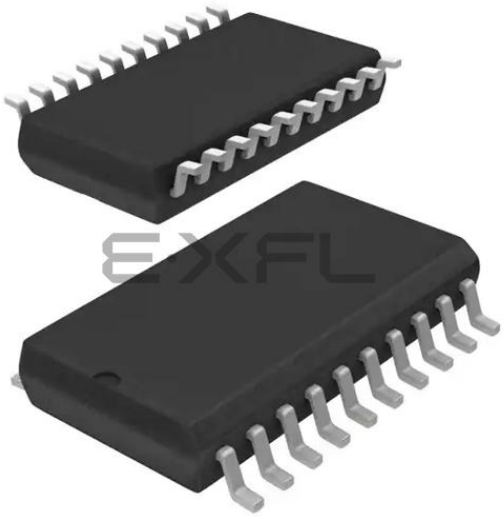

Welcome to E-XFL.COM

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, UART/USART
Peripherals	Brown-out Detect/Reset, LED, LVD, POR, PWM, Temp Sensor, WDT
Number of I/O	17
Program Memory Size	16KB (16K 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 7x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f1680sh020eg

- Optional 16-bit Multi-Channel Timer which supports four Capture/Compare/PWM modules (44-pin packages only)
- Watchdog Timer (WDT) with dedicated internal RC oscillator
- 17 to 37 General-Purpose Input/Output (GPIO) pins depending upon package
- Up to 8 direct LED drives with programmable drive current capability
- Up to 31 interrupt sources with up to 24 interrupt vectors
- On-Chip Debugger (OCD)
- Power-On Reset (POR) and Voltage Brown-Out (VBO) protection
- Built-in Low-Voltage Detection (LVD) with programmable voltage threshold
- 32kHz secondary oscillator for Timers
- Internal Precision Oscillator (IPO) with output frequency in the range of 43.2kHz to 11.0592MHz
- Crystal oscillator with three power settings and external RC network option
- Wide operation voltage range: 1.8V–3.6V
- 20-, 28-, 40- and 44-pin packages
- 0°C to +70°C (standard) and –40°C to +105°C (extended) operating temperature ranges

1.2. Part Selection Guide

Table 1 displays basic features and package styles available for each of the F1680 Series MCUs.

Table 1. Z8 Encore! XP F1680 Series Part Selection Guide

Part Number	Flash (KB)	RAM (B)	Program RAM (B)	NVDS (B)	I/O	ADC Inputs	SPI	I^2C	UARTs	Packages
Z8F2480	24	2048	1024	—	17–37	7–8	0–1	1	1–2	20-, 28-, 40- and 44-pin
Z8F1680	16	2048	1024	256	17–37	7–8	0–1	1	1–2	20-, 28-, 40- and 44-pin
Z8F0880	8	1024	1024	128	17–37	7–8	0–1	1	1–2	20-, 28-, 40- and 44-pin

Table 8. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)[1]	Page #
Analog-to-Digital Converter (ADC)				
F70	ADC Control 0	ADCCTL0	00	189
F71	ADC Raw Data High Byte	ADCRD_H	80	191
F72	ADC Data High Byte	ADCD_H	XX	191
F73	ADC Data Low Bits	ADCD_L	XX	192
F74	ADC Sample Settling Time	ADCSST	FF	193
F75	Sample Time	ADCST	XX	194
F76	ADC Clock Prescale Register	ADCCP	00	195
F77–F7F	Reserved	—	XX	
Low-Power Control				
F80	Power Control 0	PWRCTL0	80	44
F81	Reserved	—	XX	
LED Controller				
F82	LED Drive Enable	LEDEN	00	66
F83	LED Drive Level High Bit	LEDLVLH	00	67
F84	LED Drive Level Low Bit	LEDLVLL	00	67
F85	Reserved	—	XX	
Oscillator Control				
F86	Oscillator Control 0	OSCCTL0	A0	319
F87	Oscillator Control 1	OSCCTL1	00	320
F88–F8F	Reserved			
Comparator 0				
F90	Comparator 0 Control	CMP0	14	257
Comparator 1				
F91	Comparator 1 Control	CMP1	14	258
F92–F9F	Reserved	—	XX	

