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	23
Program Memory Size	24KB (24K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3K 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	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f2480sj020sg

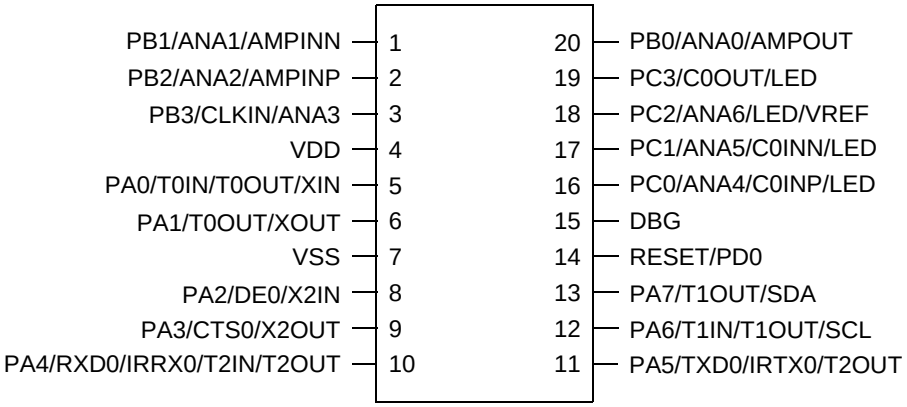

Figure 2. Z8F2480, Z8F1680 and Z8F0880 in 20-Pin SOIC, SSOP or PDIP Packages

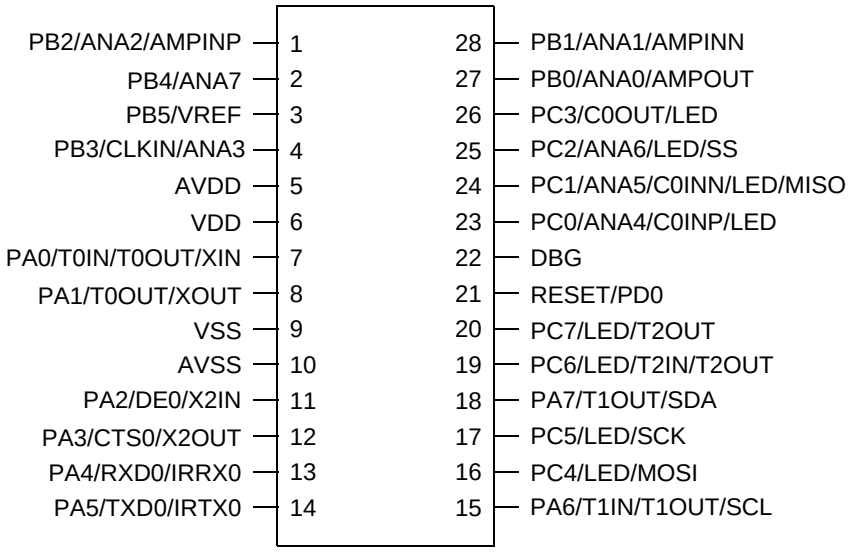

Figure 3. Z8F2480, Z8F1680 and Z8F0880 in 28-Pin SOIC, SSOP or PDIP Packages

5.2.4. External Reset Input

The $\overline{\text{RESET}}$ pin has a Schmitt-triggered input and an internal pull-up resistor. When the $\overline{\text{RESET}}$ pin is asserted for a minimum of four system clock cycles, the device progresses through the System Reset sequence. Because of the possible asynchronicity of the system clock and reset signals, the required reset duration can be as short as three clock periods and as long as four. A reset pulse three clock cycles in duration might trigger a Reset; a pulse four cycles in duration always triggers a Reset.

While the $\overline{\text{RESET}}$ input pin is asserted Low, the F1680 Series MCU remains in the Reset state. If the $\overline{\text{RESET}}$ pin is held Low beyond the System Reset time-out, the device exits the Reset state on the system clock rising edge following $\overline{\text{RESET}}$ pin deassertion. Following a System Reset initiated by the external $\overline{\text{RESET}}$ pin, the EXT status bit in the RSTSTAT Register is set to 1.

