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	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 7x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f0880hh020eg

Table of Contents

1.4. An Overview of the eZ8 CPU and its Peripherals

Zilog's eZ8 CPU, latest 8-bit CPU meets the continuing demand for faster and more code-efficient microcontrollers. It executes a superset of the original Z8® instruction set. The eZ8 CPU features include:

- Direct register-to-register architecture allows each register to function as an accumulator, improving execution time and decreasing the required program memory
- Software stack allows greater depth in subroutine calls and interrupts more than hardware stacks
- Compatible with existing Z8 code
- Expanded internal Register File allows access up to 4KB
- New instructions improve execution efficiency for code developed using higher-level programming languages including C
- Pipelined instruction fetch and execution
- New instructions for improved performance including BIT, BSWAP, BTJ, CPC, LDC, LDCI, LEA, MULT and SRL
- New instructions support 12-bit linear addressing of the register file
- Up to 10 MIPS operation
- C-Compiler friendly
- 2 to 9 clock cycles per instruction

For more details about eZ8 CPU, refer to the eZ8 CPU Core User Manual (UM0128), available for download at www.zilog.com.

1.4.1. General-Purpose Input/Output

The F1680 MCU features 17 to 37 port pins (Ports A–E) for general purpose input/output (GPIO) pins. The number of GPIO pins available is a function of package. Each pin is individually programmable.

1.4.2. Flash Controller

The Flash Controller is used to program and erase Flash memory. The Flash Controller supports protection against accidental program and erasure.

Table 9. Reset and Stop Mode Recovery Characteristics and Latency

	Reset Characteristics and Latency		
Reset Type	**Control Registers**	**eZ8 CPU**	**Reset Latency (Delay)**
System Reset (non-POR Reset)	Reset (as applicable)	Reset	68 Internal Precision Oscillator Cycles after IPO starts up
System Reset (POR Reset)	Reset (as applicable)	Reset	68 Internal Precision Oscillator Cycles + 50 ms Wait time
System Reset with Crystal Oscillator Enabled	Reset (as applicable)	Reset	568–10068 Internal Precision Oscillator Cycles after IPO starts up; see Table 141 on page 280 for a description of the EXTLTMG user option bit.
Stop Mode Recovery	Unaffected, except RSTSTAT and OSCCTL registers	Reset	4 Internal Precision Oscillator Cycles after IPO starts up

During a System Reset or Stop Mode Recovery, the Internal Precision Oscillator (IPO) requires 4 µs to start up. When the reset type is a System Reset, the F1680 Series MCU is held in Reset for 68 IPO cycles. If the crystal oscillator is enabled in Flash option bits, the Reset period is increased to 568–10068 IPO cycles. For more details, see Table 141 on page 280 for a description of the EXTLTMG user option bit. When the reset type is a Stop Mode Recovery, the F1680 Series MCU goes to NORMAL Mode immediately after 4 IPO cycles. The total Stop Mode Recovey delay is less than 6 µs. When a Reset occurs due to a VBO condition, this delay is measured from the time the supply voltage first exceeds the VBO level (discussed later in this chapter). When a Reset occurs due to a POR condition, this delay is measured from the time that the supply voltage first exceeds the POR level. If the external pin reset remains asserted at the end of the Reset period, the device remains in reset until the pin is deasserted.

Note: After a Stop Mode Recovery, the external crystal oscillator is unstable. Use software to wait until it is stable before you can use it as main clock.

At the beginning of Reset, all GPIO pins are configured as inputs with pull-up resistor disabled, except PD0 that is shared with the Reset pin. On Reset, the Port D0 pin is configured as a bidirectional open-drain Reset. The pin is internally driven Low during port reset, after which the user code can reconfigure this pin as a general-purpose output.

During Reset, the eZ8 CPU and on-chip peripherals are idle; however, the on-chip crystal oscillator and WDT oscillator continue to function.

eZ8 CPU services the Timer interrupt request following the normal Stop Mode Recovery sequence.

