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	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	I²C, SCI, SPI
Peripherals	LCD, LVD, POR, PWM, WDT
Number of I/O	24
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 8x12b
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	80-LQFP
Supplier Device Package	80-FQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08lc60lk

Control bits inside the ICG determine which source is connected.

- FFE is a control signal generated inside the ICG. If the frequency of ICGOUT > 4 × the frequency of ICGERCLK, this signal is a logic 1 and the fixed-frequency clock will be the ICGERCLK. Otherwise the fixed-frequency clock will be BUSCLK.
- ICGLCLK — Development tools can select this internal self-clocked source (~ 8 MHz) to speed up BDC communications in systems where the bus clock is slow.
- ICGERCLK — External reference clock can be selected as the real-time interrupt clock source.

After entering active background mode, the CPU is held in a suspended state waiting for serial background commands rather than executing instructions from the user's application program.

Background commands are of two types:

- Non-intrusive commands, defined as commands that can be issued while the user program is running. Non-intrusive commands can be issued through the BKGD pin while the MCU is in run mode; non-intrusive commands can also be executed when the MCU is in the active background mode. Non-intrusive commands include:
  - Memory access commands
  - Memory-access-with-status commands
  - BDC register access commands
  - The BACKGROUND command
- Active background commands, which can only be executed while the MCU is in active background mode. Active background commands include commands to:
  - Read or write CPU registers
  - Trace one user program instruction at a time
  - Leave active background mode to return to the user's application program (GO)

The active background mode is used to program a bootloader or user application program into the FLASH program memory before the MCU is operated in run mode for the first time. When MC9S08LC60 Series MCUs are shipped from the Freescale factory, the FLASH program memory is erased by default unless specifically noted, so there is no program that could be executed in run mode until the FLASH memory is initially programmed. The active background mode can also be used to erase and reprogram the FLASH memory after it has been previously programmed.

For additional information about the active background mode, refer to the Development Support chapter.

3.5 Wait Mode

Wait mode is entered by executing a WAIT instruction. Upon execution of the WAIT instruction, the CPU enters a low-power state in which it is not clocked. The I bit in the condition code register (CCR) is cleared when the CPU enters wait mode, enabling interrupts. When an interrupt request occurs, the CPU exits wait mode and resumes processing, beginning with the stacking operations leading to the interrupt service routine.

While the MCU is in wait mode, there are some restrictions on which background debug commands can be used. Only the BACKGROUND command and memory-access-with-status commands are available while the MCU is in wait mode. The memory-access-with-status commands do not allow memory access, but they report an error indicating that the MCU is in either stop or wait mode. The BACKGROUND command can be used to wake the MCU from wait mode and enter active background mode.

4.2 Register Addresses and Bit Assignments

The registers in the MC9S08LC60 Series are divided into these three groups:

- Direct-page registers are located in the first 128 locations in the memory map, so they are accessible with efficient direct addressing mode instructions.
- High-page registers are used much less often, so they are located above 0x1800 in the memory map. This leaves more room in the direct page for more frequently used registers and variables.
- The nonvolatile register area consists of a block of 16 locations in FLASH memory at 0xFFB0–0xFFBF.

  Nonvolatile register locations include:

  — NVPROT and NVOPT are loaded into working registers at reset

  — An 8-byte backdoor comparison key which optionally allows a user to gain controlled access to secure memory

  Because the nonvolatile register locations are FLASH memory, they must be erased and programmed like other FLASH memory locations.

Direct-page registers can be accessed with efficient direct addressing mode instructions. Bit manipulation instructions can be used to access any bit in any direct-page register. Table 4-2 is a summary of all user-accessible direct-page registers and control bits.

The direct page registers in Table 4-2 can use the more efficient direct addressing mode which only requires the lower byte of the address. Because of this, the lower byte of the address in column one is shown in bold text. In Table 4-3 and Table 4-4 the whole address in column one is shown in bold. In Table 4-2, Table 4-3, and Table 4-4, the register names in column two are shown in bold to set them apart from the bit names to the right. Cells that are not associated with named bits are shaded. A shaded cell with a 0 indicates this unused bit always reads as a 0. Shaded cells with dashes indicate unused or reserved bit locations that could read as 1s or 0s.

5.8.2 System Reset Status Register (SRS)

This register includes six read-only status flags to indicate the source of the most recent reset. When a debug host forces reset by writing 1 to BDFR in the SBDFR register, none of the status bits in SRS will be set. Writing any value to this register address clears the COP watchdog timer without affecting the contents of this register. The reset state of these bits depends on what caused the MCU to reset.

	7	6	5	4	3	2	1	0
R	POR	PIN	COP	ILOP	0	ICG	LVD	0
W	Writing any value to SRS address clears COP watchdog timer.							
POR:	1	0	0	0	0	0	1	0
LVR:	u	0	0	0	0	0	1	0
Any other reset:	0	Note(1)	Note(1)	Note(1)	0	Note(1)	0	0

u = Unaffected by reset

1 Any of these reset sources that are active at the time of reset will cause the corresponding bit(s) to be set; bits corresponding to sources that are not active at the time of reset will be cleared.