5.2.5. External Reset Indicator

During System Reset or when enabled by the GPIO logic (see the Port A–E Control Registers section on page 60), the $\overline{\text{RESET}}$ pin functions as an open-drain (active Low) reset mode indicator in addition to the input functionality. This Reset output feature allows the F1680 Series MCU to reset other components to which it is connected, even if that reset is caused by internal sources such as POR, VBO, or WDT events.

After an internal Reset event occurs, the internal circuitry begins driving the $\overline{\text{RESET}}$ pin Low. The $\overline{\text{RESET}}$ pin is held Low by the internal circuitry until the appropriate delay listed in Table 9 on page 32 has elapsed.

5.2.6. On-Chip Debugger Initiated Reset

A POR can be initiated using the OCD by setting the RST bit in the OCD Control Register. The OCD block is not reset, but the rest of the chip goes through a normal System Reset. The RST bit automatically clears during the system reset. Following the System Reset the POR bit in the WDT Control Register is set.

5.3. Stop Mode Recovery

STOP Mode is entered by execution of a stop instruction by the eZ8 CPU. For detailed STOP Mode information, see the Low-Power Modes section on page 42. During Stop Mode Recovery, the CPU is held in reset for 4 IPO cycles.

Stop Mode Recovery does not affect On-chip registers other than the Reset Status (RSTSTAT) register and the Oscillator Control Register (OSCCTL). After any Stop Mode Recovery, the IPO is enabled and selected as the system clock. If another system clock source is required or IPO disabling is required, the Stop Mode Recovery code must

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.

7.11.14. LED Drive Level Registers

Two LED Drive Level registers consist of the LED Drive Level High Bit Register (LEDLVLH[7:0]) and the LED Drive Level Low Bit Register (LEDLVLL[7:0]), as shown in Tables 34 and 35. Two control bits, LEDLVLH[x] and LEDLVLL[x], are used to select one of four programmable current drive levels for each associated Port C[x] pin. Each Port C pin is individually programmable.

Table 34. LED Drive Level High Bit Register (LEDLVLH)

Bits	7	6	5	4	3	2	1	0
Field	LEDLVLH							
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	F83H							

Table 35. LED Drive Level Low Bit Register (LEDLVLL)

Bits	7	6	5	4	3	2	1	0
Field	LEDLVLL							
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	F84H							

Bit	Description
[7:0] LEDLVLH, LEDLVLL	**LED Drive Level High Bit** **LED Drive Level Low Bit** These bits are used to set the LED drive current. {LEDLVLH[x], LEDLVLL[x]}, in which x=Port C[0] to Port C[7]. Select one of the following four programmable current drive levels for each Port C pin. 00 = 3 mA 01 = 7 mA 10 = 13 mA 11 = 20 mA

9.1. Architecture

Figure 11 displays the architecture of the timers.