5.3.3. Stop Mode Recovery Using Comparator Interrupt

If Comparator enabled for STOP Mode operation interrupts during STOP Mode, the device undergoes a Stop Mode Recovery sequence. In the Reset Status Register, the stop bit is set to 1. If the F1680 Series MCU is configured to respond to interrupts, the eZ8 CPU services the comparator interrupt request following the normal Stop Mode Recovery sequence.

5.3.4. Stop Mode Recovery Using GPIO Port Pin Transition

Each of the GPIO port pins can be configured as a Stop Mode Recovery input source. On any GPIO pin enabled as a Stop Mode Recovery source, a change in the input pin value (from High to Low or from Low to High) initiates Stop Mode Recovery. In the Reset Status Register, the stop bit is set to 1.

! **Caution:** In STOP Mode, the GPIO Port Input Data registers (PxIN) are disabled. The Port Input Data registers record the Port transition only if the signal stays on the Port pin until the end of the Stop Mode Recovery delay. As a result, short pulses on the Port pin can initiate Stop Mode Recovery without being written to the Port Input Data Register or without initiating an interrupt (if enabled for that pin).

5.3.5. Stop Mode Recovery Using External $\overline{\text{RESET}}$ Pin

When the F1680 Series MCU is in STOP Mode and the external $\overline{\text{RESET}}$ pin is driven Low, a System Reset occurs. Because of a glitch filter operating on the $\overline{\text{RESET}}$ pin, the Low pulse must be greater than the minimum width specified, or it is ignored. For details, see the Electrical Characteristics chapter on page 349.

5.4. Low-Voltage Detection

In addition to the VBO Reset described earlier, it is also possible to generate an interrupt when the supply voltage drops below a user-selected value. For more details about the available Low-Voltage Detection (LVD) threshold levels, see the Trim Option Bits at Address 0000H (TTEMP0) section on page 282.

When the supply voltage drops below the LVD threshold, the LVD bit of the RSTSTAT Register is set to 1. This bit remains 1 until the low-voltage condition elapses. Reading or

6.4. Power Control Register Definitions

The following sections describe the power control registers.

6.4.0.1. Power Control Register 0

Each bit of the following registers disables a peripheral block, either by gating its system clock input or by removing power from the block.

The default state of the low-power operational amplifier is OFF. To use the low-power operational amplifier, clear the TRAM bit by turning it ON. Clearing this bit might interfere with normal ADC measurements on ANA0 (the LPO output). This bit enables the amplifier even in STOP Mode. If the amplifier is not required in STOP Mode, disable it. Failure to perform this results in STOP Mode currents greater than specified.

Note: This register is only reset during a POR sequence; other system reset events do not affect it.

Table 14. Power Control Register 0 (PWRCTL0)

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	TRAM	Reserved		LVD/VBO	TEMP	Reserved	COMP0	COMP1
Reset	1	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	F80H							

Bit	Description
[7] TRAM	**Low-Power Operational Amplifier Disable** 0 = Low-Power Operational Amplifier is enabled (this applies even in STOP Mode). 1 = Low-Power Operational Amplifier is disabled.
[6:5]	Reserved; must be 0.
[4] LVD/VBO	**Low-Voltage Detection/Voltage Brown-Out Detector Disable** 0 = LVD/VBO Enabled. 1 = LVD/VBO Disabled. The LVD and VBO circuits are enabled or disabled separately to minimize power consumption in low-power modes. The LVD is controlled by the LVD/VBO bit only in all modes. The VBO is set by the LVD/VBO bit and the VBO_AO bit of Flash Option bit at Program Memory Address 0000H. Table 15 on page 45 lists the setup condition for LVD and VBO circuits in different operation modes.
[3] TEMP	**Temperature Sensor Disable** 0 = Temperature Sensor Enabled. 1 = Temperature Sensor Disabled.

7.4. Direct LED Drive

The Port C pins provide a current synchronized output capable of driving an LED without requiring an external resistor. The output synchronizes current at programmable levels of 3mA, 7mA, 13mA and 20mA. This mode is enabled through the Alternate Function sub-register AFS1 and is programmable through the LED control registers. For proper function, the LED anode must be connected to V_{DD} and the cathode to the GPIO pin.