Figure 5-3. System Reset Status (SRS)

Table 5-4. SRS Field Descriptions

Field	Description
7 POR	**Power-On Reset** — Reset was caused by the power-on detection logic. Because the internal supply voltage was ramping up at the time, the low-voltage reset (LVD) status bit is also set to indicate that the reset occurred while the internal supply was below the LVD threshold. 0 Reset not caused by POR. 1 POR caused reset.
6 PIN	**External Reset Pin** — Reset was caused by an active-low level on the external reset pin. 0 Reset not caused by external reset pin. 1 Reset came from external reset pin.
5 COP	**Computer Operating Properly (COP) Watchdog** — Reset was caused by the COP watchdog timer timing out. This reset source may be blocked by COPE = 0. 0 Reset not caused by COP timeout. 1 Reset caused by COP timeout.
4 ILOP	**Illegal Opcode** — Reset was caused by an attempt to execute an unimplemented or illegal opcode. The STOP instruction is considered illegal if stop is disabled by STOPE = 0 in the SOPT register. The BGND instruction is considered illegal if active background mode is disabled by ENBDM = 0 in the BDCSC register. 0 Reset not caused by an illegal opcode. 1 Reset caused by an illegal opcode.
2 ICG	**Internal Clock Generation Module Reset** — Reset was caused by an ICG module reset. 0 Reset not caused by ICG module. 1 Reset caused by ICG module.
1 LVD	**Low Voltage Detect** — If the LVD reset is enabled (LVDE = LVDRE = 1) and the supply drops below the LVD trip voltage, an LVD reset occurs. The LVD function is disabled when the MCU enters stop. To maintain LVD operation in stop, the LVDSE bit must be set. 0 Reset not caused by LVD trip or POR. 1 Reset caused by LVD trip or POR.

6.2.3 Port B Registers

This section provides information about all registers and control bits associated with the parallel I/O ports. The parallel I/O registers are located in page zero of the memory map.

Refer to tables in Chapter 4, “Memory” for the absolute address assignments for all parallel I/O registers. This section refers to registers and control bits only by their names. A Freescale-provided equate or header file normally is used to translate these names into the appropriate absolute addresses.

6.2.3.1 Port B Data Registers (PTBD)

Port B parallel I/O function is controlled by the data and data direction registers in this section.

	7	6	5	4	3	2	1	0
R W	PTBD7	PTBD6	PTBD5	PTBD4	PTBD3	PTBD2[1]	PTBD1	PTBD0
Reset	0	0	0	0	0	0	0	0

Figure 6-10. Port B Data Register (PTBD)

1 Reads of PTBD2 always return the contents of PTBD2, regardless of the value stored in the bit PTBDD2

Table 6-6. PTBD Field Descriptions

Field	Description
7:0 PTBD[7:0]	**Port B Data Register Bits** — For port B pins that are inputs, reads return the logic level on the pin. For port B pins that are configured as outputs, reads return the last value written to this register. Writes are latched into all bits of this register. For port B pins that are configured as outputs, the logic level is driven out the corresponding MCU pin. Reset forces PTBD to all 0s, but these 0s are not driven out the corresponding pins because reset also configures all port pins as high-impedance inputs with pullups disabled.

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

7.4.1 Edge Only Sensitivity

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

7.4.2 Edge and Level Sensitivity

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

7.4.3 KBI Pullup/Pulldown Resistors

The KBI pins can be configured to use an internal pullup/pulldown resistor using the associated I/O port pullup enable register. If an internal resistor is enabled, the KBIxES register is used to select whether the resistor is a pullup (KBEDGn = 0) or a pulldown (KBEDGn = 1).

7.4.4 KBI Initialization

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

1. Mask keyboard interrupts by clearing KBIE in KBIxSC.
2. Enable the KBI polarity by setting the appropriate KBEDGn bits in KBIxES.
3. If using internal pullup/pulldown device, configure the associated pullup enable bits in PTxPE.
4. Enable the KBI pins by setting the appropriate KBIPEn bits in KBIxPE.
5. Write to KBACK in KBIxSC to clear any false interrupts.
6. Set KBIE in KBIxSC to enable interrupts.

Chapter 9
Liquid Crystal Display Driver (S08LCDV1)

9.1 Introduction

The LCD driver module is a CMOS charge pump voltage inverter that is designed for low-voltage, low-power operation. The LCD driver module is designed to generate the appropriate waveforms to drive multiplexed numeric, alpha-numeric, or custom LCD panels. Depending on LCD module hardware and software configuration, the LCD panels can be either 3 V or 5 V and be driven by different waveform modes. The LCD module also has several timing and control settings that can be software configured depending on the applications requirements. Timing and control consists of registers and control logic for:

- LCD frame frequency
- Duty cycle select
- Frontplane/backplane select and enable
- Blink modes and frequency
- Operation in low-power modes

In a 64-pin package, the LCD module can be configured to drive 4 backplanes/32 frontplanes (128 segments) or 3 backplanes/33 frontplanes (99 segments). In a 80-pin package, the LCD module can be configured to drive a maximum of 4 backplanes/40 frontplanes (160 segments) or 3 backplanes/41 frontplanes (123 segments). These configurations are summarized in Table 9-1.

Table 9-1. Configuration Options by Package Type

Package Type	No. of Backplanes	No. of Frontplanes	No. of Segments
64-pin	4	32	128
	3	33	99
80-pin	4	40	160
	3	41	123

Figure 9-1 shows the MC9S08LC60 Series block diagram with the LCD highlighted.

9.2 External Signal Description

The LCD module has several external pins dedicated to power supply and also LCD frontplane/backplane signaling. Table 9-3 itemizes all the LCD external pins. See the Pins and Connections chapter for device-specific pin configurations.

Table 9-3. Signal Properties

Name	Port	Function	Reset State
3 backplanes	BP[2:0],	Backplane waveform signals that connect directly to the pads	High impedance
40 frontplanes	FP[39:0]	Frontplane waveform signals that connect directly to the pads	High impedance
Frontplane/backplane	BP3/FP40	Switchable frontplane/backplane signal that connects directly to the pads	High impedance
LCD voltage	V_{LCD}	LCD supply voltage	—
LCD bias voltages	V_{LL1}, V_{LL2}, V_{LL3}	LCD bias voltages	—
LCD charge pump capacitance	V_{cap1}, V_{cap2}	Charge pump capacitor pins	—

9.2.1 BP[2:0]

This output signal vector represents the analog backplane waveforms of the LCD module. These signals are connected to the back plane of the LCD panel.

9.2.2 FP[39:0]

This output signal vector represents the analog frontplane waveforms of the LCD module. These signals are connected to the front plane of the LCD panel.

