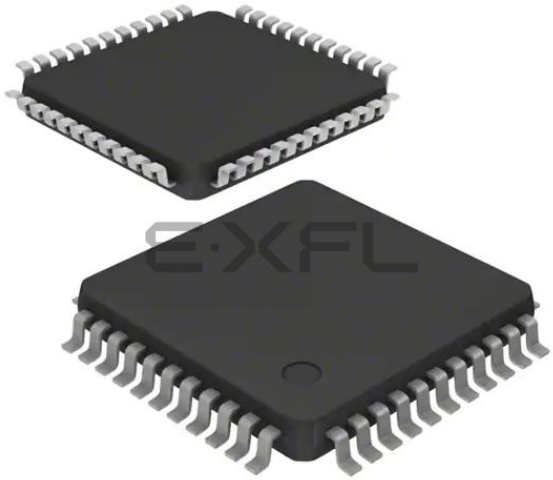


Welcome to E-XFL.COM

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	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LED, LVD, POR, PWM, Temp Sensor, WDT
Number of I/O	37
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	44-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f0880an020sg

Table 4. Signal Descriptions (Continued)

Signal Mnemonic	I/O	Description
Oscillators		
XIN	I	External Crystal Input: The input pin to the crystal oscillator. A crystal can be connected between the pin and the XOUT pin to form the oscillator. In addition, this pin is used with external RC networks or external clock drivers to provide the system clock.
XOUT	O	External Crystal Output: This pin is the output of the crystal oscillator. A crystal can be connected between it and the XIN pin to form the oscillator.
X2IN	I	Watch Crystal Input: The input pin to the low-power 32 kHz oscillator. A watch crystal can be connected between the X2IN and the X2OUT pin to form the oscillator.
X2OUT	O	Watch Crystal Output: This pin is the output from the low power 32 kHz oscillator. A watch crystal can be connected between the X2IN and the X2OUT pin to form the oscillator.
Clock Input		
CLKIN	I	Clock Input Signal: This pin can be used to input a TTL-level signal to be used as the system clock.
LED Drivers		
LED	O	Direct LED Drive Capability: All Port C pins have the capability to drive an LED without any other external components. These pins have programmable drive strengths set by the GPIO block.
On-Chip Debugger		
DBG	I/O	Debug: This signal is the control and data input and output of the On-Chip Debugger. **Caution:** The DBG pin is open-drain and requires an external pull-up resistor to ensure proper operation.
Reset		
RESET	I/O	RESET: Generates a Reset when asserted (driven Low). Also serves as a Reset indicator; the Z8 Encore! XP forces this pin Low when in Reset. This pin is open-drain and features an enabled internal pull-up resistor.
Power Supply		
V_{DD}	I	Digital Power Supply.
AV_{DD}	I	Analog Power Supply.
V_{SS}	I	Digital Ground.
AV_{SS}	I	Analog Ground.

Note: The AV_{DD} and AV_{SS} signals are available only in 28-pin, 40-pin and 44-pin packages.

Table 6. F1680 Series MCU Program Memory Maps (Continued)

Program Memory Address (Hex)	Function
Z8F1680 Device	
0000–0001	Flash option bits
0002–0003	Reset vector
0004–0005	WDT interrupt vector
0006–0007	Illegal instruction trap
0008–0037	Interrupt vectors*
0038–003D	Oscillator fail traps*
003E–3FFF	Program Flash
E000–E3FF	1KB PRAM
Z8F0880 Device	
0000–0001	Flash option bits
0002–0003	Reset vector
0004–0005	WDT interrupt vector
0006–0007	Illegal instruction trap
0008–0037	Interrupt vectors*
0038–003D	Oscillator fail traps*
003E–1FFF	Program Flash
E000–E3FF	1KB PRAM

Note: *See Table 36 on page 69 for a list of interrupt vectors and traps.

3.3. Data Memory

The F1680 Series MCU does not use the eZ8 CPU's 64KB Data Memory address space.

3.4. Flash Information Area

Table 7 describes the F1680 Series MCU Flash Information Area. This 512-byte Information Area is accessed by setting bit 7 of the Flash Page Select Register to 1. When access is enabled, the Flash Information Area is mapped into the Program Memory and overlays the 512bytes at addresses FE00H to FFFFH. When the Information Area access is enabled, all reads from these Program Memory addresses return the Information Area data rather than the Program Memory data. Access to the Flash Information Area is read-only.

