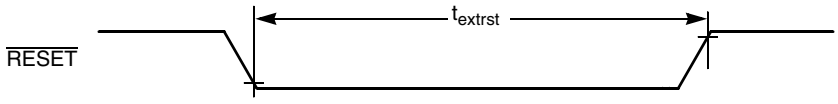
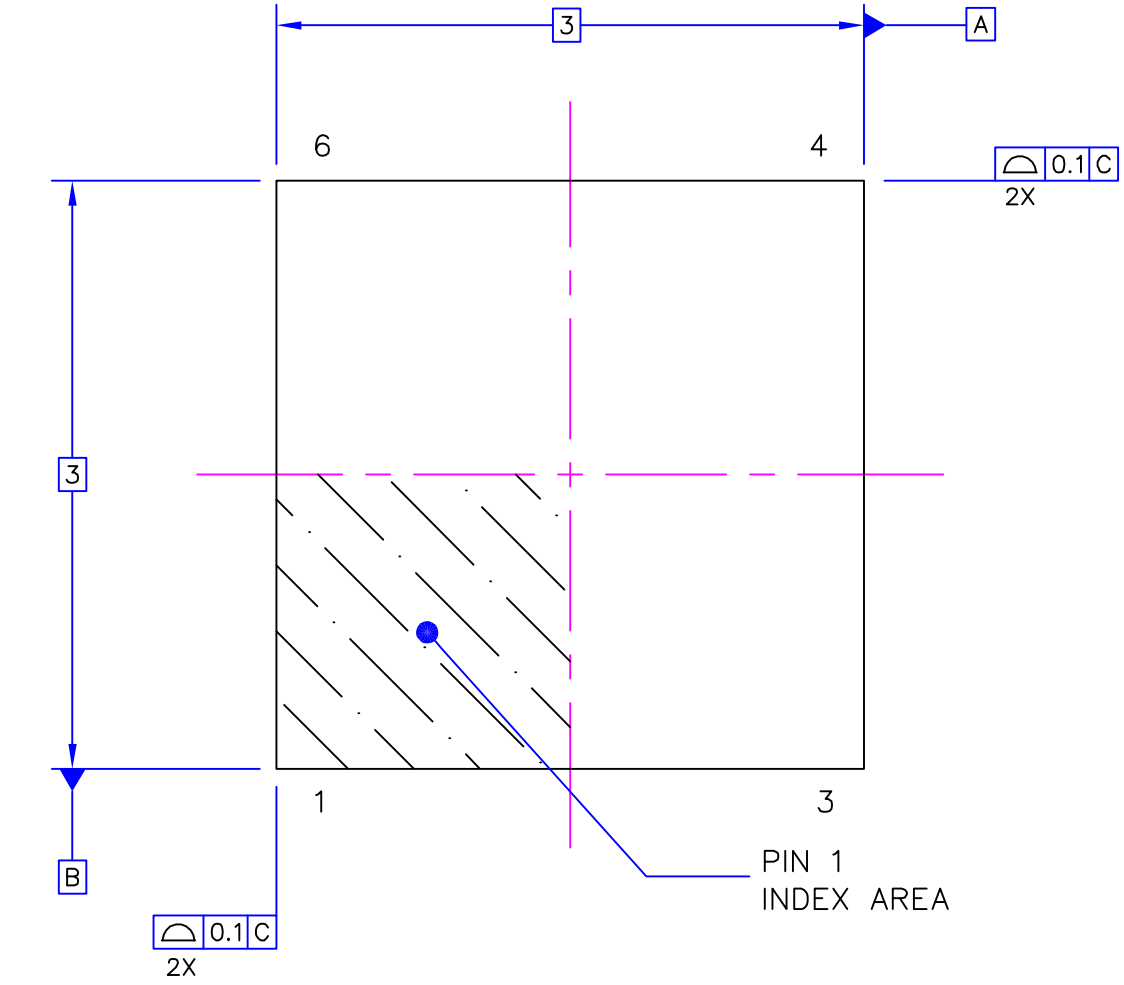
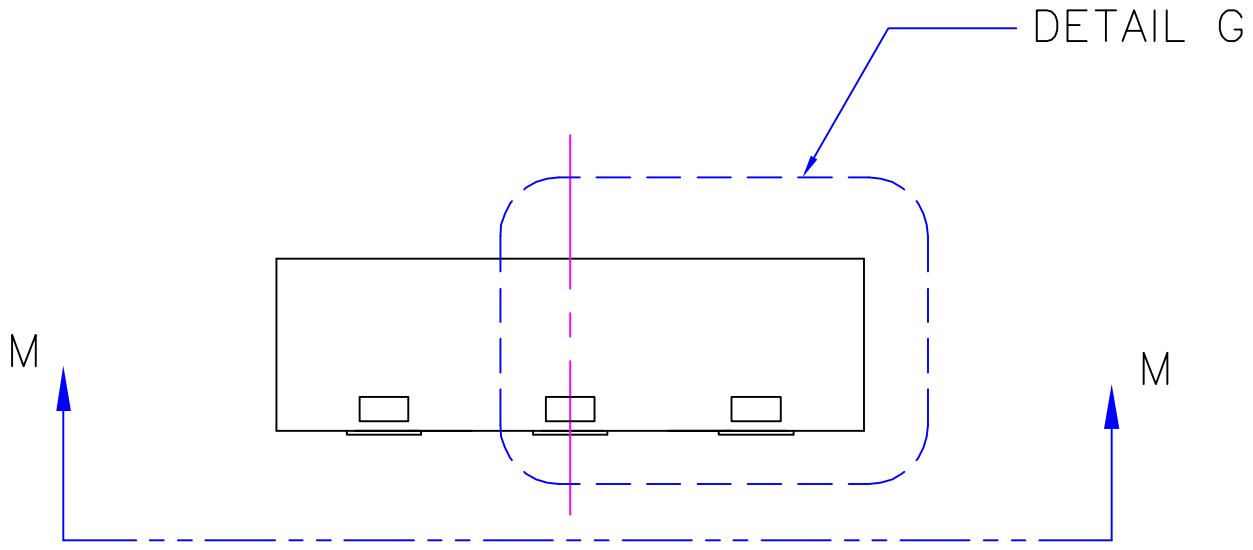