9.2.3 BP3/FP40

This signal vector represents either an analog frontplane or backplane waveform of the LCD module depending on the configuration of the DUTY[1:0] bit field.

9.2.4 V_{LCD}

V_{LCD} is the positive supply voltage for the LCD module waveform generation. V_{LCD} can internally derive from V_{DD} or externally derive from voltage source range from 0.9 to 1.8 Volts. V_{LCD} is connected to a switch capacitor charge pump DC/DC converter (voltage doubler and voltage tripler) which is able to generate double or triple V_{LCD} in order to support either 3-V or 5-V LCD glass. V_{LCD} is also connected internally to an internal ADC channel in order to monitor the V_{LCD} magnitude.

9.2.5 V_{LL1}, V_{LL2}, V_{LL3}

V_{LL1}, V_{LL2}, and V_{LL3} are bias voltages for the LCD module driver waveforms. Internally generated using the internal charge pump when it is enabled.

9.4.1.4.6 Example 6: 1/4 Duty Multiplexed with 1/3 Bias Mode (Low-power Waveform)

Duty=1/4: DUTY[1:0] = 11 (All available backplanes used)

Low-power Waveform selected: LPWAVE = 1

LCDRAM = 1001

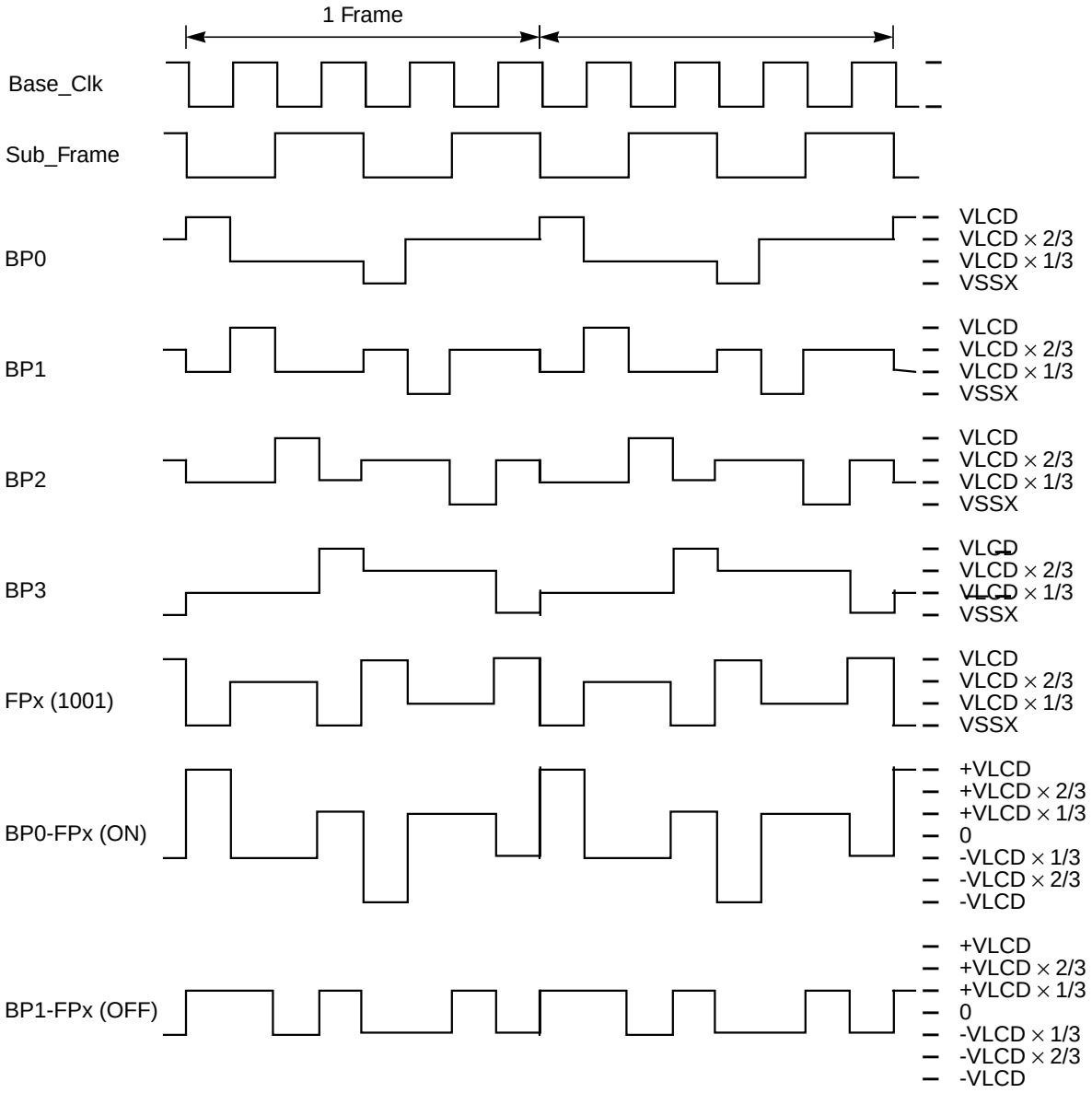


Figure 9-16. 1/4 Duty and 1/3 Bias (Low-power Waveform)

9.4.3.1 LCD Segment Blinking

To configure all LCD segments to blink regardless of the contents of the LCDRAM bits while LCDDRMS = 1, the BLKMODE bit in the LCDBCTL control register must to set to 1. To configure individual LCD segments to blink, the BLKMODE bit in the LCDBCTL control register must be deasserted.

If BLKMODE = 0, asserting the LCDRAM FP[n]BP[x] bits while LCDDRMS = 0 and LCDDRMS = 1 enables the LCD segment connected between FP[n] and BP[x] to blink when BLINK = 1. Each LCDRAM register controls two frontplane drivers.

