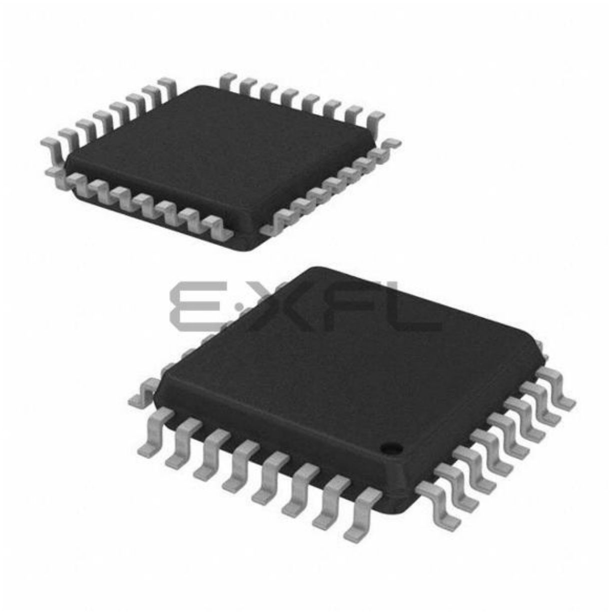
NXP USA Inc. - MCHC908RF2MFAE Datasheet



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	HC08
Core Size	8-Bit
Speed	4MHz
Connectivity	-
Peripherals	LVD, POR, PWM, RF Mod
Number of I/O	12
Program Memory Size	2KB (2K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mchc908rf2mfae

MC68HC908RF2

Data Sheet

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify you have the latest information available, refer to:

http://motorola.com/semiconductors

The following revision history table summarizes changes contained in this document. For your convenience, the page number designators have been linked to the appropriate location.

Section 1. General Description

1.1 Introduction

The MC68HC908RF2 MCU is a member of the low-cost, high-performance M68HC08 Family of 8-bit microcontroller units (MCUs). Optimized for low-power operation and available in a small 32-pin low-profile quad flat pack (LQFP), this MCU is well suited for remote keyless entry (RKE) transmitter designs.

All MCUs in the family use the enhanced M68HC08 central processor unit (CPU08) and are available with a variety of modules, memory sizes and types, and package types.

1.2 Features

Features MCU include:

- High-performance M68HC08 architecture
- Fully upward-compatible object code with M6805, M146805, and M68HC05 Families
- Maximum internal bus frequency of 4 MHz at 3.3 volts
- Maximum internal bus frequency of 2 MHz at 1.8 volts
- Internal oscillator requiring no external components:
  - Software selectable bus frequencies
  - ± 25 percent accuracy with trim capability to ±2 percent
  - Option to allow use of external clock source or external crystal/ceramic resonator
- 2 Kbytes of on-chip FLASH memory
- FLASH program memory security[1]
- 128 bytes of on-chip RAM
- 16-bit, 2-channel timer interface module (TIM)

[1] No security feature is absolutely secure. However, Motorola’s strategy is to make reading or copying the FLASH difficult for unauthorized users.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0028	Timer Channel 1 Status and Control Register (TSC1)	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0		▨					
	See page 143.	Reset:	0	0	0	0	0	0	0	0
$0029	Timer Channel 1 Register High (TCH1H)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
	See page 146.	Reset:	Indeterminate after reset							
$002A	Timer Channel 1 Register Low (TCH1L)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
	See page 146.	Reset:	Indeterminate after reset							
$002B ↓ $0035	Unimplemented		▨	▨	▨	▨	▨	▨	▨	▨
$0036	Internal Clock Generator Control Register (ICGCR)	Read:	CMIE	CMF	CMON	CS	ICGON	ICGS	ECGON	ECGS
		Write:		▨				▨		▨
	See page 88.	Reset:	0	0	0	0	1	0	0	0
$0037	Internal Clock Generator Multiplier Register (ICGMR)	Read:	R	N6	N5	N4	N3	N2	N1	N0
		Write:								
	See page 90.	Reset:	0	0	0	1	0	1	0	1
$0038	Internal Clock Generator Trim Register (ICGTR)	Read:	TRIM7	TRIM6	TRIM5	TRIM4	TRIM3	TRIM2	TRIM1	TRIM0
		Write:								
	See page 90.	Reset:	1	0	0	0	0	0	0	0
$0039	ICG DCO Divider Control Register (ICGDVR)	Read:	R	R	R	R	DDIV3	DDIV2	DDIV1	DDIV0
		Write:								
	See page 91.	Reset:	0	0	0	0	U	U	U	U
$003A	ICG DCO Stage Register (ICGDSR)	Read:	DSTG7	DSTG6	DSTG5	DSTG4	DSTG3	DSTG2	DSTG1	DSTG0
		Write:								
	See page 91.	Reset:	Unaffected by reset							
$003B	Reserved		R	R	R	R	R	R	R	R
$003C ↓ $003F	Unimplemented		▨	▨	▨	▨	▨	▨	▨	▨
$FE00	SIM Break Status Register (SBSR)	Read:	R	R	R	R	R	R	SBSW	R
		Write:							See Note	
	See page 126.	Reset:							0	