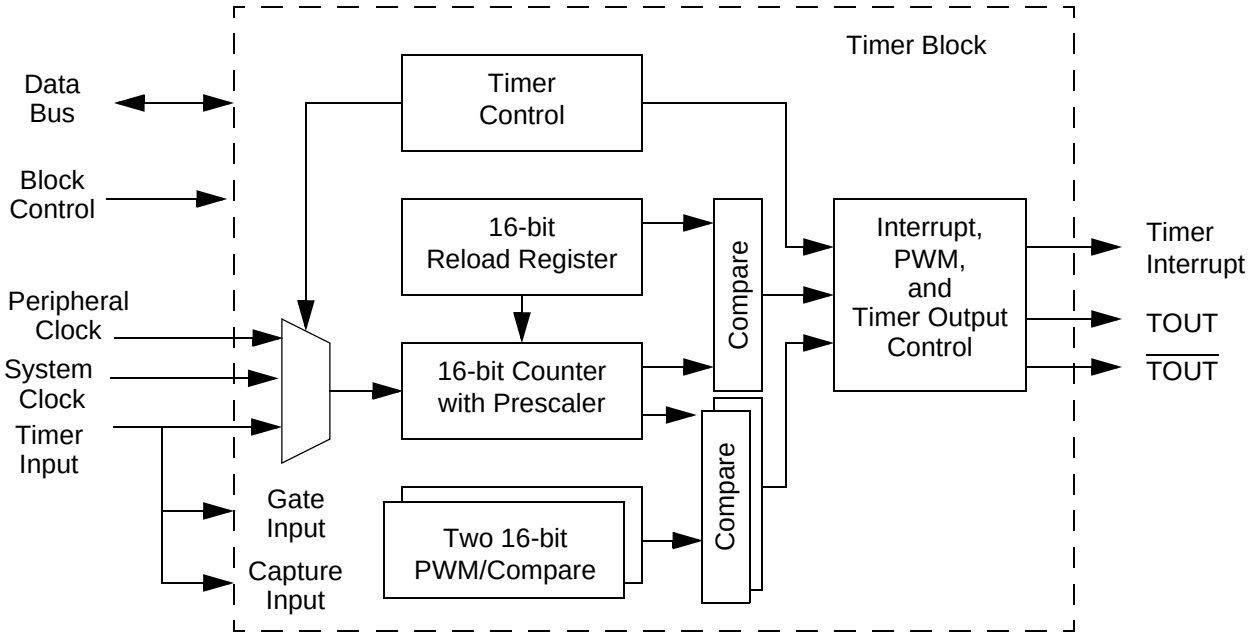

Figure 11. Timer Block Diagram

9.2. Operation

The timers are 16-bit up-counters. Minimum time-out delay is set by loading the value 0001H into the Timer Reload High and Low Byte registers and setting the prescale value to 1. Maximum time-out delay is set by loading the value 0000H into the Timer Reload High and Low Byte registers and setting the prescale value to 128. If the Timer reaches FFFFH, the timer rolls over to 0000H and continues counting.

9.2.1. Timer Clock Source

The timer clock source can come from either the peripheral clock or the system clock. Peripheral clock is based on a low frequency/low power 32 kHz secondary oscillator that can be used with external watch crystal. Peripheral clock source is only available for driving Timer and Noise Filter operation. It is not supported for other peripherals.

For timer operation in STOP Mode, peripheral clock must be selected as the clock source. Peripheral clock can be selected as source for both ACTIVE and STOP Mode operation.

10.3.3. PWM Output Operation

In a PWM OUTPUT operation, the timer generates a PWM output signal on the channel output pin (T4CHA, T4CHB, T4CHC, or T4CHD). The channel output toggles whenever the timer count matches the channel compare value (defined in the MCTCHyH and MCTCHyL registers). In addition, a channel interrupt is generated and the channel event flag is set in the status register. The timer continues counting according to its programmed mode.

The channel output signal begins with the output value = CHPOL and then transitions to $\overline{\text{CHPOL}}$ when timer value matches the PWM value. If timer mode is Count Modulo, the channel output signal returns to output = CHPOL when timer reaches the reload value and is reset. If timer mode is Count up/down, channel output signal returns to output = CHPOL when the timer count matches the PWM value again (when counting down).

10.3.4. Capture Operation

In a CAPTURE operation, the current timer count is recorded when the selected transition occurs on T4CHA, T4CHB, T4CHC or T4CHD. The Capture count value is written to the Channel High and Low Byte registers. In addition, a channel interrupt is generated and the channel event flag (CHyEF) is set in the Channel Status Register. The CHPOL bit in the Channel Control Register determines if the Capture occurs on a rising edge or a falling edge of the Channel Input signal. The timer continues counting according to the programmed mode.