9.4.3.2 Blink Frequency

The LCD module waveform base clock is the basis for the calculation of the LCD module blink frequency. The LCD module blink frequency is equal to the LCD module waveform base clock divided by the BRATE[2:0] divider. Table 9-17 shows LCD module blink frequency calculations for all values of BRATE[2:0] and LCLK[2:0].

Table 9-17. Blink Frequency Calculations
(Blink Rate = LCD Base (Hz) ÷ Blink Divider)

LCLK[2:0]	LCD Base Frequency (Hz)	Blink Frequency							
0	2049.3	64.0	32.0	16.0	8.00	4.00	2.00	1.00	0.50
1	1024.7	32.0	16.0	8.00	4.00	2.00	1.00	0.50	0.25
2	512.3	16.0	8.00	4.00	2.00	1.00	0.50	0.25	0.13
3	256.2	8.00	4.00	2.00	1.00	0.50	0.25	0.13	0.06
4	128.1	4.00	2.00	1.00	0.50	0.25	0.13	0.06	0.03
5	64.0	2.00	1.00	0.50	0.25	0.13	0.06	0.03	0.02
6	32.0	1.00	0.50	0.25	0.13	0.06	0.03	0.02	0.01
7	16.0	0.50	0.25	0.13	0.06	0.03	0.02	0.01	0.00
		Blink Divider = $2^{(5+ BRATE[2:0])}$							
		32	**64**	**128**	**256**	**512**	**1024**	**2048**	**4096**

1 Shaded table entries are out of specification and are not valid

9.4.4 LCD Charge Pump, Voltage Divider, and Power Supply Operation

This section describes the LCD charge pump, voltage divider, and LCD power supply configuration options. Figure 9-17 provides a block diagram for the LCD charge pump and a V_{LCD} voltage divider.

The VSUPPLY[1:0] bit field in the LCDSUPPLY register is used to configure the LCD module power supply source. VSUPPLY[1:0] indicates the state of internal signals used to configure power switches as shown in the Table in Figure 9-17. The block diagram in Figure 9-17 illustrates several potential operational modes for the LCD module including configuration of the LCD module power supply source using internal V_{DD} or an external supply.

9.5.2.4 Initialization Example 4

Example 3 LCD setup requirements are reiterated in the table below:

Example	Operating Voltage, V_{DD}	LCD Clock Source	LCD Glass Operating Voltage	Required LCD segments	LCD Frame Rate	Blinking Mode/Rate	Behavior in STOP3 and WAIT modes	LCD Power Input
4	1.8-V	External 32.768 kHz	5-V	123	30 Hz	all segment 2.0 Hz	WAIT: off STOP3: off	Power via V_{LCD}

Table 9-25 lists the required setup values required to initialize the LCD as specified by Example 4:

Table 9-25. Initialization Register Values for Example 4

Register	Bit/bit field	Binary Value	Comment
LCDCLKS 00000000	SOURCE	0	Selects the external clock reference as the LCD clock input External clock reference = 0; Bus clock = 1
	DIV16	0	Adjusts the LCD clock input (see table 9-12)
	CLKADJ[5:0]	000000	Adjusts the LCD clock input (see table 9-12)
LCDSUPPLY 1100XX10	LCDCPEN	1	Enable the charge pump
	LCDCPMS	1	For 3-V LCD glass, select tripler mode Doubler mode = 0; Tripler mode = 1
	HDRVBUF	X	High drive buffer
	CPCADJ[1:0]	00	Configure LCD charge pump clock source
	BBYPASS	X	Buffer Bypass; Buffer mode = 0; Unbuffered mode = 1
	VSUPPLY[1:0]	10	When VSUPPLY[1:0] = 10, the LCD must be externally powered via V_{LCD} (see table 9-16). For 5-V glass, the nominal value of V_{LCD} should be 1.67-V.
LCDCR1 XXXXXX00	LCDWAI	0	LCD is “off” in WAIT mode
	LCDSTP3	0	LCD is “off” in STOP3 mode
LCDCR0 0X010X11	LCLK[2:0]	010	For 1/3 duty cycle, select the closest value to the desired 30 Hz LCD frame frequency (see table 9-13). Note the LCD base frequency - 128.1 Hz
	LPWAVE	X	Low power waveform
	DUTY[1:0]	10	For 123 segments (3x41), select 1/3 duty cycle (see table 9-11)
LCDBCTL 0XXX1001	BLKMODE	1	Blink all segments; Blink Segments = 0; Blink All = 1
	BRATE[2:0]	001	Using the LCD base frequency for the selected LCD frame frequency, select 2.0 Hz blink frequency (see table 9-15).
FPENR[5:0]	FPENR0 FPENR1 FPENR2 FPENR3 FPENR4 FPENR5	11111111 11111111 11111111 11111111 11111111 XXXXXXX1	All 41 LCD frontplanes need to be enabled.

entering off mode. If CLKS bits are set to 01 or 11 coming out of the Off state, the ICG enters this mode until ICGDCLK is stable as determined by the DCOS bit. After ICGDCLK is considered stable, the ICG automatically closes the loop by switching to FLL engaged (internal or external) as selected by the CLKS bits.