Using all Port C pins in LED drive mode with maximum current can result in excessive total current. For the maximum total current for the applicable package, see the Electrical Characteristics chapter on page 349.

7.5. Shared Reset Pin

On all the devices, the Port D0 pin shares function with a bidirectional reset pin. Unlike all other I/O pins, this pin does not default to GPIO pin on power-up. This pin acts as a bidirectional input/open-drain output reset with an internal pull-up until user software reconfigures it as GPIO PD0. The Port D0 pin is output-only when in GPIO Mode, and must be configured as an output. PD0 supports the High Drive feature but not the Stop Mode Recovery feature.

7.6. Crystal Oscillator Override

For systems using the crystal oscillator, PA0 and PA1 is used to connect the crystal. When the main crystal oscillator is enabled (see the Oscillator Control1 Register section on page 320), the GPIO settings are overridden and PA0 and PA1 is disabled.

7.7. 32kHz Secondary Oscillator Override

For systems using a 32kHz secondary oscillator, PA2 and PA3 is used to connect a watch crystal. When the 32kHz secondary oscillator is enabled (see the Oscillator Control1 Register section on page 320), the GPIO settings are overridden and PA2 and PA3 is disabled.

7.8. 5V Tolerance

All GPIO pins, including those that share functionality with an ADC, crystal or comparator signals are 5V-tolerant and can handle inputs higher than V_{DD} even with the pull-ups enabled.

Table 19. Port Alternate Function Mapping, 40-/44-Pin Parts[1,2]

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Register AFS1
Port A	PA0	T0IN/$\overline{\text{T0OUT}}$	Timer 0 Input/Timer 0 Output Complement	AFS1[0]: 0
		Reserved		AFS1[0]: 1
	PA1	T0OUT	Timer 0 Output	AFS1[1]: 0
		Reserved		AFS1[1]: 1
	PA2	DE0	UART 0 Driver Enable	AFS1[2]: 0
		Reserved		AFS1[2]: 1
	PA3	$\overline{\text{CTS0}}$	UART 0 Clear to Send	AFS1[3]: 0
		Reserved		AFS1[3]: 1
	PA4	RXD0/IRRX0	UART 0/IrDA 0 Receive Data	AFS1[4]: 0
				AFS1[4]: 1
	PA5	TXD0/IRTX0	UART 0/IrDA 0 Transmit Data	AFS1[5]: 0
				AFS1[5]: 1
	PA6	T1IN/$\overline{\text{T1OUT}}$	Timer 1 Input/Timer 1 Output Complement	AFS1[6]: 0
		Reserved		AFS1[6]: 1
	PA7	T1OUT	Timer 1 Output	AFS1[7]: 0
		Reserved		AFS1[7]: 1

Notes:

1. Because there are at most two choices of alternate functions for some pins in Ports A–C, 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 and Port E pin, the Alternate Function Set registers are not implemented for Ports D and E. 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.
3. This timer function is only available in the 44-pin package; its alternate functions are reserved in the 40-pin package.

transitions from a Low to High. This configuration ensures a time gap between the removal of one PWM output and the assertion of its complement.

Observe the following steps to configure a timer for PWM DUAL OUTPUT Mode and initiate the PWM operation:

1. Write to the Timer Control 1 Register to:
   - Disable the timer
   - Configure the timer for PWM DUAL OUTPUT Mode. Setting the mode also involves writing to TMODE[3] bit in the TxCTL0 Register
   - Set the prescale value
   - Set the initial logic level (High or Low) and PWM High/Low transition for the Timer Output Alternate Function
2. Write to the Timer High and Low Byte registers to set the starting count value (typically `0001H`). This value only affects the first pass in PWM mode. After the first timer reset in PWM mode, counting always begins at the reset value of `0001H`.
3. Write to the Timer PWM0 High and Low Byte registers to set the PWM value.
4. Write to the Timer Control 0 Register:
   - To set the PWM deadband delay value
   - To choose the timer clock source
5. Write to the Timer Control 0 Register to set the timer interrupt configuration field TICONFIG.
6. Write to the Timer Reload High and Low Byte registers to set the reload value (PWM period). The reload value must be greater than the PWM value.
7. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
8. Configure the associated GPIO port pin for the Timer Output and Timer Output Complement alternate functions.
9. Write to the Timer Control 1 Register to enable the timer and initiate counting.