Note: Writing a 0 clears SBSW

▨ = Unimplemented R = Reserved U = Unaffected X = Indeterminate

Figure 2-2. Control, Status, and Data Registers (Sheet 3 of 5)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE01	SIM Reset Status Register (SRSR) See page 127.	Read:	POR	PIN	COP	ILOP	ILAD	0	LVI	0
		Write:	▨	▨	▨	▨	▨	▨	▨	▨
		POR:	1	X	X	X	X	X	X	X
$FE02	SIM Break Flag Control Register (SBFCR) See page 128.	Read:	BCFE	R	R	R	R	R	R	R
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE03 ↓ $FE04	Reserved		R	R	R	R	R	R	R	R
$FE05 ↓ $FE07	Unimplemented		▨	▨	▨	▨	▨	▨	▨	▨
$FE08	FLASH 2TS Control Register (FLCR) See page 29.	Read:	0	FDIV0	BLK1	BLK0	HVEN	MARGIN	ERASE	PGM
		Write:	▨							
		Reset:	0	0	0	0	0	0	0	0
$FE09	Reserved		R	R	R	R	R	R	R	R
$FE0A ↓ $FE0B	Unimplemented		▨	▨	▨	▨	▨	▨	▨	▨
$FE0C	Break Address Register High (BRKH) See page 161.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0D	Break Address Register Low (BRKL) See page 161.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0E	Break Status and Control Register (BSCR) See page 160.	Read:	BRKE	BRKA	0	0	0	0	0	0
		Write:			▨	▨	▨	▨	▨	▨
		Reset:	0	0	0	0	0	0	0	0
$FE0F	LVI Status Register (LVISR) See page 105.	Read:	LVIOUT	0	LOWV	0	0	0	0	0
		Write:	▨	▨	▨	▨	▨	▨	▨	▨
		Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented   R = Reserved   U = Unaffected   X = Indeterminate

Figure 2-2. Control, Status, and Data Registers (Sheet 4 of 5)

Program and erase operations are facilitated through control bits in a memory mapped register. Details for these operations appear later in this section. Memory in the FLASH 2TS array is organized into pages within rows. For the 2-Kbyte array on the MC68HC908RF2, the page size is one byte. There are eight pages (or eight bytes) per row. Programming operations are performed on a page basis, one byte at a time. Erase operations are performed on a block basis. The minimum block size is one row of eight bytes. Refer to **Table 2-3** for additional block size options.

NOTE: *Sometimes a program disturb condition, in which case an erased bit on the row being programmed unintentionally becomes programmed, occurs. The embedded smart programming algorithm implements a margin read technique to avoid program disturb. The margin read step of the smart programming algorithm is used to ensure programmed bits are programmed to sufficient margin for data retention over the device's lifetime. In the application code, perform an erase operation after eight program operations (on the same row) to further avoid program disturb.*

For availability of programming tools and more information, contact a local Motorola representative.

NOTE: *A security feature prevents viewing of the FLASH 2TS contents.*[1]

2.5.1 FLASH 2TS Control Register

The FLASH 2TS control register (FLCR) controls program, erase, and margin read operations.

Address: $FE08

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	FDIV0	BLK1	BLK0	HVEN	MARGIN	ERASE	PGM
Write:	▨							
Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented

Figure 2-3. FLASH 2TS Control Register (FLCR)

FDIV0 — Frequency Divide Control Bit

This read/write bit selects the factor by which the charge pump clock is divided from the system clock. See **2.5.2 FLASH 2TS Charge Pump Frequency Control**.

BLK1 — Block Erase Control Bit

This read/write bit together with BLK0 allows erasing of blocks of varying size. See **2.5.3 FLASH 2TS Erase Operation** for a description of available block sizes.

[1] No security feature is absolutely secure. However, Motorola's strategy is to make reading or copying the FLASH 2TS difficult for unauthorized users.

Page Program/Margin Read Procedure

Note: This algorithm is mandatory for programming the FLASH 2TS.

Note: This page program algorithm assumes the page/s to be programmed are initially erased.