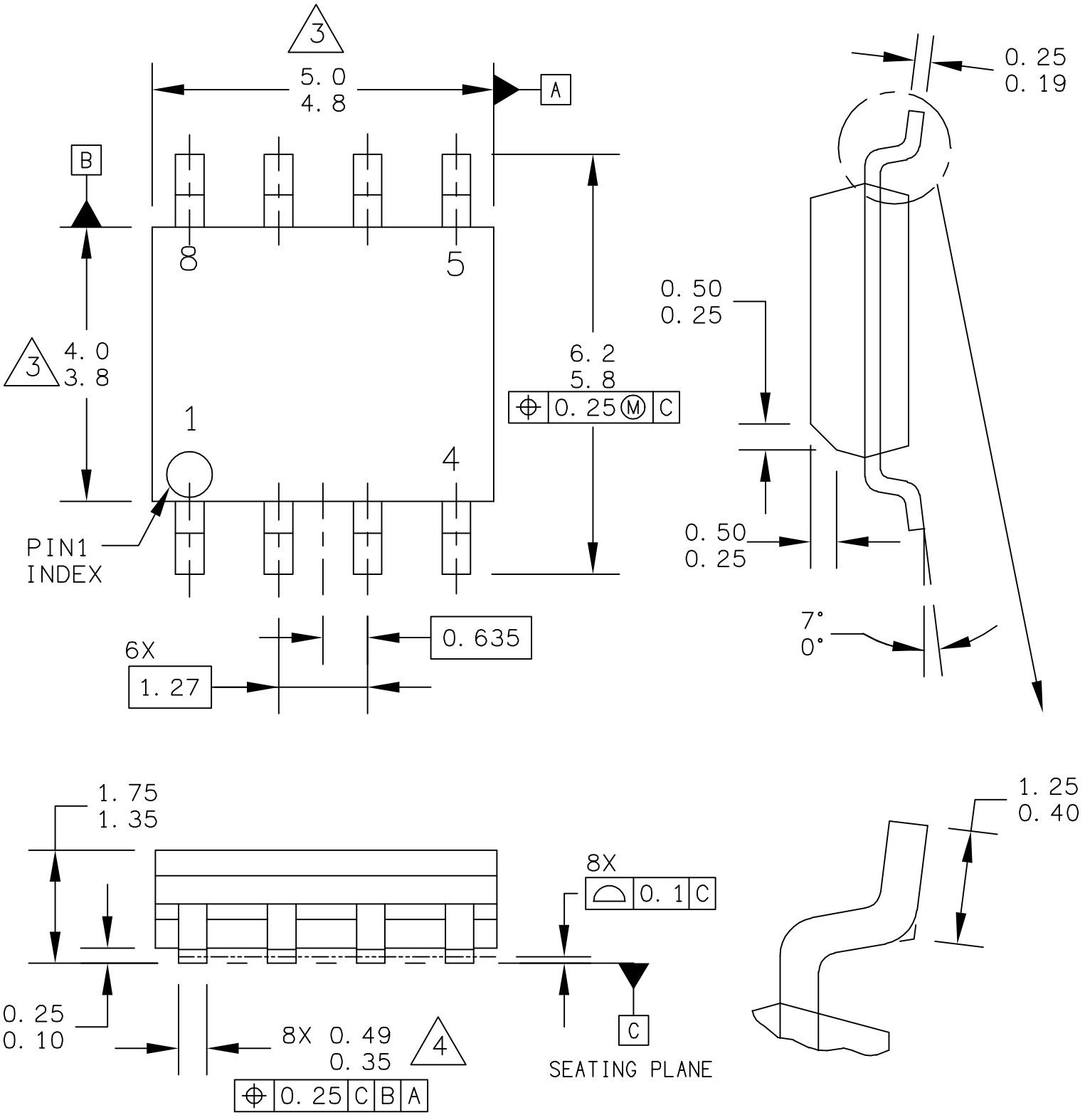
Figure A-1. Reset Timing

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



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	