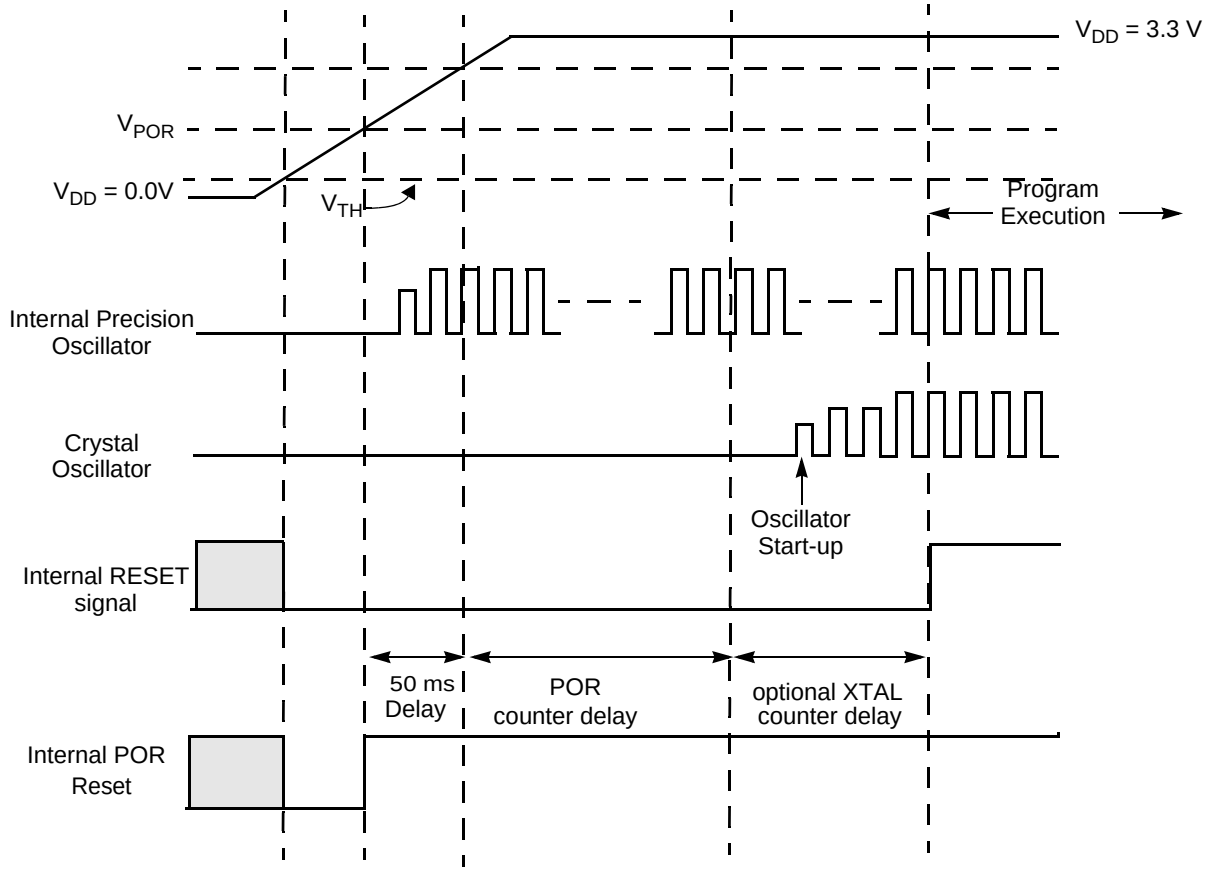
5.2.1. Power-On Reset

Each device in the Z8 Encore! XP F1680 Series contains an internal Power-On Reset (POR) circuit. The POR circuit monitors the supply voltage and holds the whole device in the Reset state until the supply voltage reaches a safe circuit operating level when the device is powered on.

After power on, the POR circuit keeps idle until the supply voltage drops below V_{TH} voltage. Figure 7 on page 35 displays this POR timing.

After the F1680 Series MCU exits the POR state, the eZ8 CPU fetches the Reset vector. Following this POR, the POR/VBO status bit in the Reset Status Register is set to 1.

For the POR threshold voltage (V_{POR}) and POR start voltage V_{TH} , see the Electrical Characteristics chapter on page 349.



Notes
1. Not to Scale.
2. Internal Reset and POR Reset are active Low.