10.4. Multi-Channel Timer Interrupts

The Multi-Channel Timer provides a single interrupt which has five possible sources. These sources are the internal timer and the four channel inputs (T4CHA, T4CHB, T4CHC, T4CHD).

10.4.1. Timer Interrupt

If enabled by the TCIEN bit of the MCTCTL0 Register, the timer interrupt will be generated when the timer completes a count cycle. This occurs during transition from counter = reload register value to counter = 0 in count modulo mode and occurs during transition from counter = 1 to counter = 0 in count up/down mode.

10.4.2. Capture/Compare Channel Interrupt

A channel interrupt is generated whenever there is a successful Capture/Compare Event on the Timer Channel and the associated CHIEN bit is set.

While writing a subregister, first write the subaddress to Timer Subaddress Register, then write data to subregister0, subregister1, or subregister2. A read is the same as a write.

Table 69. Multi-Channel Timer Address Map

Address/Subaddress	Register/Subregister Name
Direct Access Register	
FA0	Timer (Counter) High
FA1	Timer (Counter) Low
FA2	Timer Reload High
FA3	Timer Reload Low
FA4	Timer Subaddress
FA5	Subregister 0
FA6	Subregister 1
FA7	Subregister 2
Subregister 0	
0	Timer Control 0
1	Channel Status 0
2	Channel A Capture/Compare High
3	Channel B Capture/Compare High
4	Channel C Capture/Compare High
5	Channel D Capture/Compare High
Subregister 1	
0	Timer Control 1
1	Channel Status 1
2	Channel A Capture/Compare Low
3	Channel B Capture/Compare Low
4	Channel C Capture/Compare Low
5	Channel D Capture/Compare Low
Subregister 2	
0	Reserved
1	Reserved
2	Channel A Control
3	Channel B Control
4	Channel C Control
5	Channel D Control

Bit	Description (Continued)
[2] SBRK	**Send Break** This bit pauses or breaks data transmission. Sending a break interrupts any transmission in progress, so ensure that the transmitter has completed sending data before setting this bit. In standard UART mode, the duration of the break is determined by how long the software leaves this bit asserted. Also the duration of any required stop bits following the break must be timed by software before writing a new byte to be transmitted to the Transmit Data Register. In LIN mode, the master sends a Break character by asserting SBRK. The duration of the break is timed by hardware and the SBRK bit is deasserted by hardware when the Break is completed. The duration of the Break is determined by the TxBreakLength field of the LIN Control Register. One or two stop bits are automatically provided by the hardware in LIN mode as defined by the stop bit. 0 = No break is sent. 1 = The output of the transmitter is 0.
[1] STOP	**Stop Bit Select** 0 = The transmitter sends one stop bit. 1 = The transmitter sends two stop bits.
[0] LBEN	**Loop Back Enable** 0 = Normal operation. 1 = All transmitted data is looped back to the receiver within the IrDA module.

12.3.6. LIN-UART Control 1 Registers

Multiple registers, shown in Tables 90 and 91, are accessible by a single bus address. The register selected is determined by the Mode Select (MSEL) field. These registers provide additional control over LIN-UART operation.

Table 100. LIN-UART Baud Rates, 1.8432 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	12	9.60	0.00
625.0	N/A	N/A	N/A	4.80	24	4.80	0.00
250.0	N/A	N/A	N/A	2.40	48	2.40	0.00
115.2	1	115.2	0.00	1.20	96	1.20	0.00
57.6	2	57.6	0.00	0.60	192	0.60	0.00
38.4	3	38.4	0.00	0.30	384	0.30	0.00
19.2	6	19.2	0.00				