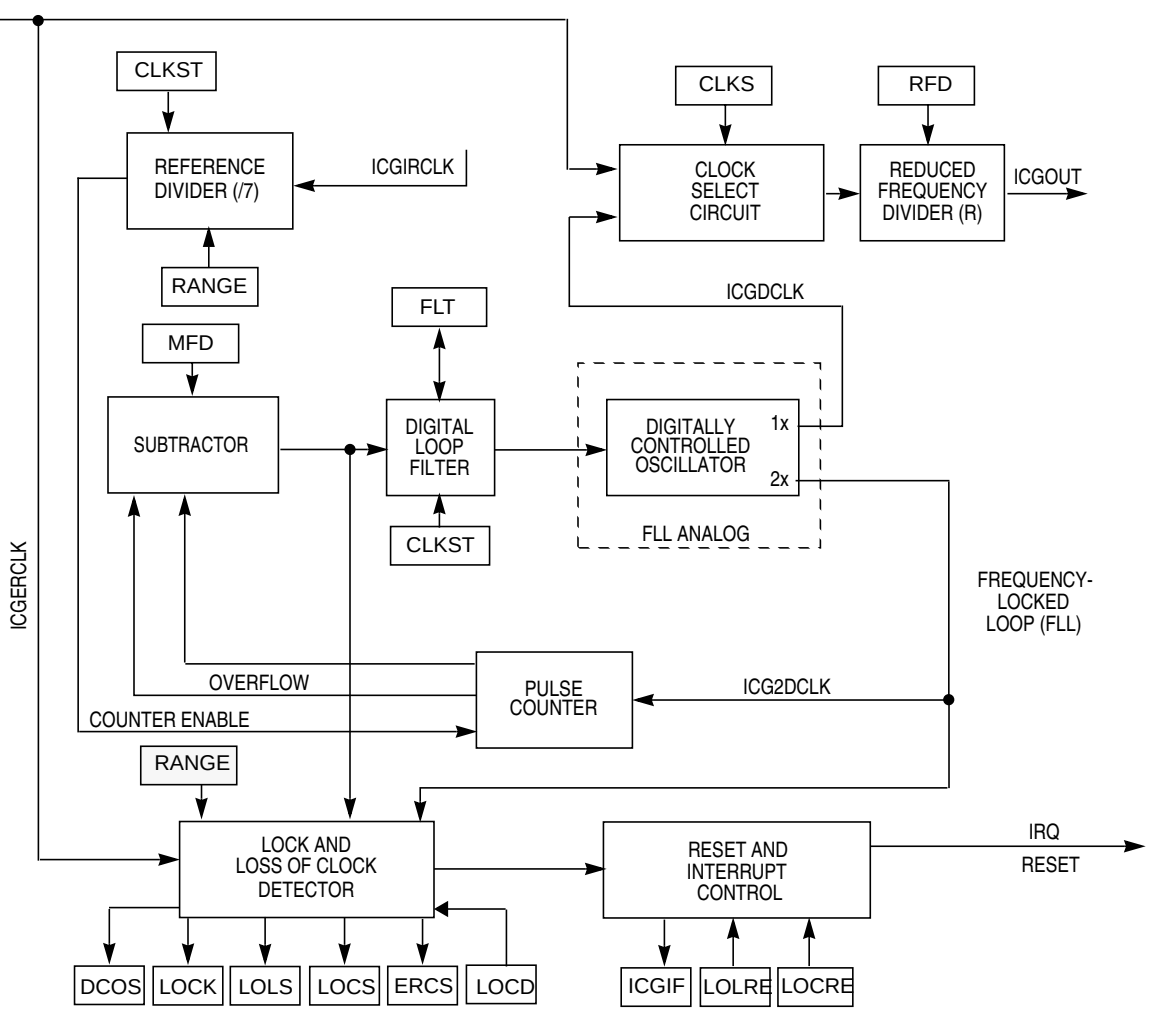


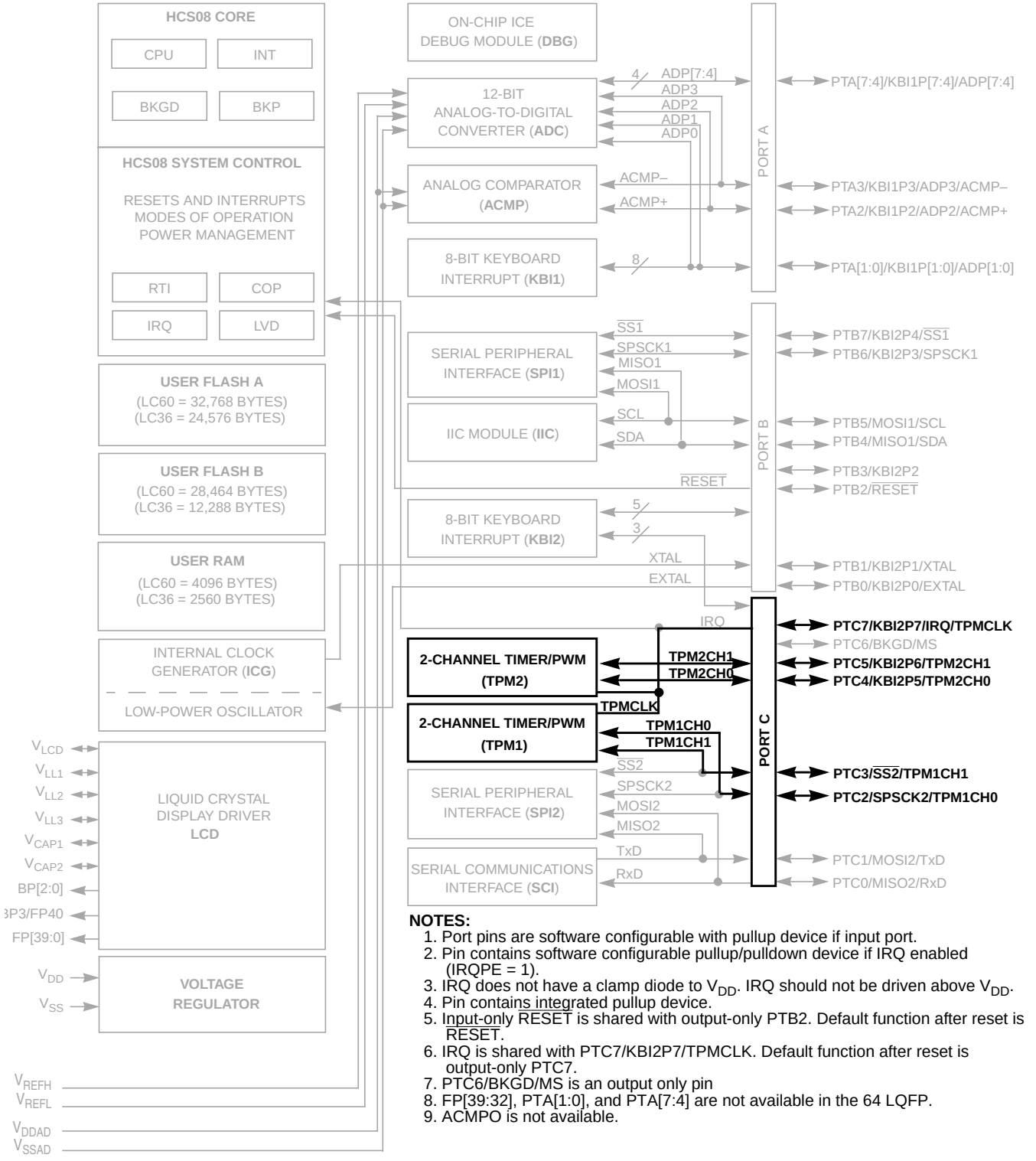
Figure 10-13. Detailed Frequency-Locked Loop Block Diagram