Figure 6. Power-On Reset Operation

7.11.12. Port A–E Output Data Register

The Port A–E Output Data Register, shown in Table 32, controls the output data to the pins.

Table 32. Port A–E Output Data Register (P*x*OUT)

Bits	7	6	5	4	3	2	1	0
Field	POUT7	POUT6	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FD3H, FD7H, FDBH, FDFH, FE3H							

Bit	Description
[7:0] POUT	**Port Output Data** These bits contain the data to be driven to the port pins. The values are only driven if the corresponding pin is configured as an output and the pin is not configured for Alternate Function operation. 0 = Drive a logical 0 (Low). 1 = Drive a logical 1 (High). High value is not driven if the drain has been disabled by setting the corresponding Port Output Control Register bit to 1.

7.11.13. LED Drive Enable Register

The LED Drive Enable Register, shown in Table 33, activates the controlled current drive. The Port C pin must first be enabled by setting the Alternate Function Register to select the LED function.

Table 33. LED Drive Enable (LEDEN)

Bits	7	6	5	4	3	2	1	0
Field	LEDEN[7:0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F82H							

Bit	Description
[7:0] LEDEN	**LED Drive Enable** These bits determine which Port C pins are connected to an internal current sink. 0 = Tristate the Port C pin. 1 = Connect controlled current synch to Port C pin.

Table 44. IRQ1 Enable High Bit Register (IRQ1ENH)

Bits	7	6	5	4	3	2	1	0
Field	PA7VENH	PA6C0ENH	PA5C1ENH	PAD4ENH	PAD3ENH	PAD2ENH	PAD1ENH	PA0ENH
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FC4H							

Bit	Description
[7] PA7VENH	Port A Bit[7] or LVD Interrupt Request Enable High Bit.
[6] PA6C0ENH	Port A Bit[6] or Comparator 0 Interrupt Request Enable High Bit.
[5] PA5C1ENH	Port A Bit[5] or Comparator 1 Interrupt Request Enable High Bit.
[4:1] PAD*x*ENH	Port A or Port D Bit[*x*] (*x*=1, 2, 3, 4) Interrupt Request Enable High Bit.
[0] PA0ENH	Port A Bit[0] Interrupt Request Enable High Bit. See the Shared Interrupt Select Register (IRQSS) on page 82 to determine a selection of either Port A or Port D as the interrupt source.

$$\text{COMPARE Mode Time (s)} = \frac{(\text{Compare Value - Start Value}) \times \text{Prescale}}{\text{Timer Clock Frequency (Hz)}}$$

generated the interrupt, read the associated GPIO input value and compare to the value stored in the TPOL bit.

The timer counts up to the 16-bit reload value stored in the Timer Reload High and Low Byte registers. The timer input is the timer clock. When reaching the reload value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes (assuming the Timer Input signal is still asserted). Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state (from Low to High or from High to Low) at timer reset.

Observe the following steps to configure a timer for GATED Mode and initiate the count:

1. Write to the Timer Control 1 Register to:
   - Disable the timer
   - Configure the timer for GATED Mode
   - Set the prescale value
2. Write to the Timer Control 2 Register to choose the timer clock source.
3. Write to the Timer Control 0 Register to set the timer interrupt configuration field TICONFIG.
4. Write to the Timer High and Low Byte registers to set the starting count value. This value only affects the first pass in GATED Mode. After the first timer reset in GATED Mode, counting always begins at the reset value of 0001H.
5. Write to the Timer Reload High and Low Byte registers to set the reload value.
6. If required, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers. By default, the timer interrupt will be generated for both input deassertion and reload events. If required, configure the timer interrupt to be generated only at the Input Deassertion event or the Reload event by setting TICONFIG field of the Timer Control 0 Register.
7. Configure the associated GPIO port pin for the Timer Input alternate function.
8. Write to the Timer Control 1 Register to enable the timer.
9. Assert the Timer Input signal to initiate the counting.