Notes:
1. XX=Undefined.
2. The Reserved space can be configured as General-Purpose Register File RAM depending on the user option bits (see the User Option Bits chapter on page 277) and the on-chip PRAM size (see the Ordering Information chapter on page 372). If the PRAM is programmed as General-Purpose Register File RAM on Reserved space, the starting address always begins immediately after the end of General-Purpose Register File RAM.

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

Table 17. Port Alternate Function Mapping, 20-Pin Parts[1,2] (Continued)

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Register AFS1
Port B	PB0	Reserved		AFS1[0]: 0
		ANA0/AMPOUT	ADC Analog Input/LPO Output	AFS1[0]: 1
	PB1	Reserved		AFS1[1]: 0
		ANA1/AMPINN	ADC Analog Input/LPO Input (N)	AFS1[1]: 1
	PB2	Reserved		AFS1[2]: 0
		ANA2/AMPINP	ADC Analog Input/LPO Input (P)	AFS1[2]: 1
	PB3	CLKIN	External Clock Input	AFS1[3]: 0
		ANA3	ADC Analog Input	AFS1[3]: 1
Port C	PC0	Reserved		AFS1[0]: 0
		ANA4/C0INP/LED	ADC or Comparator 0 Input (P), or LED drive	AFS1[0]: 1
	PC1	Reserved		AFS1[1]: 0
		ANA5/C0INN/LED	ADC or Comparator 0 Input (N), or LED drive	AFS1[1]: 1
	PC2	Reserved		AFS1[2]: 0
		VREF/ANA6/LED	Voltage Reference or ADC Analog Input or LED Drive	AFS1[2]: 1
	PC3	C0OUT	Comparator 0 Output	AFS1[3]: 0
		LED	LED drive	AFS1[3]: 1
Port D	PD0	$\overline{\text{RESET}}$	External Reset	N/A

Notes:
1. Because there are at most two choices of alternate functions for some pins in Port A, the Alternate Function Set Register (AFS2) is implemented but not used to select the function. The alternate function selection must also be enabled, as described in the Port A–E Alternate Function Subregisters section on page 61.
2. Because there is only one alternate function for each Port D pin, the Alternate Function Set registers are not implemented for Port D. Enabling the alternate function selections automatically enables the associated alternate function, as described in the Port A–E Alternate Function Subregisters section on page 61.

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

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 COUNTER Mode. After the first timer reload in COUNTER Mode, counting always begins at the reset value of 0001H. Generally, in COUNTER Mode the Timer High and Low Byte registers must be written with the value 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.
7. Configure the associated GPIO port pin for the Timer Input alternate function.
8. When using the Timer Output function, configure the associated GPIO port pin for the Timer Output alternate function.
9. Write to the Timer Control 1 Register to enable the timer.

In COUNTER Mode, the number of Timer Input transitions since the timer start is calculated using the following equation:

COUNTER Mode Timer Input Transitions = Current Count Value - Start Value

9.2.3.5. COMPARATOR COUNTER Mode

In COMPARATOR COUNTER Mode, the timer counts output transitions from an analog comparator output. The assignment of a comparator to a timer is based on the TIMTRG bits in the CMP0 and CMP1 registers. The TPOL bit in the Timer Control 1 Register selects whether the count occurs on the rising edge or the falling edge of the comparator output signal. In COMPARATOR COUNTER Mode, the prescaler is disabled.

! Caution: The frequency of the comparator output signal must not exceed one-fourth the timer clock frequency.

Upon reaching the reload value stored in the Timer Reload High and Low Byte registers, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes. Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state (from Low to High or High to Low) at timer reload.