10.5.3 FLL Engaged, Internal Clock (FEI) Mode

FLL engaged internal (FEI) is entered when any of the following conditions occur:

- CLKS bits are written to 01
- The DCO clock stabilizes (DCOS = 1) while in SCM upon exiting the off state with CLKS = 01

In FLL engaged internal mode, the reference clock is derived from the internal reference clock ICGIRCLK, and the FLL loop will attempt to lock the ICGDCLK frequency to the desired value, as selected by the MFD bits.



NOTES:

1. Port pins are software configurable with pullup device if input port.
2. Pin contains software configurable pullup/pulldown device if IRQ enabled (IRQPE = 1).
3. IRQ does not have a clamp diode to V_{DD}. IRQ should not be driven above V_{DD}.
4. Pin contains integrated pullup device.
5. Input-only RESET is shared with output-only PTB2. Default function after reset is RESET.
6. IRQ is shared with PTC7/KBI2P7/TPMCLK. Default function after reset is output-only PTC7.
7. PTC6/BKGD/MS is an output only pin
8. FP[39:32], PTA[1:0], and PTA[7:4] are not available in the 64 LQFP.
9. ACMPO is not available.

Figure 11-1. MC9S08LC60 Series Block Diagram Highlighting TPM Block and Pins

13.3 Modes of Operation

13.3.1 SPI in Stop Modes

The SPI is disabled in all stop modes, regardless of the settings before executing the STOP instruction. During either stop1 or stop2 mode, the SPI module will be fully powered down. Upon wake-up from stop1 or stop2 mode, the SPI module will be in the reset state. During stop3 mode, clocks to the SPI module are halted. No registers are affected. If stop3 is exited with a reset, the SPI will be put into its reset state. If stop3 is exited with an interrupt, the SPI continues from the state it was in when stop3 was entered.

13.4 Register Definition

The SPI has five 8-bit registers to select SPI options, control baud rate, report SPI status, and for transmit/receive data.

Refer to the direct-page register summary in the Memory chapter of this data sheet for the absolute address assignments for all SPI registers. This section refers to registers and control bits only by their names, and a Freescale-provided equate or header file is used to translate these names into the appropriate absolute addresses.

13.4.1 SPI Control Register 1 (SPIxC1)

This read/write register includes the SPI enable control, interrupt enables, and configuration options.

	7	6	5	4	3	2	1	0
R W	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
Reset	0	0	0	0	0	1	0	0

Figure 13-5. SPI Control Register 1 (SPIxC1)

Table 13-1. SPIxC1 Field Descriptions

Field	Description
7 SPIE	**SPI Interrupt Enable (for SPRF and MODF)** — This is the interrupt enable for SPI receive buffer full (SPRF) and mode fault (MODF) events. 0 Interrupts from SPRF and MODF inhibited (use polling) 1 When SPRF or MODF is 1, request a hardware interrupt
6 SPE	**SPI System Enable** — Disabling the SPI halts any transfer that is in progress, clears data buffers, and initializes internal state machines. SPRF is cleared and SPTEF is set to indicate the SPI transmit data buffer is empty. 0 SPI system inactive 1 SPI system enabled
5 SPTIE	**SPI Transmit Interrupt Enable** — This is the interrupt enable bit for SPI transmit buffer empty (SPTEF). 0 Interrupts from SPTEF inhibited (use polling) 1 When SPTEF is 1, hardware interrupt requested

Table 13-7. SPIxS Register Field Descriptions

Field	Description
7 SPRF	**SPI Read Buffer Full Flag** — SPRF is set at the completion of an SPI transfer to indicate that received data may be read from the SPI data register (SPIxD). SPRF is cleared by reading SPRF while it is set, then reading the SPI data register. 0 No data available in the receive data buffer 1 Data available in the receive data buffer
5 SPTEF	**SPI Transmit Buffer Empty Flag** — This bit is set when there is room in the transmit data buffer. It is cleared by reading SPIxS with SPTEF set, followed by writing a data value to the transmit buffer at SPIxD. SPIxS must be read with SPTEF = 1 before writing data to SPIxD or the SPIxD write will be ignored. SPTEF generates an SPTEF CPU interrupt request if the SPTIE bit in the SPIxC1 is also set. SPTEF is automatically set when a data byte transfers from the transmit buffer into the transmit shift register. For an idle SPI (no data in the transmit buffer or the shift register and no transfer in progress), data written to SPIxD is transferred to the shifter almost immediately so SPTEF is set within two bus cycles allowing a second 8-bit data value to be queued into the transmit buffer. After completion of the transfer of the value in the shift register, the queued value from the transmit buffer will automatically move to the shifter and SPTEF will be set to indicate there is room for new data in the transmit buffer. If no new data is waiting in the transmit buffer, SPTEF simply remains set and no data moves from the buffer to the shifter. 0 SPI transmit buffer not empty 1 SPI transmit buffer empty
4 MODF	**Master Mode Fault Flag** — MODF is set if the SPI is configured as a master and the slave select input goes low, indicating some other SPI device is also configured as a master. The $\overline{SS}$ pin acts as a mode fault error input only when MSTR = 1, MODFEN = 1, and SSOE = 0; otherwise, MODF will never be set. MODF is cleared by reading MODF while it is 1, then writing to SPI control register 1 (SPIxC1). 0 No mode fault error 1 Mode fault error detected