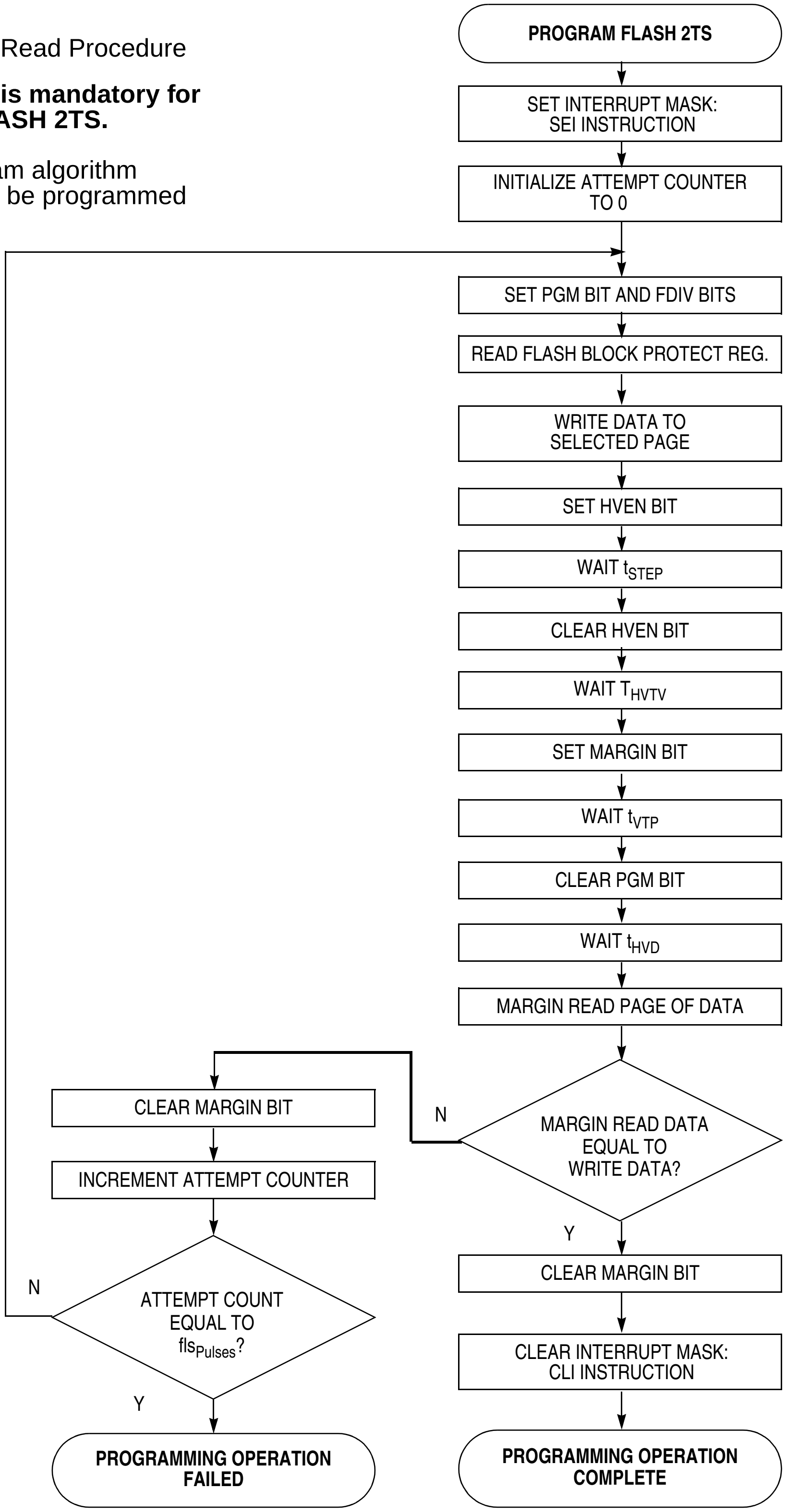


Figure 2-4. Smart Programming Algorithm Flowchart

The row whose cycling bit is different will be erased and the entire row will be programmed with the given data, including a toggled version of the cycling bit.

2.5.8.5 Example Routine Calls

This code is for illustrative purposes only and does not represent valid syntax for any particular assembler.

```
RAM             EQU     $80
RDVRRNG         EQU     $F000
PRGRNGE         EQU     $F003
ERANRGE         EQU     $F006
REDPROG         EQU     $F009
;*************************************************************
 RAM Definitions for Subroutines
;**************************************************************

ORG     RAM+8

CTRLBYT         RMB     1
CPUSPD          RMB     1
LADDR           RMB     2
BUMPS           RMB     1
DERASE          RMB     2

;Allocation of "DATA" space is dependent on the device and
;application

DATA   RMB     8
;*************************************************************
; CALLING EXAMPLE FOR READ/VERIFY A RANGE (RDVRRNG)
;**************************************************************
LDA    #$FF             ;TARGET IS RAM
LDHX    #$7807          ;END AFTER FIRST ROW
STHX    LADDR
LDHX    #$7800          ;START AT FIRST ROW
JSR     RDVRRNG         ;DATA WILL CONTAIN FLASH INFO
;************************************************************;
CALLING EXAMPLE FOR ERASE A RANGE (RNGEERA)
;**************************************************************
MOV    #$08,CPUSPD      ;Load Bus frequency in MHz * 4
MOV    #$60,CTRLBYT     ;Bits 5&6 hold the block size to erase
                        ;00  Full Array
                        ;20  One-Half Array
                        ;40  Eight Rows
                        ;60  Singe Row
                        ;Remember a Row is 1 byte

                        ;Set erase time in uS/24, number in
                        ;decimal
LDHX   #100000/24
STHX   DERASE
```

STOP — STOP Instruction Enable Bit

STOP enables the STOP instruction.

1 = STOP instruction enabled
0 = STOP instruction treated as illegal opcode

COPD — COP Disable Bit

COPD disables the COP module. (See **Section 4. Computer Operating Properly Module (COP)**.)

1 = COP module disabled
0 = COP module enabled

The method of changing the unadjusted operating point is by changing the size of the capacitor. This capacitor is designed with 639 equally sized units, 384 of which are always connected. The remaining 255 units are put in by adjusting the ICG trim factor (TRIM). The default value for TRIM is $80, or 128 units, making the default capacitor size 512. Each unit added or removed will adjust the output frequency by about ±0.195 percent of the unadjusted frequency (adding to TRIM will decrease frequency). Therefore, the frequency of IBASE can be changed to ±25 percent of its unadjusted value, which is enough to cancel the process variability mentioned before.

The best way to trim the internal clock is to use the timer to measure the width of an input pulse on an input capture pin (this pulse must be supplied by the application and should be as long or wide as possible). Considering the prescale value of the timer and the theoretical (zero error) frequency of the bus (307.2 kHz *N/4), the error can be calculated. This error, expressed as a percentage, can be divided by 0.195 percent and the resultant factor added or subtracted from TRIM. This process should be repeated to eliminate any residual error.