The PWM period is calculated using the following equation:

If an initial starting value other than `0001H` is loaded into the Timer High and Low Byte registers, the ONE-SHOT Mode equation must be used to determine the first PWM time-out period.

If TPOL is set to 0, the ratio of the PWM output High time to the total period is calculated using the following equation:

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

1. Write to the Timer Control 1 Register to:
   - Disable the timer
   - Configure the timer for CAPTURE RESTART Mode. Setting the mode also involves writing to TMODE[3] bit in the TxCTL0 Register
   - Set the prescale value
   - Set the Capture edge (rising or falling) for the Timer Input
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 (typically `0001H`).
5. Write to the Timer Reload High and Low Byte registers to set the reload value.
6. Clear the Timer PWM High and Low Byte registers to `0000H`. This allows user software to determine if interrupts are generated by either a Capture Event or a Reload. If the PWM High and Low Byte registers still contain `0000H` after the interrupt, then the interrupt is generated by a Reload.
7. 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.
8. Configure the associated GPIO port pin for the Timer Input alternate function.
9. Write to the Timer Control 1 Register to enable the timer and initiate counting.

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

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

9.3.7. Timer 0–2 Noise Filter Control Register

The Timer 0–2 Noise Filter Control Register (TxNFC) enables and disables the Timer Noise Filter and sets the noise filter control.

Table 67. Timer 0–2 Noise Filter Control Register (TxNFC)

Bit	7	6	5	4	3	2	1	0
Field	NFEN	NFCTL			Reserved			
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W			R			
Address	F2CH, F2DH, F2EH							

Bit	Description
[7] NFEN	**Noise Filter Enable** 0 = Noise Filter is disabled. 1 = Noise Filter is enabled. Receive data is preprocessed by the noise filter.
[6:4] NFCTL	**Noise Filter Control** This field controls the delay and noise rejection characteristics of the Noise Filter. The wider the counter the more delay that is introduced by the filter and the wider the noise event that will be filtered. 000 = 2-bit up/down counter 001 = 3-bit up/down counter 010 = 4-bit up/down counter 011 = 5-bit up/down counter 100 = 6-bit up/down counter 101 = 7-bit up/down counter 110 = 8-bit up/down counter 111 = 9-bit up/down counter
[3:0]	Reserved; must be 0.

12.3.2. LIN-UART Receive Data Register

Data bytes received through the RxD pin are stored in the LIN-UART Receive Data Register as shown in Table 84. The read-only LIN-UART Receive Data Register shares a Register File address with the write-only LIN-UART Transmit Data Register.

Table 84. LIN-UART Receive Data Register (U0RXD = F40H)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	RxD							
Reset	X	X	X	X	X	X	X	X
R/W	R	R	R	R	R	R	R	R
Address	F40H, F48H							

Note: R = Read.

Bit	Description
[7:0] RxD	**Receive Data** LIN–UART receiver data byte from the RxD pin.

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

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

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

14.3.5. Sample Settling Time Register

The Sample Settling Time Register, shown in Table 105, is used to program the length of time from the SAMPLE/$\overline{\text{HOLD}}$ signal to the start signal, when the conversion can begin. The number of clock cycles required for settling will vary from system to system depending on the system clock period used. The system designer should program this register to contain the number of clocks required to meet a 0.5 µs minimum settling time.

Table 105. Sample Settling Time (ADCSST)

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved				SST			
Reset	0				1	1	1	1
R/W	R				R/W			
Address	F74H							

Bit Position	**Value (H)**	**Description**
[7:4]	0	Reserved; must be 0.
[3:0] SST	0–F	Sample settling time in number of system clock periods to meet 0.5 µs minimum.