13.4.5 SPI Data Register (SPIxD)

	7	6	5	4	3	2	1	0
R W	Bit 7	6	5	4	3	2	1	Bit 0
Reset	0	0	0	0	0	0	0	0

Figure 13-9. SPI Data Register (SPIxD)

Reads of this register return the data read from the receive data buffer. Writes to this register write data to the transmit data buffer. When the SPI is configured as a master, writing data to the transmit data buffer initiates an SPI transfer.

Data should not be written to the transmit data buffer unless the SPI transmit buffer empty flag (SPTEF) is set, indicating there is room in the transmit buffer to queue a new transmit byte.

Data may be read from SPIxD any time after SPRF is set and before another transfer is finished. Failure to read the data out of the receive data buffer before a new transfer ends causes a receive overrun condition and the data from the new transfer is lost.

15.4.4.2 Completing Conversions

A conversion is completed when the result of the conversion is transferred into the data result registers, ADCRH and ADCRL. This is indicated by the setting of COCO. An interrupt is generated if AIEN is high at the time that COCO is set.

A blocking mechanism prevents a new result from overwriting previous data in ADCRH and ADCRL if the previous data is in the process of being read while in 12-bit or 10-bit MODE (the ADCRH register has been read but the ADCRL register has not). When blocking is active, the data transfer is blocked, COCO is not set, and the new result is lost. In the case of single conversions with the compare function enabled and the compare condition false, blocking has no effect and ADC operation is terminated. In all other cases of operation, when a data transfer is blocked, another conversion is initiated regardless of the state of ADCO (single or continuous conversions enabled).

If single conversions are enabled, the blocking mechanism could result in several discarded conversions and excess power consumption. To avoid this issue, the data registers must not be read after initiating a single conversion until the conversion completes.

15.4.4.3 Aborting Conversions

Any conversion in progress will be aborted when:

- A write to ADCSC1 occurs (the current conversion will be aborted and a new conversion will be initiated, if ADCH are not all 1s).
- A write to ADCSC2, ADCCFG, ADCCVH, or ADCCVL occurs. This indicates a mode of operation change has occurred and the current conversion is therefore invalid.
- The MCU is reset.
- The MCU enters stop mode with ADACK not enabled.

When a conversion is aborted, the contents of the data registers, ADCRH and ADCRL, are not altered but continue to be the values transferred after the completion of the last successful conversion. In the case that the conversion was aborted by a reset, ADCRH and ADCRL return to their reset states.

15.4.4.4 Power Control

The ADC module remains in its idle state until a conversion is initiated. If ADACK is selected as the conversion clock source, the ADACK clock generator is also enabled.

Power consumption when active can be reduced by setting ADLPC. This results in a lower maximum value for f_{ADCK} (see the electrical specifications).

15.4.4.5 Sample Time and Total Conversion Time

The total conversion time depends on the sample time (as determined by ADLSMP), the MCU bus frequency, the conversion mode (8-bit, 10-bit or 12-bit), and the frequency of the conversion clock (f_{ADCK}). After the module becomes active, sampling of the input begins. ADLSMP is used to select between short (3.5 ADCK cycles) and long (23.5 ADCK cycles) sample times.When sampling is complete, the converter is isolated from the input channel and a successive approximation algorithm is performed to determine the

Table A-5. DC Characteristics (Sheet 2 of 2)
(Temperature Range = -40 to 85°C Ambient)

Parameter	Symbol	Min	Typical[1]	Max	Unit
Output high voltage ($V_{DD} \geq 1.8$ V) I_{OH} = –2 mA (ports A, B, and C)	V_{OH}	V_{DD} – 0.5		—	V
Output high voltage (all port pins) I_{OH} = –10 mA ($V_{DD} \geq 2.7$ V) I_{OH} = –6 mA ($V_{DD} \geq 2.3$ V) I_{OH} = –3 mA ($V_{DD} \geq 1.8$ V)		V_{DD} – 0.5		— — —	
Maximum total I_{OH} for all port pins	$\lvert I_{OHT} \rvert$	—		60	mA
Output low voltage ($V_{DD} \geq 1.8$ V) I_{OL} = 2.0 mA (ports A, B, and C)	V_{OL}	—		0.5	V
Output low voltage (all port pins) I_{OL} = 10.0 mA ($V_{DD} \geq 2.7$ V) I_{OL} = 6 mA ($V_{DD} \geq 2.3$ V) I_{OL} = 3 mA ($V_{DD} \geq 1.8$ V)		— — —		0.5 0.5 0.5	
Maximum total I_{OL} for all port pins	I_{OLT}	—		60	mA
dc injection current[4], [5], [6], [7], [8] $V_{IN} < V_{SS}$, $V_{IN} > V_{DD}$ Single pin limit Total MCU limit, includes sum of all stressed pins	$\lvert I_{IC} \rvert$	— —		0.2 5	mA mA
Input capacitance (all non-supply pins)[2]	C_{In}	—		7	pF

1 Typicals are measured at 25°C.

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

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

4 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.

5 All functional non-supply pins are internally clamped to V_{SS} and V_{DD}.

6 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.