NOTE: *It is recommended that the user preserve a copy of the contents of the ICG trim register (ICGTR) in non-volitale memory.*

Address $7FEF is reserved for an optional factory-determined value. Consult with a local Motorola representative for more information and availability of this option.

6.5 Low-Power Modes

The WAIT and STOP instructions put the MCU in low power-consumption standby modes.

6.5.1 Wait Mode

The ICG remains active in wait mode. If enabled, the ICG interrupt to the CPU can bring the MCU out of wait mode.

In some applications, low-power consumption is desired in wait mode and a high-frequency clock is not needed. In these applications, reduce power consumption by either selecting a low-frequency external clock and turning the internal clock generator off, or reducing the bus frequency by minimizing the ICG multiplier factor (N) before executing the WAIT instruction.

6.5.2 Stop Mode

The ICG is disabled in stop mode. Upon execution of the STOP instruction, all ICG activity will cease and the output clocks (CGMXCLK and CGMOUT) will be held low. Power consumption will be minimal.

The STOP instruction does not affect the values in the ICG registers. Normal execution will resume after the MCU exits stop mode.

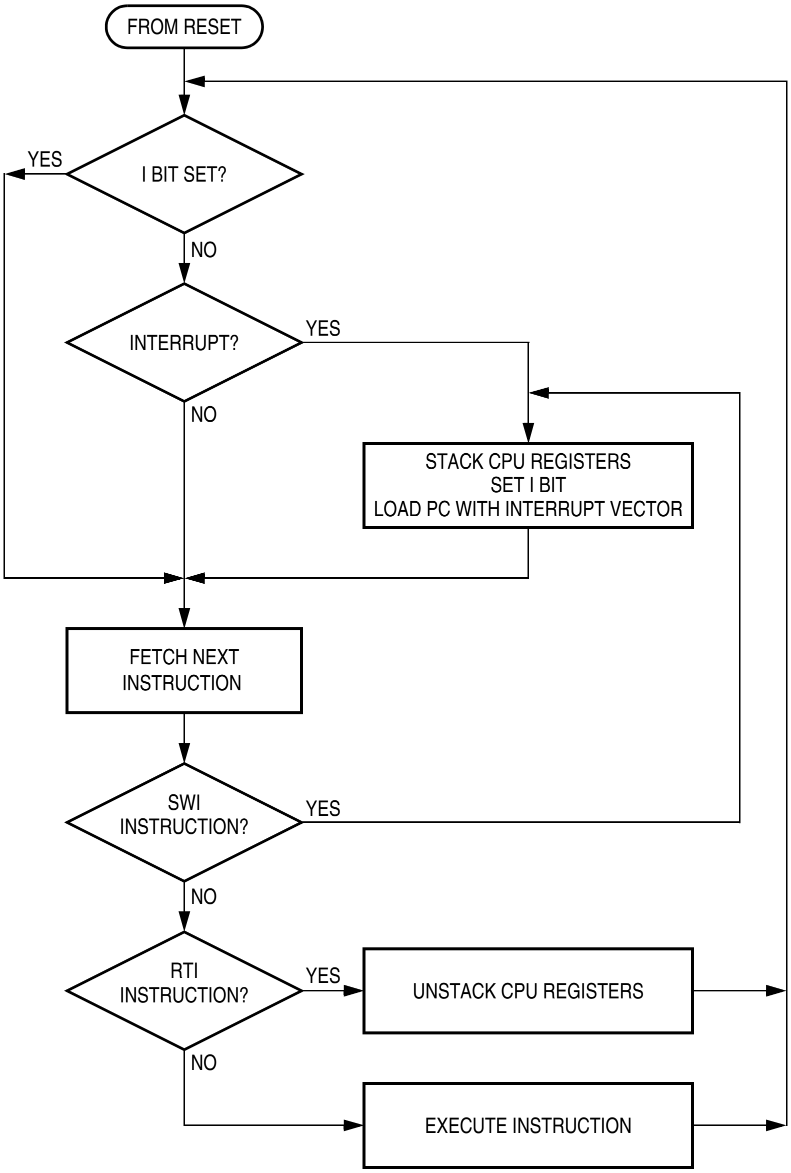


Figure 7-4. IRQ Interrupt Flowchart

If the MODEI bit is clear, the $\overline{\text{IRQ}}$ pin is falling-edge sensitive only. With MODEI clear, a vector fetch or software clear immediately clears the IRQ1 latch.The IRQ1F bit in the INTKBSCR register can be used to check for pending interrupts. The IRQ1F bit is not affected by the IMASKI bit, which makes it useful in applications where polling is preferred.

Use the BIH or BIL instruction to read the logic level on the $\overline{\text{IRQ}}$ pin.

NOTE: *When using the level-sensitive interrupt trigger, avoid false interrupts by masking interrupt requests in the interrupt routine.*

signal then follows the sequence from the falling edge of $\overline{\text{RST}}$ shown in **Figure 10-5**.

The COP reset is asynchronous to the bus clock.

The active reset feature allows the part to issue a reset to peripherals and other chips within a system built around the MCU.