Table 109. ESPI Data Register (ESPIDATA)

Bits	7	6	5	4	3	2	1	0
Field	DATA							
Reset	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F60H							

Bit	Description
[7:0] DATA	**Data** Transmit and/or receive data. Writes to the ESPIDATA register load the Shift Register. Reads from the ESPIDATA register return the value of the Receive Data Register.

16.4.2. ESPI Transmit Data Command and Receive Data Buffer Control Register

The ESPI Transmit Data Command and Receive Data Buffer Control Register, shown in Table 110, provides control of the $\overline{SS}$ pin when it is configured as an output (MASTER Mode), clear receive data buffer function and flag. The CRDR, TEOF and SSV bits can be controlled by a bus write to this register.

Table 110. ESPI Transmit Data Command and Receive Data Buffer Control Register (ESPITDCR)

Bits	7	6	5	4	3	2	1	0
Field	CRDR	RDFLAG		Reserved			TEOF	SSV
Reset	0	00		0	0	0	0	0
R/W	R/W	R		R	R	R	R/W	R/W
Address	F61H							

Bit	Description
[7] CRDR	**Clear Receive Data Register** Writing 1 to this bit is used to clear all data in receive data buffer.
[6:5] RDFLAG	**Receive Data Buffer Flag** This bit is used to indicate how many bytes stored in receive buffer. 00 = 0 or 4 bytes (see RDRNE in the ESPI Status Register). 01 = 1 byte. 02 = 2 bytes. 03 = 3 bytes.
[4:2]	**Reserved** These bits are reserved and must be programmed to 000.

The first 7 bits transmitted in the first byte are `11110XX`. The 2 `XX` bits are the two most significant bits of the 10-bit address. The lowest bit of the first byte transferred is the Read/Write control bit (which is cleared to 0). The transmit operation is performed in the same manner as 7-bit addressing.

Observe the following steps for a master transmit operation to a 10-bit addressed slave:

1. The software initializes the MODE field in the I^{2}C Mode Register for MASTER/SLAVE Mode with 7- or 10-bit addressing (the I^{2}C bus protocol allows the mixing of slave address types). 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^{2}C Control Register.
2. The software asserts the TXI bit of the I^{2}C Control Register to enable transmit interrupts.
3. The I^{2}C interrupt asserts because the I^{2}C Data Register is empty.
4. The software responds to the TDRE interrupt by writing the first Slave Address byte (`11110xx0`). The least-significant bit must be 0 for the write operation.
5. The software asserts the start bit of the I^{2}C Control Register.
6. The I^{2}C controller sends a start condition to the I^{2}C Slave.
7. The I^{2}C controller loads the I^{2}C Shift Register with the contents of the I^{2}C Data Register.
8. After one bit of the address is shifted out by the SDA signal, the transmit interrupt asserts.
9. The software responds by writing the second byte of address into the contents of the I^{2}C Data Register.
10. The I^{2}C controller shifts the remainder of the first byte of the address and the Write bit out via the SDA signal.
11. The I^{2}C slave sends an Acknowledge by pulling the SDA signal Low during the next High period of SCL. The I^{2}C controller sets the ACK bit in the I^{2}C Status Register.

    If the slave does not acknowledge the first address byte, the I^{2}C controller sets the NCKI bit in the I^{2}C Status Register, sets the ACKV bit and clears the ACK bit in the I^{2}C State Register. The software responds to the Not Acknowledge interrupt by setting the stop bit and clearing the TXI bit. The I^{2}C controller flushes the second address byte from the 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^{2}C controller loads the I^{2}C Shift Register with the contents of the I^{2}C Data Register (2nd address byte).
13. The I^{2}C controller shifts the second address byte out via the SDA signal. After the first bit has been sent, the transmit interrupt asserts.

17.3.3. I^2C Control Register

The I^2C Control Register, shown in Table 121, enables and configures I^2C operation.

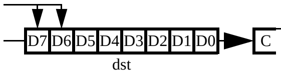
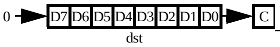
Note: The R/W1 bit can be set (written to 1) when IEN = 1, but cannot be cleared (written to 0).