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

8 IRQ does not have a clamp diode to V_{DD}. Do not drive IRQ above V_{DD}.

Table A-8. 3 Volt 12-bit ADC Characteristics (V_{REFH} = V_{DDAD}, V_{REFL} = V_{SSAD}) (continued)

Characteristic	Conditions	C	Symbol	Min	Typ[1]	Max	Unit	Comment
Zero-Scale Error (64-pin package only)	12 bit mode	T	E_{ZS}				LSB^2	V_{ADIN} = V_{SSAD}
	10 bit mode	P		—	±1.5	±2.1		
	8 bit mode	P		—	±0.5	±0.7		
Full-Scale Error (80-pin package only)	12 bit mode	T	E_{FS}	—	±1.0	—	LSB^2	V_{ADIN} = V_{DDAD}
	10 bit mode	P		—	±0.5	±1		
	8 bit mode	P		—	±0.5	±0.5		
Full-Scale Error (64-pin package only)	12 bit mode	T	E_{FS}				LSB^2	V_{ADIN} = V_{DDAD}
	10 bit mode	P		—	±1	±1.5		
	8 bit mode	P		—	±0.5	±0.5		
Quantization Error	12 bit mode	D	E_Q	—	±0.5	—	LSB^2	
	10 bit mode			—	—	±0.5		
	8 bit mode			—	—	±0.5		
Input Leakage Error	12 bit mode	D	E_{IL}	—	±2	—	LSB^2	Pad leakage[4] * R_{AS}
	10 bit mode			—	±0.2	±4		
	8 bit mode			—	±0.1	±1.2		
Temp Sensor Slope	-40°C– 25°C	D	m	—	1.646	—	mV/°C	
	25°C– 125°C			—	1.769	—		
Temp Sensor Voltage	25°C	D	V_{TEMP25}	—	701.2	—	mV	

1 Typical values assume V_{DDAD} = 3.0V, Temp = 25°C, f_{ADCK}=1.0MHz unless otherwise stated. Typical values are for reference only and are not tested in production.

2 1 LSB = $(V_{REFH} - V_{REFL})/2^N$

3 Monotonicity and No-Missing-Codes guaranteed in 10 bit and 8 bit modes

4 Based on input pad leakage current. Refer to pad electricals.

A.8 LCD Characteristics

Table A-9. LCD Electricals, 3-V Glass

Characteristic	Symbol	Min	Typ	Max	Unit
LCD Supply Voltage	V_{LCD}	0.9	-	1.8	V
LCD Frame Frequency	f_{Rame}	25	30	100	Hz

Note: Current consumption data based on using the external 32-kHz oscillator with LCD configured using the low-power wave forms option, a 1/4 duty, and a 32-Hz frame frequency. C_{LCD} = C_{BYLCD} = 100 nF; 160 segment 2000 pF LCD panel.

Table A-12. ICG Frequency Specifications
(V_{DDA} = V_{DDA} (min) to V_{DDA} (max), Temperature Range = 0 to 70°C Ambient)

Characteristic	Symbol	Min	Typical	Max	Unit
Duty cycle of input clock [4] (REFS = 0)	t_{dc}	40	—	60	%
Output clock ICGOUT frequency CLKS = 10, REFS = 0 All other cases	f_{ICGOUT}	f_{Extal} (min) f_{lo} (min)		f_{Extal} (max) $f_{ICGDCLKmax}$ (max)	MHz
Minimum DCO clock (ICGDCLK) frequency	$f_{ICGDCLKmin}$	8	—		MHz
Maximum DCO clock (ICGDCLK) frequency	$f_{ICGDCLKmax}$		—	40	MHz
Self-clock mode (ICGOUT) frequency [1]	f_{Self}	$f_{ICGDCLKmin}$		$f_{ICGDCLKmax}$	MHz
Self-clock mode reset (ICGOUT) frequency	f_{Self_reset}	5.5	8	10.5	MHz
Loss of reference frequency [2] Low range High range	f_{LOR}	5 50		25 500	kHz
Loss of DCO frequency [3]	f_{LOD}	0.5		1.5	MHz
Crystal start-up time [4], [5] Low range High range	t_{CSTL} t_{CSTH}	— —	430 4	— —	ms
FLL lock time [4, 6] Low range High range	t_{Lockl} t_{Lockh}	— —		5 5	ms
FLL frequency unlock range	n_{Unlock}	–4*N		4*N	counts
FLL frequency lock range	n_{Lock}	–2*N		2*N	counts
ICGOUT period jitter, [4, 7] measured at f_{ICGOUT} Max Long term jitter (averaged over 2 ms interval)	C_{Jitter}	—		0.2	% f_{ICG}
Internal oscillator deviation from trimmed frequency[8] V_{DD} = 1.8 – 3.6 V, (constant temperature) V_{DD} = 3.0 V ±10%, –40° C to 85° C	ACC_{int}	— —	±0.5 ±0.5	±2 ±2	%

1 Self-clocked mode frequency is the frequency that the DCO generates when the FLL is open-loop.

2 Loss of reference frequency is the reference frequency detected internally, which transitions the ICG into self-clocked mode if it is not in the desired range.

3 Loss of DCO frequency is the DCO frequency detected internally, which transitions the ICG into FLL bypassed external mode (if an external reference exists) if it is not in the desired range.

4 This parameter is characterized before qualification rather than 100% tested.

5 Proper PC board layout procedures must be followed to achieve specifications.

6 This specification applies to the period of time required for the FLL to lock after entering FLL engaged internal or external modes. If a crystal/resonator is being used as the reference, this specification assumes it is already running.

7 Jitter is the average deviation from the programmed frequency measured over the specified interval at maximum f_{ICGOUT}. Measurements are made with the device powered by filtered supplies and clocked by a stable external clock signal. Noise injected into the FLL circuitry via V_{DDA} and V_{SSA} and variation in crystal oscillator frequency increase the C_{Jitter} percentage for a given interval.

8 See Figure A-10