- Configure the timer for DEMODULATION Mode. Setting the mode also involves writing to the TMODEHI bit in the TxCTL0 Register
- Set the prescale value
- Set the TPOL bit to set the Capture edge (rising or falling) for the Timer Input. This setting applies only if the TPOLHI bit in the TxCTL2 Register is not set

2. Write to the Timer Control 2 Register to:
   - Choose the timer clock source
   - Set the TPOLHI bit if the Capture is required on both edges of the input signal
3. Write to the Timer Control 0 Register to set the timer interrupt configuration field TICONFIG.
4. Write to the Timer High and Low Byte registers to set the starting count value (typically `0001H`).
5. Write to the Timer Reload High and Low Byte registers to set the reload value.
6. Clear the Timer TxPWM0 and TxPWM1 High and Low Byte registers to `0000H`.
7. If required, enable the noise filter and set the noise filter control by writing to the relevant bits in the Noise Filter Control Register.
8. If required, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers. By default, the timer interrupt will be generated for both input capture and reload events. If required, configure the timer interrupt to be generated only at the input Capture event or the Reload event by setting TICONFIG field of the Timer Control 0 Register.
9. Configure the associated GPIO port pin for the Timer Input alternate function.
10. Write to the Timer Control 1 Register to enable the timer. Counting will start on the occurrence of the first external input transition.

In DEMODULATION Mode, the elapsed time from timer start to Capture event can be calculated using the following equation:

$$\text{Capture Elapsed Time (s)} = \frac{(\text{Capture Value} - \text{Start Value}) \times \text{Prescale}}{\text{Timer Clock Frequency (Hz)}}$$

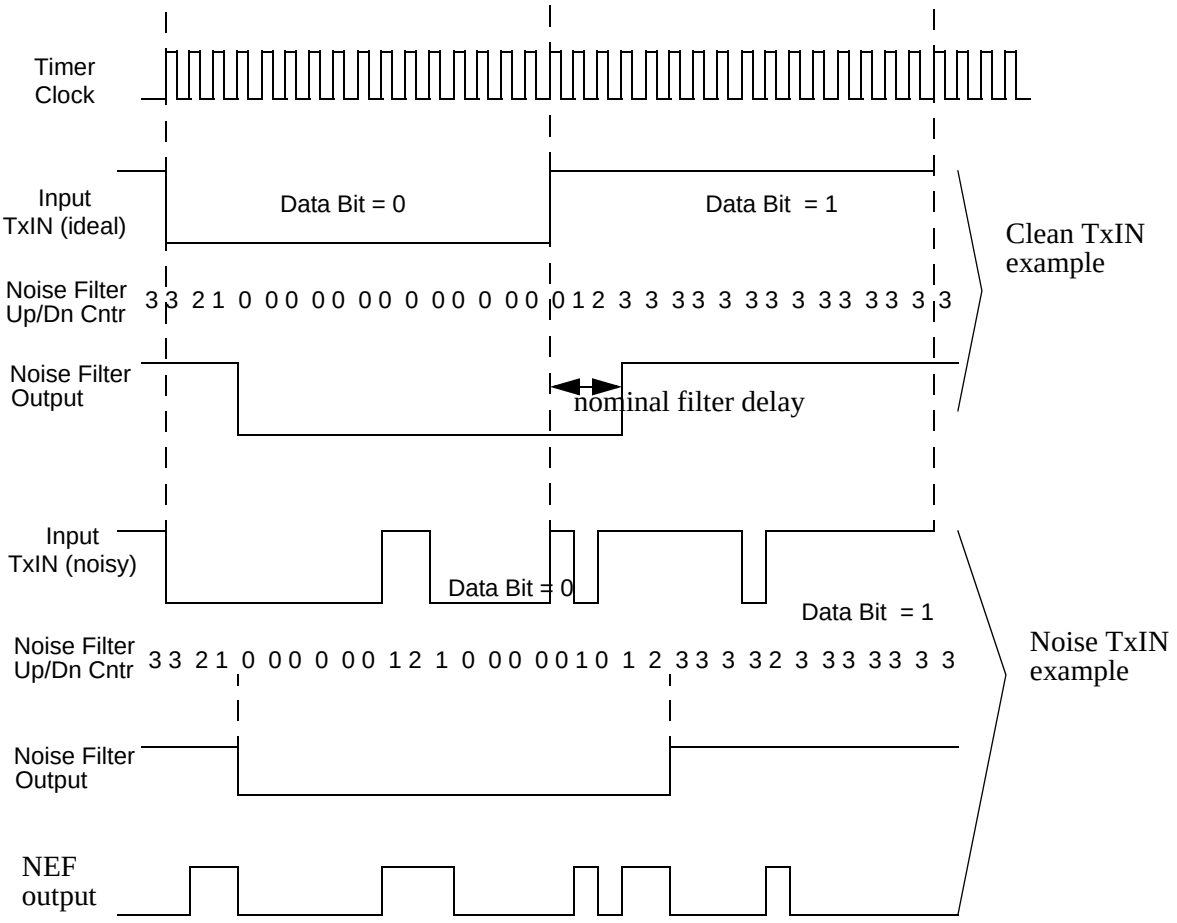


Figure 13. Noise Filter Operation

9.3. Timer Control Register Definitions

This section defines the features of the following Timer Control registers.