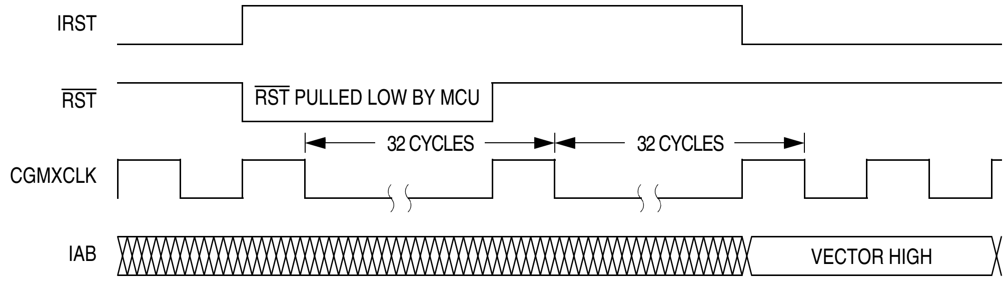


Figure 10-5. Internal Reset Timing

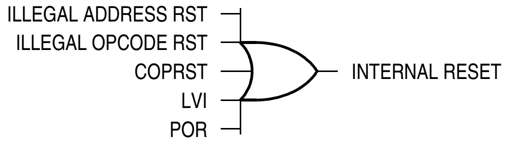


Figure 10-6. Sources of Internal Reset

10.3.2.1 Power-On Reset

When power is first applied to the MCU, the power-on reset module (POR) generates a pulse to indicate that power-on has occurred. The external reset pin ($\overline{\text{RST}}$) is held low while the SIM counter counts out 4096 CGMXCLK cycles. Another 64 CGMXCLK cycles later, the CPU and memories are released from reset to allow the reset vector sequence to occur.

At power-on, these events occur:

- A POR pulse is generated.
- The internal reset signal is asserted.
- The SIM enables CGMOUT.
- Internal clocks to the CPU and modules are held inactive for 4096 CGMXCLK cycles to allow stabilization of the oscillator.
- The $\overline{\text{RST}}$ pin is driven low during the oscillator stabilization time.
- The POR bit of the SIM reset status register (SRSR) is set and all other bits in the register are cleared.

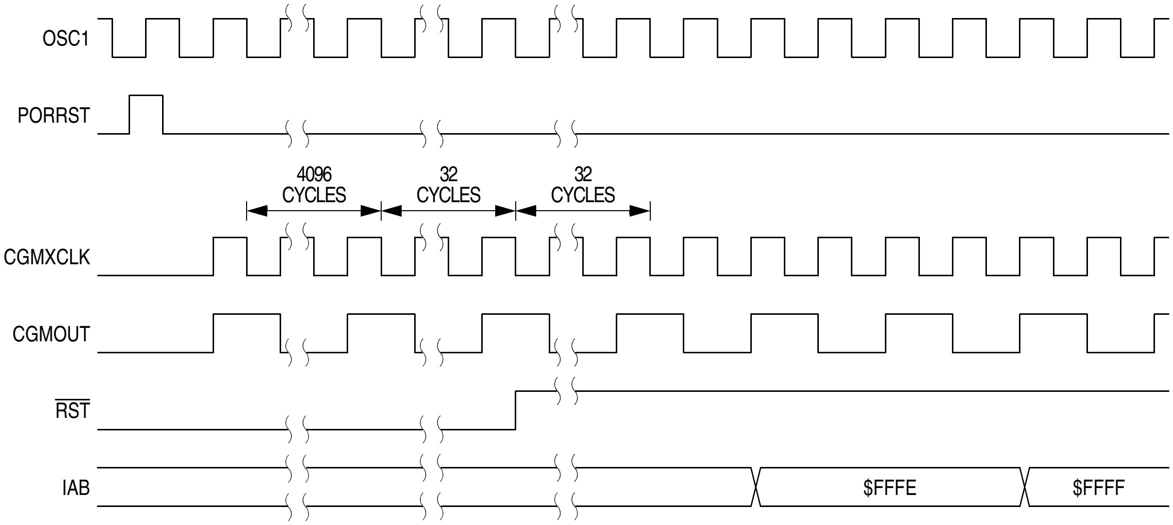


Figure 10-7. POR Recovery

10.3.2.2 Computer Operating Properly (COP) Reset

The overflow of the COP counter causes an internal reset and sets the COP bit in the SIM reset status register (SRSR) if the COPD bit in the CONFIG register is at 0. (See **Section 4. Computer Operating Properly Module (COP)**.)

10.3.2.3 Illegal Opcode Reset

The SIM decodes signals from the CPU to detect illegal instructions. An illegal instruction sets the ILOP bit in the SIM reset status register (SRSR) and causes a reset.

If the stop enable bit, STOP, in the configuration register (CONFIG) is 0, the SIM treats the STOP instruction as an illegal opcode and causes an illegal opcode reset.

10.3.2.4 Illegal Address Reset

An opcode fetch from an unmapped address generates an illegal address reset. The SIM verifies that the CPU is fetching an opcode prior to asserting the ILAD bit in the SIM reset status register (SRSR) and resetting the MCU. A data fetch from an unmapped address does not generate a reset.