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

10.5. Low-Power Modes

The Z8 Encore! XP F1680 Series of MCUs contains power-saving features. The highest level of power reduction is provided by STOP Mode. The next level of power reduction is provided by HALT Mode.

10.5.1. Operation in HALT Mode

When the eZ8 CPU is operating in HALT Mode, the Multi-Channel Timer will continue to operate if enabled. To minimize current in HALT Mode, the Multi-Channel Timer must be disabled by clearing the TEN control bit.

10.5.2. Operation in STOP Mode

When the eZ8 CPU is operating in STOP Mode, the Multi-Channel Timer ceases to operate because the system clock has stopped. The registers are not reset and operation will resume after Stop Mode Recovery occurs.

10.5.3. Power Reduction During Operation

Deassertion of the TEN bit will inhibit clocking of the entire Multi-Channel Timer block. Deassertion of the CHEN bit of individual channels will inhibit clocking of channel-specific logic to minimize power consumption of unused channels. The CPU can still read and write to the registers when the enable bit(s) are deasserted.

10.6. Multi-Channel Timer Applications Examples

This section provides two brief examples that describe how the the F1680 Series multi-channel timer can be used in your application.

10.6.1. PWM Programmable Deadband Generation

The count up/down mode supports motor control applications that require dead time between output signals. Figure 17 displays dead time generation between two channels operating in count up/down mode.

10.7.8. Multi-Channel Timer Channel-y Control Registers

Each channel has a control register to enable the channel, select the input/output polarity, enable channel interrupts and select the channel mode of operation.

Table 78. Multi-Channel Timer Channel Control Register (MCTCHyCTL)[1]

Bit	7	6	5	4	3	2	1	0
Field	CHEN	CHPOL	CHIEN	CHUE	Reserved	CHOP		
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
Address	See note 2.							

Notes:
1. y = A, B, C, D.
2. If 02H, 03H, 04H and 05H are in the Subaddress Register, they are accessible through Subregister 2.

Bit	Description
[7] CHEN	**Channel Enable** 0 = Channel is disabled. 1 = Channel is enabled.
[6] CHPOL	**Channel Input/Output Polarity** Operation of this bit is a function of the current operating method of the channel. **ONE-SHOT Operation** When the channel is disabled, the Channel Output signal is set to the value of this bit. When the channel is enabled, the Channel Output signal toggles for one system clock on reaching the Channel Capture/Compare Register value. **CONTINUOUS COMPARE Operation** When the channel is disabled, the Channel Output signal is set to the value of this bit. When the channel is enabled, the Channel Output signal toggles (from Low to High or High to Low) on reaching the Channel Capture/Compare Register value. **PWM OUTPUT Operation** 0 = Channel Output is forced Low when the channel is disabled. When enabled, the Channel Output is forced High on Channel Capture/Compare Register value match and forced Low on reaching the Timer Reload Register value (modulo mode) or counting down through the channel Capture/Compare register value (count up/down mode). 1 = Channel Output is forced Low when the channel is disabled. When enabled, the Channel Output is forced High on Channel Capture/Compare Register value match and forced Low on reaching the Timer Reload Register value (modulo mode) or counting down through the channel Capture/Compare register value (count up/down mode). **CAPTURE Operation** 0 = Count is captured on the rising edge of the Channel Input signal. 1 = Count is captured on the falling edge of the Channel Input signal.

8. Execute an EI instruction to enable interrupts.

The LIN-UART is now configured for interrupt-driven data transmission. Because the LIN-UART Transmit Data Register is empty, an interrupt is generated immediately. When the LIN-UART Transmit interrupt is detected and there is transmit data ready to send, the associated interrupt service routine (ISR) performs the following:

1. If in MULTIPROCESSOR Mode, writes to the LIN-UART Control 1 Register to select the outgoing address bit:
   - Sets the Multiprocessor Bit Transmitter (MPBT) if sending an address byte, clears it if sending a data byte.
2. Writes the data byte to the LIN-UART Transmit Data Register. The transmitter automatically transfers the data to the Transmit Shift Register and transmits the data.
3. Executes the IRET instruction to return from the interrupt-service routine and wait for the Transmit Data Register to again become empty.