Timer 0–2 High and Low Byte Registers: see page 109

Timer Reload High and Low Byte Registers: see page 109

Timer 0–2 PWM0 High and Low Byte Registers: see page 110

Timer 0–2 PWM1 High and Low Byte Registers: see page 111

Timer 0–2 Control Registers: see page 112

Timer 0–2 Status Registers: see page 118

Timer 0–2 Noise Filter Control Register: see page 119

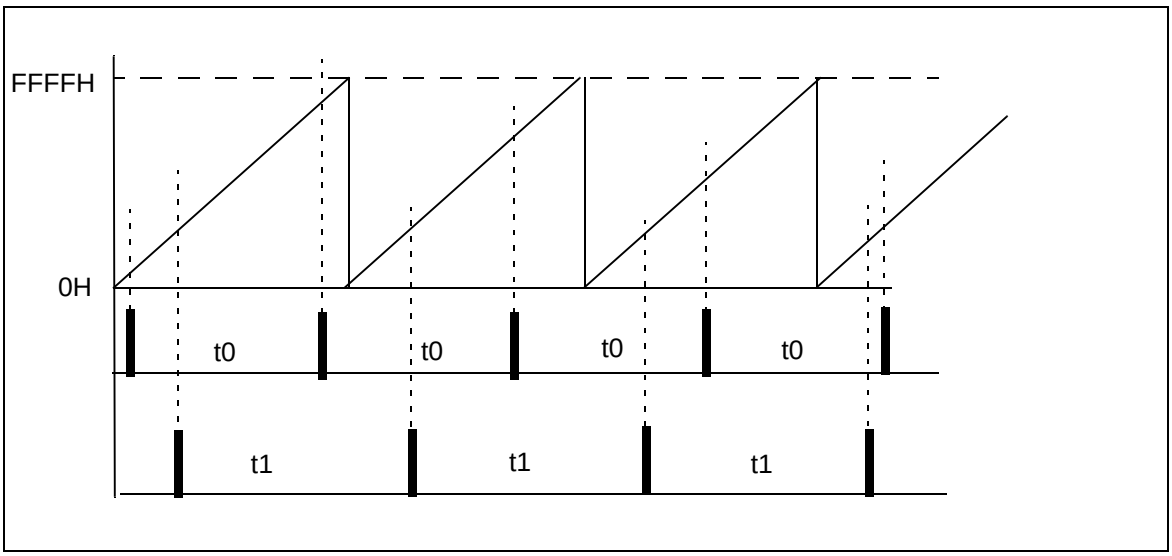


Figure 18. Count Max Mode with Channel Compare

10.7. Multi-Channel Timer Control Register Definitions

This section defines the features of the following Multi-Channel Timer Control registers.

Multi-Channel Timer High and Low Byte Registers: see page 130

Multi-Channel Timer Reload High and Low Byte Registers: see page 130

Multi-Channel Timer Subaddress Register: see page 131

Multi-Channel Timer Subregister x (0, 1, or 2): see page 132

Multi-Channel Timer Control 0, Control 1 Registers: see page 132

Multi-Channel Timer Channel Status 0 and Status 1 Registers: see page 135

Multi-Channel Timer Channel-y Control Registers: see page 137

Multi-Channel Timer Channel-y High and Low Byte Registers: see page 139

10.7.1. Multi-Channel Timer Address Map

Table 69 defines the byte address offsets for the Multi-channel Timer registers. For saving address space, a subaddress is used for the Timer Control 0, Timer Control 1, Channel Status 0, Channel Status 1, Channel-y Control, and Channel-y High and Low byte registers. Only the Timer High and Low Byte registers and the Reload High and Low Byte registers can be directly accessed.