Table 121. I^2C Control Register (I2CCTL)

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	IEN	START	STOP	BIRQ	TXI	NAK	FLUSH	FILTEN
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W1	R/W1	R/W	R/W	R/W1	W	R/W
Address	F52H							

Bit	Description
[7] IEN	**I^2C Enable** This bit enables the I^2C controller.
[6] START	**Send Start Condition** When set, this bit causes the I^2C controller (when configured as the master) to send a start condition. After it is asserted, this bit is cleared by the I^2C controller after it sends the start condition or by deasserting the IEN bit. If this bit is 1, it cannot be cleared by writing to the bit. After this bit is set, a start condition is sent if there is data in the I2CDATA or I^2C Shift Register. If there is no data in one of these registers, the I^2C controller waits until data is loaded. If this bit is set while the I^2C controller is shifting out data, it generates a restart condition after the byte shifts and the Acknowledge phase completes. If the stop bit is also set, it waits until the stop condition is sent before the start condition. If start is set while a SLAVE Mode transaction is underway to this device, the start bit will be cleared and ARBLST bit in the Interrupt Status Register will be set.
[5] STOP	**Send Stop Condition** When set, this bit causes the I^2C controller (when configured as the master) to send the stop condition after the byte in the I^2C Shift Register has completed transmission or after a byte is received in a receive operation. When set, this bit is reset by the I^2C controller after a stop condition has been sent or by deasserting the IEN bit. If this bit is 1, it cannot be cleared to 0 by writing to the register. If stop is set while a SLAVE Mode transaction is underway, the stop bit is cleared by hardware.
[4] BIRQ	**Baud Rate Generator Interrupt Request** This bit is ignored when the I^2C controller is enabled. If this bit is set = 1 when the I^2C controller is disabled (IEN = 0), the baud rate generator is used as an additional timer causing an interrupt to occur every time the baud rate generator counts down to one. The baud rate generator runs continuously in this mode, generating periodic interrupts.

Table 186. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Op Code(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
SCF	C ← 1			DF	1	–	–	–	–	–	1	2
SRA dst		R		D0	*	*	*	0	–	–	2	2
		IR		D1							2	3
SRL dst	 0 D7 D6 D5 D4 D3 D2 D1 D0 C dst	R		1F C0	*	*	0	*	–	–	3	2
		IR		1F C1							3	3
SRP src	RP ← src		IM	01	–	–	–	–	–	–	2	2
STOP	STOP Mode			6F	–	–	–	–	–	–	1	2
SUB dst, src	dst ← dst – src	r	r	22	*	*	*	*	1	*	2	3
		r	Ir	23							2	4
		R	R	24							3	3
		R	IR	25							3	4
		R	IM	26							3	3
		IR	IM	27							3	4
SUBX dst, src	dst ← dst – src	ER	ER	28	*	*	*	*	1	*	4	3
		ER	IM	29							4	3
SWAP dst	dst[7:4] ↔ dst[3:0]	R		F0	X	*	*	X	–	–	2	2
		IR		F1							2	3
TCM dst, src	(NOT dst) AND src	r	r	62	–	*	*	0	–	–	2	3
		r	Ir	63							2	4
		R	R	64							3	3
		R	IR	65							3	4
		R	IM	66							3	3
		IR	IM	67							3	4
TCMX dst, src	(NOT dst) AND src	ER	ER	68	–	*	*	0	–	–	4	3
		ER	IM	69							4	3

Flags notation:
* = Value is a function of the result of the operation.
– = Unaffected.
X = Undefined.
0 = Reset to 0.
1 = Set to 1.

Chapter 29. Electrical Characteristics

The data in this chapter is prequalification and precharacterization and is subject to change. Additional electrical characteristics can be found in the individual chapters.

29.1. Absolute Maximum Ratings

Stresses greater than those listed in Table 188 can cause permanent damage to the device. These ratings are stress ratings only. Operation of the device at any condition outside those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods can affect device reliability. For improved reliability, tie unused inputs to one of the supply voltages (V_{DD} or V_{SS}).