If a transmit interrupt occurs and there is no transmit data ready to send, the interrupt service routine executes the IRET instruction. When the application does have data to transmit, software can set the appropriate interrupt request bit in the Interrupt Controller to initiate a new transmit interrupt. Another alternative would be for the software to write the data to the Transmit Data Register instead of invoking the interrupt service routine.

12.1.4. Receiving Data Using Polled Method

Observe the following steps to configure the LIN-UART for polled data reception:

1. Write to the LIN-UART Baud Rate High and Low Byte registers to set the appropriate baud rate.
2. Enable the LIN-UART pin functions by configuring the associated GPIO port pins for alternate function operation.
3. If MULTIPROCESSOR Mode is appropriate, write to the LIN-UART Control 1 Register to enable MULTIPROCESSOR (9-bit) Mode functions.
4. Write to the LIN-UART Control 0 Register to:
   a. Set the Receive Enable bit (REN) to enable the LIN-UART for data reception.
   b. If MULTIPROCESSOR Mode is not enabled, then enable parity (if appropriate), and select either even or odd parity.
5. Check the RDA bit in the LIN-UART Status 0 Register to determine if the Receive Data Register contains a valid data byte (indicated by a 1). If RDA is set to 1 to indicate available data, continue to Step 6. If the Receive Data Register is empty (indicated by a 0), continue to monitor the RDA bit that is awaiting reception of the valid data.

Note: In MULTIPROCESSOR Mode (MPEN=1), the receive-data interrupts are dependent on the multiprocessor configuration and the most recent address byte.

- A break is received
- A Receive Data Overrun or LIN Slave Autobaud Overrun Error is detected
- A Data Framing Error is detected
- A Parity Error is detected (physical layer error in LIN mode)

12.1.11.3. LIN-UART Overrun Errors

When an overrun error condition occurs, the LIN-UART prevents overwriting of the valid data currently in the Receive Data Register. The Break Detect and Overrun status bits are not displayed until after the valid data has been read.

After the valid data has been read, the OE bit of the Status 0 register is updated to indicate the overrun condition (and Break Detect, if applicable). The RDA bit is set to 1 to indicate that the Receive Data Register contains a data byte. However, because the overrun error occurred, this byte cannot contain valid data and must be ignored. The BRKD bit indicates if the overrun is caused by a break condition on the line. After reading the status byte indicating an overrun error, the Receive Data Register must be read again to clear the error bits in the LIN-UART Status 0 Register.

In LIN mode, an Overrun Error is signalled for receive-data overruns as described above and in the LIN Slave if the BRG Counter overflows during the autobaud sequence (the ATB bit will also be set in this case). There is no data associated with the autobaud overflow interrupt; however the Receive Data Register must be read to clear the OE bit. In this case, software must write a 10B to the LinState field, forcing the LIN slave back to a Wait for Break state.

12.1.11.4. LIN-UART Data- and Error-Handling Procedure

Figures 24 displays the recommended procedure for use in LIN-UART receiver interrupt service routines.

12.3.5. LIN-UART Control 0 Register

The LIN-UART Control 0 Register, shown in Table 89, configures the basic properties of LIN-UART's transmit and receive operations. A more detailed discussion of each bit follows the table.

Table 89. LIN-UART Control 0 Register (U0CTL0 = F42H)

Bit	7	6	5	4	3	2	1	0
Field	TEN	REN	CTSE	PEN	PSEL	SBRK	STOP	LBEN
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	F42H, F4AH							

Note: R/W = Read/Write.