Bit	Description (Continued)
[2] TDRE	**Transmitter Data Register Empty** This bit indicates that the Transmit Data Register is empty and ready for additional data. Writing to the Transmit Data Register resets this bit. 0 = Do not write to the Transmit Data Register. 1 = The Transmit Data Register is ready to receive an additional byte for transmission.
[1] TXE	**Transmitter Empty** This bit indicates that the Transmit Shift Register is empty and character transmission is finished. 0 = Data is currently transmitting. 1 = Transmission is complete.
[0] CTS	**Clear to Send Signal** When this bit is read it returns the level of the $\overline{\text{CTS}}$ signal. If LBEN = 1, the $\overline{\text{CTS}}$ input signal is replaced by the internal Receive Data Available signal to provide flow control in loopback mode. CTS only affects transmission if the CTSE bit = 1.

Table 86. LIN-UART Status 0 Register—LIN Mode (U0STAT0 = F41H)

Bit	7	6	5	4	3	2	1	0
Field	RDA	PLE	OE	FE	BRKD	TDRE	TXE	ATB
Reset	0	0	0	0	0	1	1	0
R/W	R	R	R	R	R	R	R	R
Address	F41H, F49H							

Note: R = Read.

Bit	Description
[7] RDA	**Receive Data Available** This bit indicates that the Receive Data Register has received data. Reading the Receive Data Register clears this bit. 0 = The Receive Data Register is empty. 1 = There is a byte in the Receive Data Register.
[6] PLE	**Physical Layer Error** This bit indicates that transmit and receive data do not match when a LIN slave or master is transmitting. This could be by a fault in the physical layer or multiple devices driving the bus simultaneously. Reading the Status 0 Register or the Receive Data Register clears this bit. 0 = Transmit and Receive data match. 1 = Transmit and Receive data do not match.

For a given LIN-UART data rate, the integer baud rate divisor value is calculated using the following equation for LIN mode UART operation:

$$\text{UART Baud Rate Divisor Value (BRG)} = \text{Round}\left(\frac{\text{System Clock Frequency (Hz)}}{\text{UART Data Rate (bits/s)}}\right)$$

The baud rate error relative to the appropriate baud rate is calculated using the following equation:

$$\text{UART Baud Rate Error (\%)} = 100 \times \left(\frac{\text{Actual Data Rate} - \text{Desired Data Rate}}{\text{Desired Data Rate}}\right)$$

For reliable communication, the LIN-UART baud rate error must never exceed 5 percent. Tables 96 through 100 provide error data for popular baud rates and commonly-used crystal oscillator frequencies for normal UART modes of operation.

Table 96. LIN-UART Baud Rates, 20.0 MHz System Clock

Applicable Rate (kHz)	BRG Divisor (Decimal)	Actual Rate (kHz)	Error (%)	Applicable Rate (kHz)	BRG Divisor (Decimal)	Actual Rate (kHz)	Error (%)
1250.0	1	1250.0	0.00	9.60	130	9.62	0.16
625.0	2	625.0	0.00	4.80	260	4.81	0.16
250.0	5	250.0	0.00	2.40	521	2.399	–0.03
115.2	11	113.64	–1.19	1.20	1042	1.199	–0.03
57.6	22	56.82	–1.36	0.60	2083	0.60	0.02
38.4	33	37.88	–1.36	0.30	4167	0.299	–0.01
19.2	65	19.23	0.16				

Table 97. LIN-UART Baud Rates, 10.0 MHz System Clock

Applicable Rate (kHz)	BRG Divisor (Decimal)	Actual Rate (kHz)	Error (%)	Applicable Rate (kHz)	BRG Divisor (Decimal)	Actual Rate (kHz)	Error (%)
1250.0	N/A	N/A	N/A	9.60	65	9.62	0.16
625.0	1	625.0	0.00	4.80	130	4.81	0.16
250.0	3	208.33	–16.67	2.40	260	2.40	–0.03
115.2	5	125.0	8.51	1.20	521	1.20	–0.03
57.6	11	56.8	–1.36	0.60	1042	0.60	–0.03

hardware detects a match to the 7-bit slave address defined in the I2CSLVAD Register and generates the slave address match interrupt (the SAM bit = 1 in the I2CISTAT Register). The I^2C controller automatically responds during the Acknowledge phase with the value in the NAK bit of the I2CCTL Register.