Bit	Description (Continued)
[6,0] ESPIEN1, ESPIEN0	**ESPI Enable and Direction Control** 00 = The ESPI block is disabled. BRG can be used as a general-purpose timer by setting BRGCTL = 1. 01 = Receive Only Mode. Use this setting in SLAVE Mode if software application is receiving data but not sending. TDRE will not assert. Transmitted data will be all 1s. Not valid in MASTER Mode since Master must source data to drive the transfer. 10 = Transmit Only Mode Use this setting in MASTER or SLAVE Mode when the software application is sending data but not receiving. RDRNE will not assert. 11 = Transmit/Receive Mode Use this setting if the software application is both sending and receiving information. Both TDRE and RDRNE will be active.
[5] BRGCTL	**Baud Rate Generator Control** The function of this bit depends upon ESPIEN1,0. When ESPIEN1,0 = 00, this bit allows enabling the BRG to provide periodic interrupts. **If the ESPI is disabled** 0 = The Baud Rate Generator timer function is disabled. Reading the Baud Rate High and Low registers returns the BRG reload value. 1 = The Baud Rate Generator timer function and time-out interrupt is enabled. Reading the Baud Rate High and Low registers returns the BRG Counter value. **If the ESPI is enabled** 0 = Reading the Baud Rate High and Low registers returns the BRG reload value. If MMEN = 1, the BRG is enabled to generate SCK. If MMEN = 0, the BRG is disabled. 1 = Reading the Baud Rate High and Low registers returns the BRG Counter value. If MMEN = 1, the BRG is enabled to generate SCK. If MMEN = 0 the BRG is enabled to provide a Slave SCK time-out. See the SLAVE Mode Abort error description on page 211. **Caution:** If reading the counter one byte at a time while the BRG is counting keep in mind that the values will not be in sync. Zilog recommends reading the counter using (2-byte) word reads.
[4] PHASE	**Phase Select** Sets the phase relationship of the data to the clock. For more information about operation of the PHASE bit, see the ESPI Clock Phase and Polarity Control section on page 201.
[3] CLKPOL	**Clock Polarity** 0 = SCK idles Low (0). 1 = SCK idles High (1).
[2] WOR	**Wire-OR (Open-Drain) Mode Enabled** 0 = ESPI signal pins not configured for open-drain. 1 = All four ESPI signal pins (SCK, $\overline{SS}$, MISO and MOSI) configured for open-drain function. This setting is typically used for multi-Master and/or Multi-Slave configurations.
[1] MMEN	**ESPI MASTER Mode Enable** This bit controls the data I/O pin selection and SCK direction. 0 = Data out on MISO, data in on MOSI (used in SPI SLAVE Mode), SCK is an input. 1 = Data out on MOSI, data in on MISO (used in SPI MASTER Mode), SCK is an output.

transferred from the Slave to the Master. The transaction field labels are defined as follows:

S Start
W Write
A Acknowledge
$\overline{A}$ Not Acknowledge
P Stop

17.2.5.4. Master Write Transaction with a 7-Bit Address

Figure 43 displays the data transfer format from a Master to a 7-bit addressed slave.

S	Slave Address	W = 0	A	Data	A	Data	A	Data	A/$\overline{A}$	P/S

Figure 43. Data Transfer Format—Master Write Transaction with a 7-Bit Address

Observe the following steps for a Master transmit operation to a 7-bit addressed slave:

1. The software initializes the MODE field in the I^2C Mode Register for MASTER/SLAVE Mode with either a 7-bit or 10-bit slave address. The MODE field selects the address width for this mode when addressed as a slave (but not for the remote slave). The software asserts the IEN bit in the I^2C Control Register.
2. The software asserts the TXI bit of the I^2C Control Register to enable transmit interrupts.
3. The I^2C interrupt asserts, because the I^2C Data Register is empty.
4. The software responds to the TDRE bit by writing a 7-bit slave address plus the Write bit (which is cleared to 0) to the I^2C Data Register.
5. The software sets the start bit of the I^2C Control Register.
6. The I^2C controller sends a start condition to the I^2C slave.
7. The I^2C controller loads the I^2C Shift Register with the contents of the I^2C Data Register.
8. After one bit of the address has been shifted out by the SDA signal, the transmit interrupt asserts.
9. The software responds by writing the transmit data into the I^2C Data Register.
10. The I^2C controller shifts the remainder of the address and the Write bit out via the SDA signal.