Bit	Description
[7] TEN	**Transmit Enable** This bit enables or disables the transmitter. The enable is also controlled by the $\overline{CTS}$ signal and the CTSE bit. If the $\overline{CTS}$ signal is Low and the CTSE bit is 1, the transmitter is enabled. 0 = Transmitter disabled. 1 = Transmitter enabled.
[6] REN	**Receive Enable** This bit enables or disables the receiver. 0 = Receiver disabled. 1 = Receiver enabled.
[5] CTSE	**Clear To Send Enable** 0 = The $\overline{CTS}$ signal has no effect on the transmitter. 1 = The LIN-UART recognizes the $\overline{CTS}$ signal as an enable control for the transmitter.
[4] PEN	**Parity Enable** This bit enables or disables parity. Even or odd is determined by the PSEL bit. 0 = Parity is disabled. This bit is overridden by the MPEN bit. 1 = The transmitter sends data with an additional parity bit and the receiver receives an additional parity bit.
[3] PSEL	**Parity Select** 0 = Even parity is sent as an additional parity bit for the transmitter/receiver. 1 = Odd parity is sent as an additional parity bit for the transmitter/receiver.

I2CISTAT Register is set to 1, thereby causing an interrupt. The RD bit is cleared to 0, indicating a Write to the slave. The I^2C controller acknowledges, indicating it is available to accept the data.

4. The software responds to the interrupt by reading the I2CISTAT Register, which clears the SAM bit. Because RD = 0, no immediate action is taken by the software until the first byte of data is received. If the software is only able to accept a single byte, it sets the NAK bit in the I2CCTL Register.
5. The Master detects the Acknowledge and sends the first byte of data.
6. The I^2C controller receives the first 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.
7. The software responds by reading the I2CISTAT Register, finding the RDRF bit = 1 and then reading the I2CDATA Register, which clears the RDRF bit. If the software can accept only one more data byte, it sets the NAK bit in the I2CCTL Register.
8. The Master and Slave loops through Step 5 to Step 7 until the Master detects a Not Acknowledge instruction or runs out of data to send.
9. The Master sends the stop or restart signal on the bus. Either of these signals can cause the I^2C controller to assert the 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.

17.2.6.7. Slave Transmit Transaction With 7-bit Address

The data transfer format for a master reading data from a slave in 7-bit address mode is displayed in Figure 49. The procedure that follows describes the I^2C Master/Slave Controller operating as a slave in 7-bit addressing mode and transmitting data to the bus master.

S	Slave Address	R = 1	A	Data	A	Data	A	P/S

Figure 49. Data Transfer Format—Slave Transmit Transaction with 7-bit Address

1. The software configures the controller for operation as a slave in 7-bit addressing mode, as follows:
   a. Initialize the MODE field in the I^2C Mode Register for either SLAVE ONLY mode or MASTER/SLAVE Mode with 7-bit addressing.
   b. Optionally set the GCE bit.
   c. Initialize the SLA[6:0] bits in the I^2C Slave Address Register.

Table 125. I²C State Register (I2CSTATE)—Description when DIAG = 0

Bits	7	6	5	4	3	2	1	0
Field	ACKV	ACK	AS	DS	10B	RSTR	SCLOUT	BUSY
Reset	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R
Address	F55H							

Bit	Description
[7] ACKV	**ACK Valid** This bit is set, if sending data (Master or Slave) and the ACK bit in this register is valid for the byte just transmitted. This bit can be monitored if it is appropriate for software to verify the ACK value before writing the next byte to be sent. To operate in this mode, the Data Register must not be written when TDRE asserts; instead, the software waits for ACKV to assert. This bit clears when transmission of the next byte begins or the transaction is ended by a stop or restart condition.
[6] ACK	**Acknowledge** This bit indicates the status of the Acknowledge for the last byte transmitted or received. This bit is set for an Acknowledge and cleared for a Not Acknowledge condition.
[5] AS	**Address State** This bit is active High while the address is being transferred on the I²C bus.
[4] DS	**Data State** This bit is active High while the data is being transferred on the I²C bus.
[3] 10B	**10B** This bit indicates whether a 7-bit or 10-bit address is being transmitted when operating as a Master. After the start bit is set, if the five most-significant bits of the address are 11110B, this bit is set. When set, it is Reset after the address has been sent.
[2] RSTR	**RESTART** This bit is updated each time a stop or restart interrupt occurs (SPRS bit set in I2CISTAT Register). 0 = Stop condition. 1 = Restart condition.
[1] SCLOUT	**Serial Clock Output** Current value of Serial Clock being output onto the bus. The actual values of the SCL and SDA signals on the I²C bus can be observed via the GPIO Input Register.
[0] BUSY	**I²C Bus Busy** 0 = No activity on the I²C Bus. 1 = A transaction is underway on the I²C bus.