10.3.2.5 Low-Voltage Inhibit (LVI) Reset

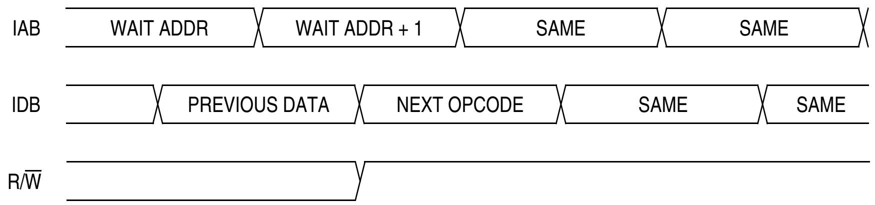
The low-voltage inhibit module (LVI) asserts its output to the SIM when the V_{DD} voltage falls to the V_{LVR} voltage. The LVI bit in the SIM reset status register (SRSR) is set and a chip reset is asserted if the LVIPWRD and LVIRSTD bits in the CONFIG register are at 0. The $\overline{RST}$ pin will be held low until the SIM counts 4096 CGMXCLK cycles after V_{DD} rises above V_{LVR}+ H_{LVR}. Another 64 CGMXCLK

10.6.1 Wait Mode

In wait mode, the CPU clocks are inactive while one set of peripheral clocks continues to run. Figure 10-12 shows the timing for wait mode entry.

A module that is active during wait mode can wake up the CPU with an interrupt if the interrupt is enabled. Stacking for the interrupt begins one cycle after the WAIT instruction during which the interrupt occurred. Refer to the wait mode subsection of each module to see if the module is active or inactive in wait mode. Some modules can be programmed to be active in wait mode.

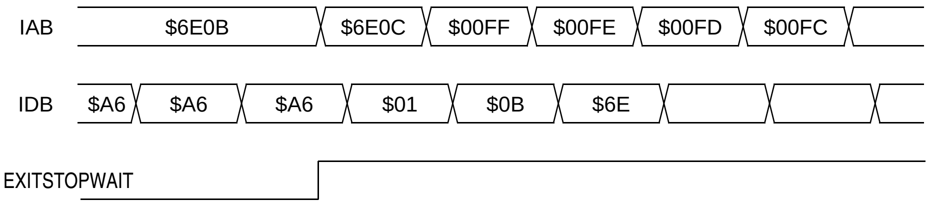
Wait mode can also be exited by a reset or break. A break interrupt during wait mode sets the SIM break stop/wait bit, SBSW, in the SIM break status register (SBSR). If the COP disable bit, COPD, in the mask option register is 0, then the computer operating properly module (COP) is enabled and remains active in wait mode.



Note: Previous data can be operand data or the WAIT opcode, depending on the last instruction.

Figure 10-12. Wait Mode Entry Timing

Figure 10-13 and Figure 10-14 show the timing for WAIT recovery.



Note: EXITSTOPWAIT = $\overline{\text{RST}}$ pin or CPU interrupt or break interrupt

Figure 10-13. Wait Recovery from Interrupt or Break

10.7.2 SIM Reset Status Register

This register contains six flags that show the source of the last reset. The status register will clear automatically after reading it. A power-on reset sets the POR bit.

Address: $FE01

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	POR	PIN	COP	ILOP	ILAD	0	LVI	0
Write:	▨	▨	▨	▨	▨	▨	▨	▨
POR:	1	X	X	X	X	X	X	X

▨ = Unimplemented X = Indeterminate

Figure 10-18. SIM Reset Status Register (SRSR)

POR — Power-On Reset Bit
1 = Last reset caused by POR circuit
0 = Read of SRSR

PIN — External Reset Bit
1 = Last reset caused by external reset pin ($\overline{\text{RST}}$)
0 = Read of SRSR

COP — Computer Operating Properly Reset Bit
1 = Last reset caused by COP counter
0 = Read of SRSR

ILOP — Illegal Opcode Reset Bit
1 = Last reset caused by an illegal opcode
0 = Read of SRSR

ILAD — Illegal Address Reset Bit (opcode fetches only)
1 = Last reset caused by an opcode fetch from an illegal address
0 = Read of SRSR

LVI — Low-Voltage Inhibit Reset Bit
1 = Last reset was caused by the LVI circuit
0 = Read of SRSR

11.7.2 TIM Channel I/O Pins (TCH0)

The channel I/O pins are programmable independently as an input capture pin or an output compare pin. TCH0 can be configured as buffered output compare or buffered PWM pins.

11.8 I/O Registers

These I/O registers control and monitor operation of the TIM:

- TIM status and control register, TSC
- TIM control registers, TCNTH and TCNTL
- TIM counter modulo registers, TMODH and TMODL
- TIM channel status and control registers, TSC0 and TSC1
- TIM channel registers, TCH0H, TCH0L, TCH1H, and TCH1L

11.8.1 TIM Status and Control Register

The TIM status and control register (TSC):

- Enables TIM overflow interrupts
- Flags TIM overflows
- Stops the TIM counter
- Resets the TIM counter
- Prescales the TIM counter clock

Address: $0020

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
Write:	0			TRST	▨			
Reset:	0	0	1	0	0	0	0	0

▨ = Unimplemented

Figure 11-5. TIM Status and Control Register

TOF — TIM Overflow Flag Bit

This read/write flag is set when the TIM counter reaches the modulo value programmed in the TIM counter modulo registers. Clear TOF by reading the TIM status and control register when TOF is set and then writing a 0 to TOF. If another TIM overflow occurs before the clearing sequence is complete, then writing 0 to TOF has no effect. Therefore, a TOF interrupt request cannot be lost due to inadvertent clearing of TOF. Reset clears the TOF bit. Writing a 1 to TOF has no effect.

1 = TIM counter has reached modulo value.
0 = TIM counter has not reached modulo value.

12.7 State Machine

Figure 12-3 details the main state machine.