Slave 10-Bit Address Recognition Mode. If IRM = 0 during the address phase and the controller is configured for MASTER/SLAVE or SLAVE 10-bit address mode, the hardware detects a match to the 10-bit slave address defined in the I2CMODE and I2CSLVAD registers and generates the slave address match interrupt (the SAM bit = 1 in the I2CISTAT Register). The I^2C controller automatically responds during the Acknowledge phase with the value in the NAK bit of the I2CCTL Register.

17.2.6.2. General Call and Start Byte Address Recognition

If GCE = 1 and IRM = 0 during the address phase and the controller is configured for MASTER/SLAVE or SLAVE in either 7- or 10-bit address modes, the hardware detects a match to the General Call Address or the start byte and generates the slave address match interrupt. A General Call Address is a 7-bit address of all 0's with the R/$\overline{W}$ bit = 0. A start byte is a 7-bit address of all 0's with the R/$\overline{W}$ bit = 1. The SAM and GCA bits are set in the I2CISTAT Register. The RD bit in the I2CISTAT Register distinguishes a General Call Address from a start byte which is cleared to 0 for a General Call Address). For a General Call Address, the I^2C controller automatically responds during the address acknowledge phase with the value in the NAK bit of the I2CCTL Register. If the software is set to process the data bytes associated with the GCA bit, the IRM bit can optionally be set following the SAM interrupt to allow the software to examine each received data byte before deciding to set or clear the NAK bit. A start byte will not be acknowledged—a requirement of the I^2C specification.

17.2.6.3. Software Address Recognition

To disable hardware address recognition, the IRM bit must be set to 1 prior to the reception of the address byte(s). When IRM = 1, each received byte generates a receive interrupt (RDRF = 1 in the I2CISTAT Register). The software must examine each byte and determine whether to set or clear the NAK bit. The slave holds SCL Low during the Acknowledge phase until the software responds by writing to the I2CCTL Register. The value written to the NAK bit is used by the controller to drive the I^2C bus, then releasing the SCL. The SAM and GCA bits are not set when IRM = 1 during the address phase, but the RD bit is updated based on the first address byte.

17.2.6.4. Slave Transaction Diagrams

In the following transaction diagrams, the shaded regions indicate data transferred from the Master to the Slave and the unshaded regions indicate the data transferred from the Slave to the Master. The transaction field labels are defined as follows:

The Flash Sector Protect Register can be configured to prevent sectors from being programmed or erased. After a sector is protected, it cannot be unprotected by user code. The Flash Sector Protect Register is cleared after reset and any previously written protection values is lost. User code must write this register in their initialization routine if they want to enable sector protection.

The Flash Sector Protect Register shares its Register File address with the Page Select Register. The Flash Sector Protect Register is accessed by writing the Flash Control Register with 5EH. After the Flash Sector Protect Register is selected, it can be accessed at the Page Select Register address. When user code writes the Flash Sector Protect Register, bits can only be set to 1. Thus, sectors can be protected, but not unprotected, via register write operations. Writing a value other than 5EH to the Flash Control Register deselects the Flash Sector Protect Register and reenables access to the Page Select Register. code:

1. Write 00H to the Flash Control Register to reset the Flash Controller.
2. Write 5EH to the Flash Control Register to select the Flash Sector Protect Register.
3. Read and/or write the Flash Sector Protect Register which is now at Register File address FF9H.
4. Write 00H to the Flash Control Register to return the Flash Controller to its reset state.

The Sector Protect Register is initialized to 0 on Reset, putting each sector into an unprotected state. When a bit in the Sector Protect Register is written to 1, the corresponding sector can no longer be written or erased. After a bit of the Sector Protect Register has been set, it can not be cleared except by a System Reset.

20.2.4. Byte Programming

Flash memory is enabled for byte programming on the active page after unlocking the Flash Controller. Erase the address(es) to be programmed using either the Page Erase or Mass Erase command prior to performing byte programming. An erased Flash byte contains all 1s (FFH). The programming operation can only be used to change bits from 1 to 0. To change a Flash bit (or multiple bits) from 0 to 1 requires execution of either the Page Erase or Mass Erase command.

Byte programming can be accomplished using the On-Chip Debugger's Write Memory command or eZ8 CPU execution of the LDC or LDCI instructions. For a description of the LDC and LDCI instructions, refer to the eZ8 CPU Core User Manual (UM0128), available for download at www.zilog.com. While the Flash Controller programs the contents of Flash memory, the eZ8 CPU idles but the system clock and on-chip peripherals continue to operate.