Table 127. I2CSTATE_L (Continued)

State I2CSTATE_H	Substate I2CSTATE_L	Substate Name	State Description
1000–1111	0111	Send/Receive bit 7	Sending/Receiving most significant bit.
	0110	Send/Receive bit 6	
	0101	Send/Receive bit 5	
	0100	Send/Receive bit 4	
	0011	Send/Receive bit 3	
	0010	Send/Receive bit 2	
	0001	Send/Receive bit 1	
	0000	Send/Receive bit 0	Sending/Receiving least significant bit.
	1000	Send/Receive Acknowledge	Sending/Receiving Acknowledge.

17.3.6. I²C Mode Register

The I²C Mode Register, shown in Table 128, provides control over master versus slave operating mode, slave address and diagnostic modes.

Table 128. I²C Mode Register (I2C Mode = F56H)

Bits	7	6	5	4	3	2	1	0
Field	Reserved	MODE[1:0]		IRM	GCE	SLA[9:8]		DIAG
Reset	0	0		0	0	0		0
R/W	R	R/W		R/W	R/W	R/W		R/W
Address	F56H							

Bit	Description
[7]	Reserved; must be 0.
[6:5] MODE[1:0]	**Selects the I²C Controller Operational Mode** 00 = MASTER/SLAVE capable (supports multi-master arbitration) with 7-bit Slave address. 01 = MASTER/SLAVE capable (supports multi-master arbitration) with 10-bit slave address. 10 = Slave Only capable with 7-bit address. 11 = Slave Only capable with 10-bit address.

```
DBG ← Size[7:0]
DBG → 1-256 data bytes
```

Write Program Memory (0AH). The Write Program Memory command writes data to Program memory. This command is equivalent to the LDC and LDCI instructions. Data can be written 1–65536 bytes at a time (65536 bytes can be written by setting size to 0). The on-chip Flash Controller must be written and unlocked for the programming operation to occur. If the Flash Controller is not unlocked, the data is discarded. If the device is not in DEBUG mode or if the Read Protect option bit is enabled, the data is discarded.

```
DBG ← 0AH
DBG ← Program Memory Address[15:8]
DBG ← Program Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG ← 1-65536 data bytes
```

Read Program Memory (0BH). The Read Program Memory command reads data from Program memory. This command is equivalent to the LDC and LDCI instructions. Data can be read 1–65536 bytes at a time (65536 bytes can be read by setting size to 0). If the device is not in DEBUG mode or if the Read Protect option bit is enabled, this command returns FFH for the data.

```
DBG ← 0BH
DBG ← Program Memory Address[15:8]
DBG ← Program Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG → 1-65536 data bytes
```

Write Data Memory (0CH). The Write Data Memory command writes data to Data Memory. This command is equivalent to the LDE and LDEI instructions. Data is written 1–65536 bytes at a time (65536 bytes can be written by setting size to 0). If the device is not in DEBUG mode or if the Read Protect option bit is enabled, the data is discarded.

```
DBG ← 0CH
DBG ← Data Memory Address[15:8]
DBG ← Data Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG ← 1-65536 data bytes
```

Read Data Memory (0DH). The Read Data Memory command reads from Data Memory. This command is equivalent to the LDE and LDEI instructions. Data can be read 1 to 65536 bytes at a time (65536 bytes can be read by setting size to 0). If the device is not in DEBUG mode, this command returns FFH for the data.

```
DBG ← 0DH
DBG ← Data Memory Address[15:8]
DBG ← Data Memory Address[7:0]
DBG ← Size[15:8]
```

23.4.4. Baud Reload Register

The Baud Reload Register, shown in Table 167, contains the measured Autobaud value.