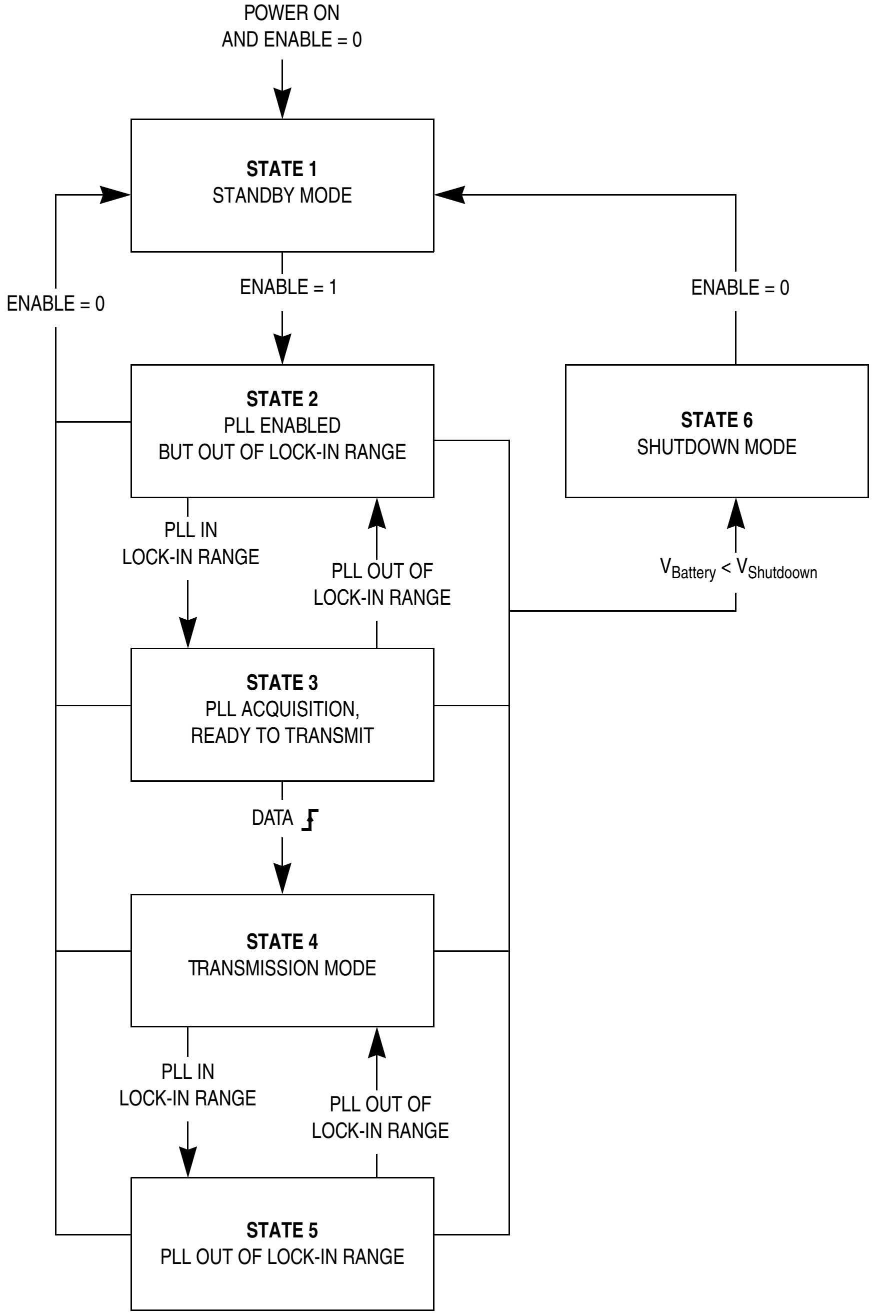


Figure 12-3. Main State Machine

State 1

The circuit is in standby mode and draws only a leakage current from the power supply.

State 2

In this state, the PLL is enabled but out of the lock-in range. Therefore the RF output stage is switched off preventing any data transmission. Data clock is available on pin DATACLK. In normal operation, this state is transitional.

State 3

In this state, the PLL is within the lock-in range.

If $t < t_{PLL\_Lock\_In}$, then the PLL can still be in acquisition mode.
If $t \geq t_{PLL\_Lock\_In}$, then the PLL is locked.

The circuit is ready to transmit in band and is waiting for the first data (see **Figure 12-4**).

State 4

A rising edge on pin DATA starts the transmission. Data entered on pin DATA are output on pin RFOUT. The modulation is the one selected through the level applied on pin MODE.

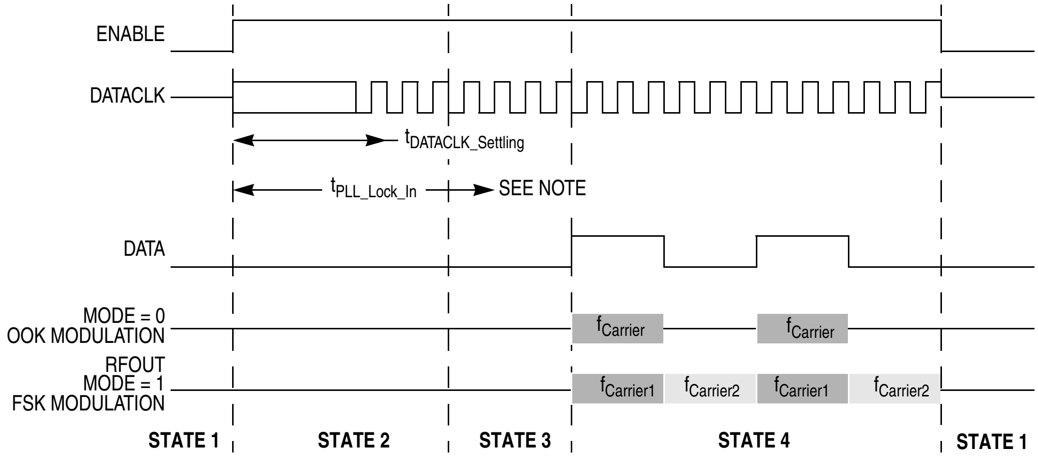
State 5

An out-of-lock condition has been detected. The RF output stage is switched off preventing any data transmission. Data clock is available on pin DATACLK.