11. The I^2C slave sends an Acknowledge (by pulling the SDA signal Low) during the next High period of SCL. The I^2C controller sets the ACK bit in the I^2C Status Register.

    If the slave does not acknowledge the address byte, the I^2C controller sets the NCKI bit in the I^2C Status Register, sets the ACKV bit and clears the ACK bit in the I^2C State Register. The software responds to the Not Acknowledge interrupt by setting the stop bit and clearing the TXI bit. The I^2C controller flushes the Transmit Data Register, sends a stop condition on the bus and clears the stop and NCKI bits. The transaction is complete and the following steps can be ignored.
12. The I^2C controller loads the contents of the I^2C Shift Register with the contents of the I^2C Data Register.
13. The I^2C controller shifts the data out via the SDA signal. After the first bit is sent, the transmit interrupt asserts.
14. If more bytes remain to be sent, return to Step 9.
15. When there is no more data to be sent, the software responds by setting the stop bit of the I^2C Control Register (or the start bit to initiate a new transaction).
16. If no additional transaction is queued by the master, the software can clear the TXI bit of the I^2C Control Register.
17. The I^2C controller completes transmission of the data on the SDA signal.
18. The I^2C controller sends a stop condition to the I^2C bus.

Note: If the slave terminates the transaction early by responding with a Not Acknowledge during the transfer, the I^2C controller asserts the NCKI interrupt and halts. The software must terminate the transaction by setting either the stop bit (end transaction) or the start bit (end this transaction, start a new one). In this case, it is not necessary for software to set the FLUSH bit of the I2CCTL Register to flush the data that was previously written but not transmitted. The I^2C controller hardware automatically flushes transmit data in the not acknowledge case.

17.2.5.5. Master Write Transaction with a 10-Bit Address

Figure 44 displays the data transfer format from a Master to a 10-bit addressed slave.

S	Slave Address 1st Byte	W = 0	A	Slave Address 2nd Byte	A	Data	A	Data	A/$\overline{A}$	F/S

Figure 44. Data Transfer Format—Master Write Transaction with a 10-Bit Address

5. The I^2C controller receives the data byte and responds with Acknowledge or Not Acknowledge depending on the state of the NAK bit in the I2CCTL Register. The I^2C controller generates the receive data interrupt by setting the RDRF bit in the I2CISTAT Register.
6. The software responds by reading the I2CISTAT Register, finding the RDRF bit = 1 and reading the I2CDATA Register clearing the RDRF bit. If software can accept only one more data byte it sets the NAK bit in the I2CCTL Register.
7. The master and slave loops through Step 4 to Step 6 until the master detects a Not Acknowledge instruction or runs out of data to send.
8. The master sends the stop or restart signal on the bus. Either of these signals can cause the I^2C controller to assert a stop interrupt (the stop bit = 1 in the I2CISTAT Register). Because the slave received data from the master, the software takes no action in response to the stop interrupt other than reading the I2CISTAT Register to clear the stop bit in the I2CISTAT Register.

17.2.6.6. Slave Receive Transaction with 10-Bit Address

The data transfer format for writing data from a master to a slave with 10-bit addressing is displayed in Figure 48. The procedure that follows describes the I^2C Master/Slave Controller operating as a slave in 10-bit addressing mode and receiving data from the bus master.