After a byte is written, the page remains unlocked, allowing for subsequent writes to other bytes on the same page. To exit programming mode and lock Flash memory, write any value to the Flash Control Register except the Mass Erase or Page Erase commands.

Table 147. Trim Option Bits at 0001H (TTEMP1)

Bits	7	6	5	4	3	2	1	0
Field	TS_NEG				TS_COARSE			
Reset	1	0	1	0	0	1	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Information Page Memory 0021H							
Note: U = Unchanged by Reset. R/W = Read/Write.								

Bit	Description
[7:4] TS_NEG	**Temperature Sensor Negative Control Trim Bits** Negative control offset trimming bits for the Temperature Sensor.
[3:0] TS_COARSE	**Temperature Sensor Coarse Control Trim Bits** Contains coarse control offset trimming bits for the Temperature Sensor.

21.2.4.2. Trim Bit Address 0002H

The Trim Option Bits Register at address 0002H, shown in Table 148, governs control of the Internal Precision Oscillator trim bits.

Table 148. Trim Option Bits at 0002H (TIPO)

Bits	7	6	5	4	3	2	1	0
Field	IPO_TRIM							
Reset	U							
R/W	R/W							
Address	Information Page Memory 0022H							
Note: U = Unchanged by Reset. R/W = Read/Write.								

Bit	Description
[7:0] IPO_TRIM	**Internal Precision Oscillator Trim Byte** Contains trimming bits for Internal Precision Oscillator.

Bit	Description
[3] BRKPC	**Break when PC == OCDCNTR** If this bit is set to 1, then the OCDCNTR Register is used as a hardware breakpoint. When the program counter matches the value in the OCDCNTR Register, DBGMODE is automatically set to 1. If this bit is set, the OCDCNTR Register does not count when the CPU is running. 0 = OCDCNTR is set up as a counter. 1 = OCDCNTR generates a hardware break when PC == OCDCNTR.
[2] BRKZRO	**Break when OCDCNTR == 0000H** If this bit is set, then the OCD automatically sets the DBGMODE bit when the OCDCNTR Register counts down to 0000H. If this bit is set, the OCDCNTR Register is not reset when the part exits DEBUG Mode. 0 = OCD does not generate BRK when OCDCNTR decrements to 0000H. 1 = OCD sets DBGMODE to 1 when OCDCNTR decrements to 0000H.
[1]	Reserved; must be 0.
[0] RST	**Reset** Setting this bit to 1 resets the device. The controller goes through a normal POR sequence with the exception that the On-Chip Debugger is not reset. This bit is automatically cleared to 0 when the reset finishes. 0 = No effect. 1 = Reset the device.

Table 184. Program Control Instructions

Mnemonic	Operands	Instruction
BRK	—	On-Chip Debugger Break
BTJ	p, bit, src, DA	Bit Test and Jump
BTJNZ	bit, src, DA	Bit Test and Jump if Non-Zero
BTJZ	bit, src, DA	Bit Test and Jump if Zero
CALL	dst	Call Procedure
DJNZ	dst, src, RA	Decrement and Jump Non-Zero
IRET	—	Interrupt Return
JP	dst	Jump
JP cc	dst	Jump Conditional
JR	DA	Jump Relative
JR cc	DA	Jump Relative Conditional
RET	—	Return
TRAP	vector	Software Trap

Table 185. Rotate and Shift Instructions

Mnemonic	Operands	Instruction
BSWAP	dst	Bit Swap
RL	dst	Rotate Left
RLC	dst	Rotate Left through Carry
RR	dst	Rotate Right
RRC	dst	Rotate Right through Carry
SRA	dst	Shift Right Arithmetic
SRL	dst	Shift Right Logical
SWAP	dst	Swap Nibbles

Upper Nibble (Hex) \ Lower Nibble (Hex)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2																
3																
4																
5																
6																
7	3, 2 **PUSH** IM															
8																
9																
A			3.3 **CPC** r1,r2	3.4 **CPC** r1,Ir2	4.3 **CPC** R2,R1	4.4 **CPC** IR2,R1	4.3 **CPC** R1,IM	4.4 **CPC** IR1,IM	5.3 **CPCX** ER2,ER1	5.3 **CPCX** IM,ER1						
B																
C	3.2 **SRL** R1	3.3 **SRL** IR1														
D																
E									5, 4 **LDWX** ER2,ER1							
F																

Figure 68. Second Op Code Map after 1FH