Table 167. Baud Reload Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved				RELOAD											
Reset	0H				000H											
R/W	R				R											

Bit	Description
[15:12]	Reserved; must be 000.
[11:0] RELOAD	**Baud Reload Value** This value is the measured Autobaud value. Its value can be calculated using the formula: $RELOAD = \frac{SYSCLK}{BAUDRATE} \times 8$

Table 189. DC Characteristics (Continued)

		T_A = 0°C to +70°C T_A = –40°C to +105°C				
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
V_{IH2}	High Level Input Voltage	0.7*V_{DD}	–	V_{DD}+0.3	V	Ports B and C (Analog)
V_{OL1}	Low Level Output Voltage	–	–	0.4	V	I_{OL} = 2 mA; V_{DD} = 3.0V High Output Drive disabled.
V_{OH1}	High Level Output Voltage	V_{DD}–0.5	–	–	V	I_{OH} = –2 mA; V_{DD} = 3.0V High Output Drive disabled.
V_{OL2}	Low Level Output Voltage	–	–	0.6	V	I_{OL} = 20 mA; V_{DD} = 3.3V High Output Drive enabled.
V_{OH2}	High Level Output Voltage	V_{DD}–0.5	–	–	V	I_{OH} = –20 mA; V_{DD} = 3.3V High Output Drive enabled.
I_{IL}	Input Leakage Current	–5	–	+5	µA	V_{DD} = 3.6V; V_{IN} = V_{DD} or V_{SS}[1]
I_{TL}	Tristate Leakage Current	–5	–	+5	µA	V_{DD} = 3.6V
I_{LED}	Controlled LED Current Drive	1.5	3	4.5	mA	{AFS2,AFS1} = {0,0}, V_{DD} = 3.3V
		2.8	7	10.5	mA	{AFS2,AFS1} = {0,1}, V_{DD} = 3.3V
		7.8	13	19.5	mA	{AFS2,AFS1} = {1,0}, V_{DD} = 3.3V
		12	20	30	mA	{AFS2,AFS1} = {1,1}, V_{DD} = 3.3V
C_{PAD}	GPIO Port Pad Capacitance	–	8.0[2]	–	pF	TBD
C_{XIN}	XIN Pad Capacitance	–	8.0[2]	–	pF	TBD
C_{XOUT}	XOUT Pad Capacitance	–	9.5[2]	–	pF	TBD
I_{PU}	Weak Pull-up Current	30	100	350	µA	V_{DD} = 3.0V–3.6V

Notes:
1. This condition excludes all pins that have on-chip pull-ups, when driven Low.
2. These values are provided for design guidance only and are not tested in production.

Chapter 30. Packaging

Zilog's F1680 Series of MCUs includes the Z8F0880, Z8F1680 and Z8F2480 devices, which are available in the following packages:

- 20-pin Plastic Dual-Inline Package (PDIP)
- 20-pin Small Outline Integrated Circuit Package (SOIC)
- 20-pin Small Shrink Outline Package (SSOP)
- 28-pin Plastic Dual-Inline Package (PDIP)
- 28-pin Small Outline Integrated Circuit Package (SOIC)
- 28-pin Small Shrink Outline Package (SSOP)
- 40-pin Plastic Dual-Inline Package (PDIP)
- 44-pin Low-Profile Quad Flat Package (LQFP)
- 44-pin Quad Flat No Lead (QFN)

Current diagrams for each of these packages are published in Zilog's Packaging Product Specification (PS0072), which is available free for download from the Zilog website.

H

I