Table 188. Absolute Maximum Ratings*

Parameter	Min	Max	Units	Notes
Ambient temperature under bias	0	+105	°C	
Storage temperature	–65	+150	°C	
Voltage on any pin with respect to V_{SS}	–0.3	+5.5	V	1
Voltage on V_{DD} pin with respect to V_{SS}	–0.3	+3.6	V	
Maximum current on input and/or inactive output pin	–5	+5	µA	
Maximum output current from active output pin	–25	+25	mA	
20-pin Packages Maximum Ratings at 0°C to 70°C				
Total power dissipation		430	mW	
Maximum current into V_{DD} or out of V_{SS}		120	mA	
28-pin Packages Maximum Ratings at 0°C to 70°C				
Total power dissipation		450	mW	
Maximum current into V_{DD} or out of V_{SS}		125	mA	
40-pin PDIP Maximum Ratings at –40°C to 70°C				
Total power dissipation		1000	mW	
Maximum current into V_{DD} or out of V_{SS}		275	mA	
40-pin PDIP Maximum Ratings at 70°C to 105°C				
Total power dissipation		540	mW	
Maximum current into V_{DD} or out of V_{SS}		150	mA	

Notes:
*Operating temperature is specified in DC Characteristics.
1. This voltage applies to all pins except the following: V_{DD}, AV_{DD}.

29.4.1. General Purpose I/O Port Input Data Sample Timing

Figure 75 displays timing of the GPIO Port input sampling. The input value on a GPIO port pin is sampled on the rising edge of the system clock. The Port value is available to the eZ8 CPU on the second rising clock edge following the change of the Port value.

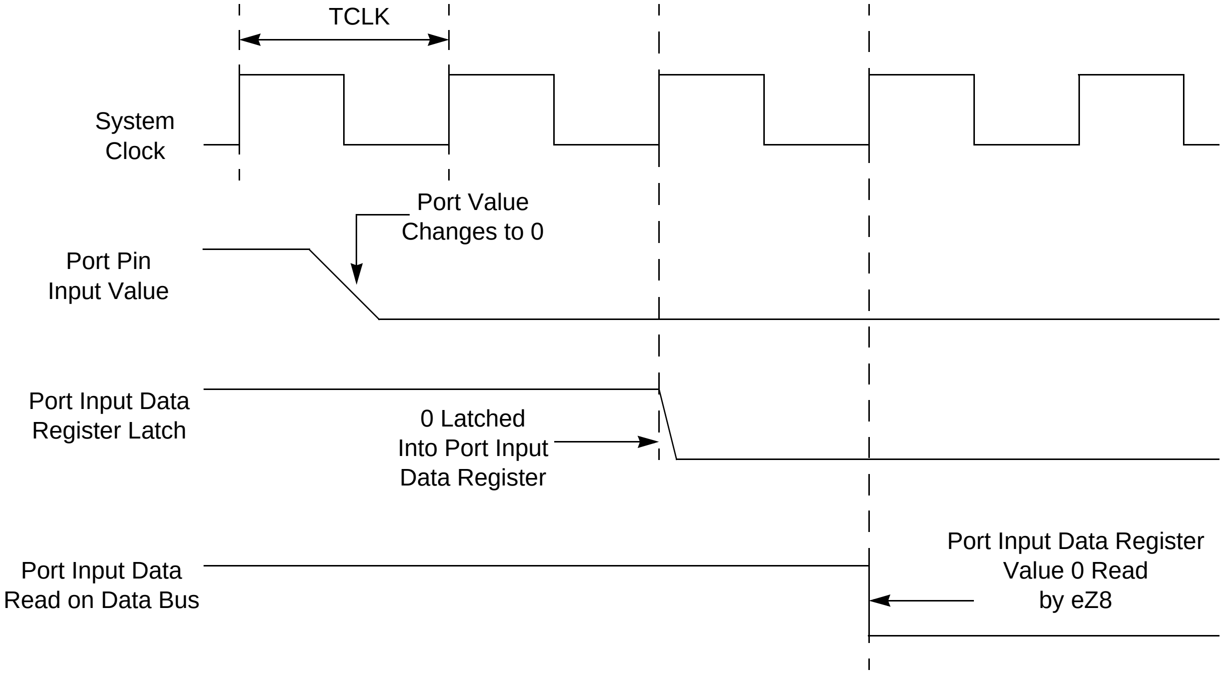

TCLK
System Clock
Port Value Changes to 0
Port Pin Input Value
Port Input Data Register Latch
0 Latched Into Port Input Data Register
Port Input Data Read on Data Bus
Port Input Data Register Value 0 Read by eZ8