State 6

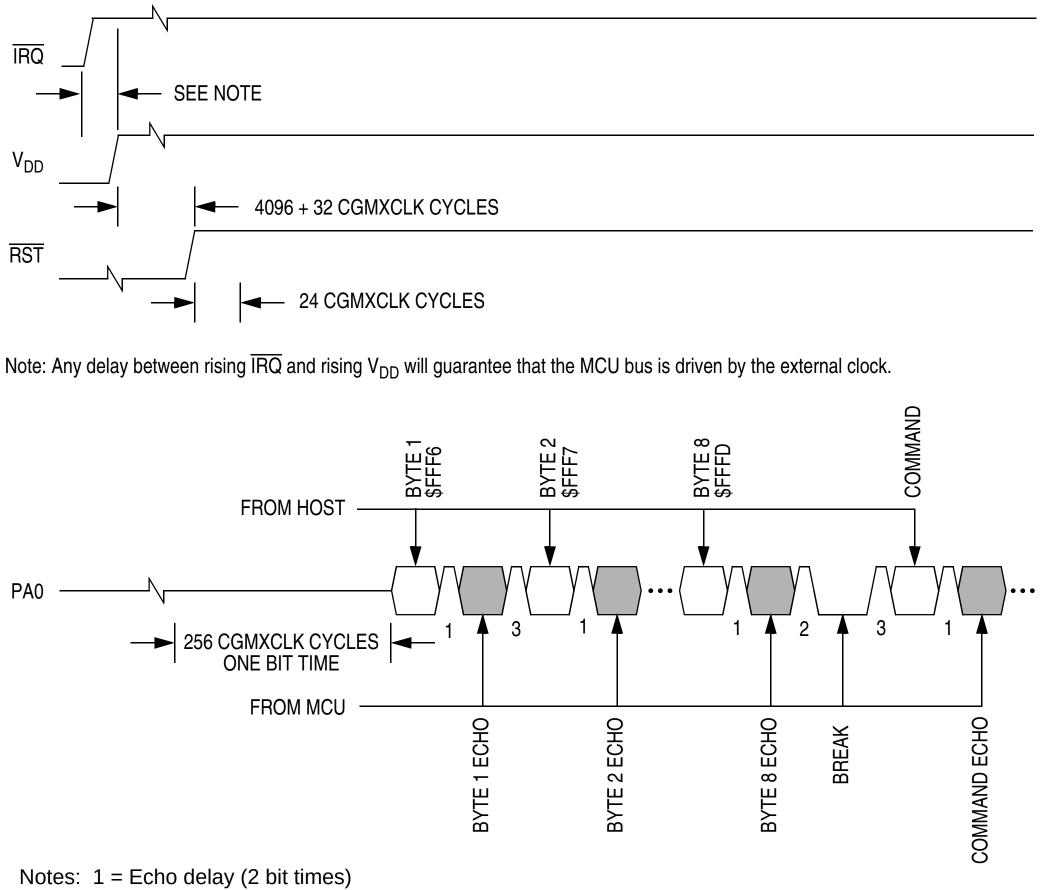
When the supply voltage falls below the shutdown voltage threshold (V_{SDWN}) the whole circuit is switched off. Applying a low level on pin ENABLE is the only condition to get out of this state.

Figure 12-4 shows the waveforms of the main signals for a typical application cycle



Note: PLL locked, circuit ready to tramsmit in band.

Figure 12-4. Signals, Waveforms, and Timing Definitions



Note: Any delay between rising $\overline{\text{IRQ}}$ and rising V_{DD} will guarantee that the MCU bus is driven by the external clock.

Notes: 1 = Echo delay (2 bit times)
2 = Data return delay (2 bit times)
3 = Wait 1 bit time before sending next byte.

Figure 13-13. Monitor Mode Entry Timing

If the received bytes match those at locations \$FFF6–\$FFFD, the host bypasses the security feature and can read all FLASH locations and execute code from FLASH. Security remains bypassed until a power-on reset occurs. After the host bypasses security, any reset other than a power-on reset requires the host to send another eight bytes, but security remains bypassed regardless of the data that the host sends.

If the received bytes do not match the data at locations \$FFF6–\$FFFD, the host fails to bypass the security feature. The MCU remains in monitor mode, but reading FLASH locations returns undefined data, and trying to execute code from FLASH causes an illegal address reset. After the host fails to bypass security, any reset other than a power-on reset causes an endless loop of illegal address resets.

After receiving the eight security bytes from the host, the MCU transmits a break character signalling that it is ready to receive a command.

NOTE: *The MCU does not transmit a break character until after the host sends the eight security bytes.*