S	Slave Address 1st Byte	W=0	A	Slave Address 2nd Byte	A	Data	A	Data	A/$\overline{A}$	P/S

Figure 48. Data Transfer Format—Slave Receive Transaction with 10-Bit Address

1. The software configures the controller for operation as a slave in 10-bit addressing mode, as follows:
   a. Initialize the MODE field in the I2CMODE Register for either SLAVE ONLY mode or MASTER/SLAVE Mode with 10-bit addressing.
   b. Optionally set the GCE bit.
   c. Initialize the SLA[7:0] bits in the I2CSLVAD Register and the SLA[9:8] bits in the I2CMODE Register.
   d. Set IEN = 1 in the I2CCTL Register. Set NAK = 0 in the I^2C Control Register.
2. The Master initiates a transfer, sending the first address byte. The I^2C controller recognizes the start of a 10-bit address with a match to SLA[9:8] and detects R/$\overline{W}$ bit = 0 (a Write from the master to the slave). The I^2C controller acknowledges, indicating it is available to accept the transaction.
3. The Master sends the second address byte. The SLAVE Mode I^2C controller detects an address match between the second address byte and SLA[7:0]. The SAM bit in the

Table 126. I2CSTATE_H

State Encoding	State Name	State Description
0000	Idle	I^2C bus is idle or I^2C controller is disabled.
0001	Slave Start	I^2C controller has received a start condition.
0010	Slave Bystander	Address did not match; ignore remainder of transaction.
0011	Slave Wait	Waiting for stop or restart condition after sending a Not Acknowledge instruction.
0100	Master Stop2	Master completing stop condition (SCL = 1, SDA = 1).
0101	Master Start/Restart	MASTER Mode sending start condition (SCL = 1, SDA = 0).
0110	Master Stop1	Master initiating stop condition (SCL = 1, SDA = 0).
0111	Master Wait	Master received a Not Acknowledge instruction, waiting for software to assert stop or start control bits.
1000	Slave Transmit Data	Nine substates, one for each data bit and one for the Acknowledge.
1001	Slave Receive Data	Nine substates, one for each data bit and one for the Acknowledge.
1010	Slave Receive Addr1	Slave receiving first address byte (7- and 10-bit addressing) Nine substates, one for each address bit and one for the Acknowledge.
1011	Slave Receive Addr2	Slave Receiving second address byte (10-bit addressing) nine substates, one for each address bit and one for the Acknowledge.
1100	Master Transmit Data	Nine substates, one for each data bit and one for the Acknowledge.
1101	Master Receive Data	Nine substates, one for each data bit and one for the Acknowledge.
1110	Master Transmit Addr1	Master sending first address byte (7- and 10-bit addressing) nine substates, one for each address bit and one for the Acknowledge.
1111	Master Transmit Addr2	Master sending second address byte (10-bit addressing) nine substates, one for each address bit and one for the Acknowledge.

Table 127. I2CSTATE_L

State I2CSTATE_H	Substate I2CSTATE_L	Substate Name	State Description
0000–0100	0000	—	There are no substates for these I2CSTATE_H values.
0110–0111	0000	—	There are no substates for these I2CSTATE_H values.
0101	0000	Master Start	Initiating a new transaction
	0001	Master Restart	Master is ending one transaction and starting a new one without letting the bus go idle.

21.2.2. Trim Bit Data Option Bits

The Trim Bit Data Register, shown in Table 142, contains the read or write data for access to the trim option bits.

Table 142. Trim Bit Data Register (TRMDR)

Bits	7	6	5	4	3	2	1	0
Field	TRMDR—Trim Bit Data							
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	FF7H							

21.2.3. Trim Bit Address Option Bits

The Trim Bit Address Register, shown in Table 143, contains the target address for access to the trim option bits. Trim Bit addresses in the range 00H-1FH map to the Information Area address range 20H-3FH, as indicated in Table 142.

Table 143. Trim Bit Address Register (TRMADR)

Bits	7	6	5	4	3	2	1	0
Field	TRMADR—Trim Bit Address (00H to 1FH)							
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	FF6H							

Table 144. Trim Bit Address Map

Trim Bit Address	Information Area Address
00H	20H
01H	21H
02H	22H
03H	23H
:	:
1FH	3FH

21.2.4.4. Trim Bit Address 0004H

The Trim Option Bits Register at address 0004H, shown in Table 151, governs control of the Temperature Sensor Test, Comparator, and ADC trim bits.