Figure 75. Port Input Sample Timing

Table 204. GPIO Port Input Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Min**	**Max**
T_{S_PORT}	Port Input Transition to XIN Rise Setup Time (Not pictured)	5	–
T_{H_PORT}	XIN Rise to Port Input Transition Hold Time (Not pictured)	0	–
T_{SMR}	GPIO Port Pin Pulse Width to ensure Stop Mode Recovery (for GPIO Port Pins enabled as SMR sources)	1 µs	

29.4.4. UART Timing

Figure 78 and Table 207 provide timing information for the UART pins for situations in which CTS is used for flow control. The CTS to DE assertion delay (T1) assumes that the Transmit Data Register has been loaded with data prior to CTS assertion.

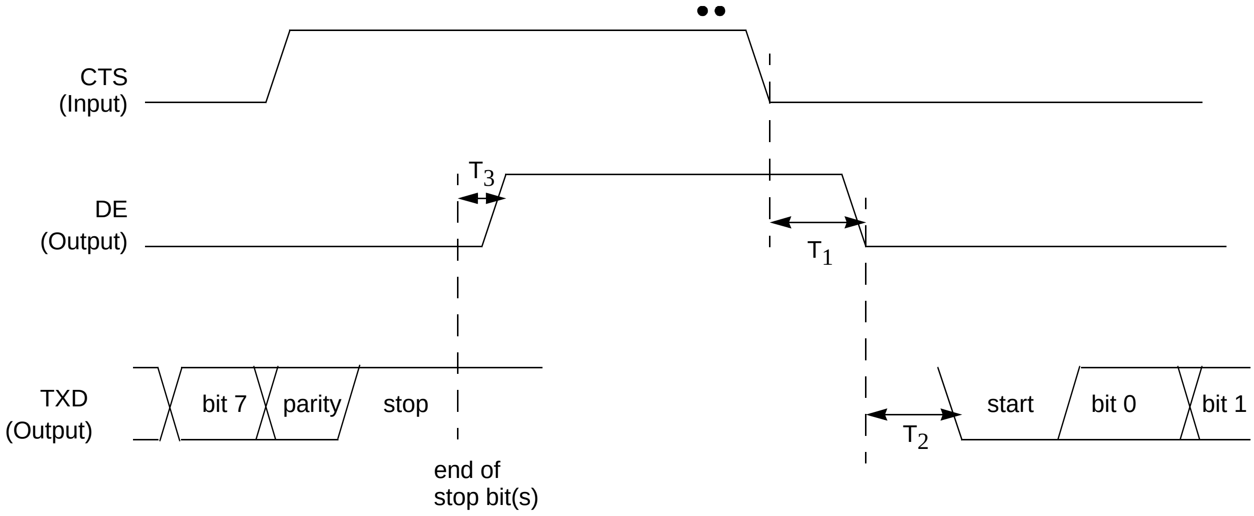


Figure 78. UART Timing With CTS

Table 207. UART Timing with CTS

		Delay (ns)	
Parameter	**Abbreviation**	**Min**	**Max**
UART			
T_1	CTS Fall to DE output delay	2 * X_{IN} period	2 * X_{IN} period + 1 bit time
T_2	DE assertion to TXD falling edge (start bit) delay	± 5	
T_3	End of stop bit(s) to DE deassertion delay	± 5	

Figure 79 and Table 208 provide timing information for the UART pins for situations in which CTS is not used for flow control. DE asserts after the Transmit Data Register has been written. DE remains asserted for multiple characters as long as the Transmit Data Register is written with the next character before the current character has completed.