Table 151. Trim Option Bits at 0004H (TCOMP_ADC)

Bits	7	6	5	4	3	2	1	0
Field	TEMPTST	COMP1_OPT	Reserved	COMP0_OPT	ADC_OPT			
Reset	U	U	U	U	U	U	U	U
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Information Page Memory 0024H							

Note: U = Unchanged by Reset. R/W = Read/Write.

Bit	Description
[7] TEMPTST	**Temperature Sensor Test Control Bit** The default is 1.
[6] COMP1_OPT	**Comparator 1 Trim Bit** This COMP1_OPT bit controls the comparator's *hys* input; see Table 152.
[5]	Reserved; must be 0.
[4] COMP0_OPT	**Comparator 0 Trim Bit** This COMP0_OPT bit controls the comparator's *hys* input; see Table 152.
[3:0] ADC_OPT	**ADC Trim Values** Contains factory trimmed values for the ADC block.

Table 152. Truth Table of HYS

HYS	Hysteresis Input
1	Enable
0	Disable

Table 173. Recommended Crystal Oscillator Specifications

Parameter	Value	Units	Comments
Frequency	32	kHz	
Resonance	Serial		
Mode	Fundamental		
Series Resistance (R_S)	160K	W	Maximum
Load Capacitance (C_L)	30	pF	Maximum
Shunt Capacitance (C_0)	5	pF	Maximum
Drive Level	0.1	mW	Maximum

The currents in Table 190 represent the power consumption without any peripherals active (unless otherwise noted). For design guidance, total power consumption will be the sum of all active peripheral currents plus the appropriate current characteristics shown below.

Table 190. Supply Current Characteristics

		T_A = 0°C to +70°C T_A = –40°C to +105°C				
Symbol	**Parameter**	**Min**	**Typical[1]**	**Max**	**Units**	**Conditions[2]**
I_{DDA1}	Active Mode Device Current Executing from Flash		8.5		mA	Typical: 20MHz[3, 4, 5, 6], V_{DD} = 3V, Flash, 25°C
I_{DDA2}	Active Mode Device Current Executing from PRAM		6		mA	Typical: 20MHz[3, 4, 5, 6], V_{DD} = 3V, PRAM, 25°C
I_{DDH}	Halt Mode Device Current		TBD		mA	Typical: 20MHz[3, 4, 5], V_{DD} typical, 25°C
I_{DDS1}	Stop Mode Device Current		2.5		µA	Typical: WDT, V_{DD} typical, 25 °C, all peripherals including VBO disabled[3, 4, 6]
I_{DDS2}	Stop Mode Device Current		<1		µA	Typical: V_{DD} typical, 25°C, all peripherals disabled including VBO and WDT[3, 4, 6]

Notes
1. These values are provided for design guidance only and are not tested in production.
2. Typical conditions are defined as 3.3 V at 25°C, unless otherwise noted.
3. All internal pull ups are disabled and all push-pull outputs are unloaded.
4. All open-drain outputs are pulled up to V_{DD}/AV_{DD} and are at a High state.
5. System clock source is an external square wave clock signal driven through the CLK-IN pin.
6. All inputs are at V_{DD}/AV_{DD} or V_{SS}/AV_{SS} as appropriate.

V

W

X

Z

Customer Support

To share comments, get your technical questions answered, or report issues you may be experiencing with our products, please visit Zilog's Technical Support page at http://support.zilog.com.

To learn more about this product, find additional documentation, or to discover other facets about Zilog product offerings, please visit the Zilog Knowledge Base at http://zilog.com/kb or consider participating in the Zilog Forum at http://zilog.com/forum.

This publication is subject to replacement by a later edition. To determine whether a later edition exists, please visit the Zilog website at http://www.zilog.com.