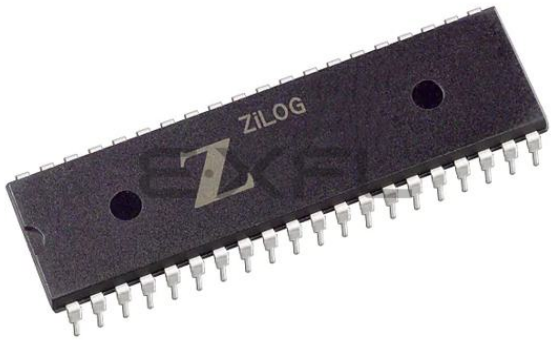

Welcome to E-XFL.COM

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	33
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	-40°C ~ 105°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.620", 15.75mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f2480pm020eg

Bit	Description
[2]	Reserved; must be 0.
[1] COMP0	**Comparator 0 Disable** 0 = Comparator 0 is Enabled (this applies even in STOP Mode). 1 = Comparator 0 is Disabled.
[0] COMP1	**Comparator 1 Disable** 0 = Comparator 1 is Enabled (this applies even in STOP Mode). 1 = Comparator 1 is Disabled.

Table 15. Setup Condition for LVD and VBO Circuits in Different Operation Modes

LVD/VBO Bit Setup[1]		LVD/VBO Bit = "0" or enabled		LVD/VBO Bit = "1" or disabled	
Operation Mode		ACTIVE/HALT Mode	STOP Mode	ACTIVE/HALT Mode	STOP Mode
VBO_AO Bit Setup	VBO_AO="1" or enabled[2]	LVD "ON"		LVD "OFF"	
		VBO "ON"		VBO "ON"	
	VBO_AO="0" or disabled[3]	LVD "ON"		LVD "OFF"	
		VBO "ON"	VBO "OFF"	VBO "OFF"	VBO "OFF"

Notes:
1. The LVD can be turned ON or OFF by the LVD/VBO bit in any mode.
2. When VBO_AO Bit is enabled, VBO is always ON for all modes no matter the setting of LVD/VBO Bit.
3. When VBO_AO Bit is disabled, VBO circuit is always OFF in STOP Mode no matter the setting of LVD/VBO Bit. And VBO can be turned On or OFF by the LVD/VBO Bit in ACTIVE and HALT modes.

Chapter 8. Interrupt Controller

The interrupt controller on the Z8 Encore! XP F1680 Series products prioritizes the interrupt requests from the on-chip peripherals and the GPIO port pins. The interrupt controller includes the following features:

- Thirty-one interrupt sources using twenty-four unique interrupt vectors
  - 16 GPIO port pin interrupt sources (seven interrupt vectors are shared; see Table 36)
  - 15 on-chip peripheral interrupt sources (three interrupt vectors are shared; see Table 36)
- Flexible GPIO interrupts
  - Twelve selectable rising and falling edge GPIO interrupts
  - Four dual-edge interrupts
- Three levels of individually programmable interrupt priority
- WDT can be configured to generate an interrupt

Interrupt requests (IRQs) allow peripheral devices to suspend CPU operation in an orderly manner and force the CPU to start an interrupt service routine (ISR). Usually this interrupt service routine is involved with the exchange of data, status information, or control information between the CPU and the interrupting peripheral. When the service routine is completed, the CPU returns to the operation from which it was interrupted.

The eZ8 CPU supports both vectored and polled interrupt handling. For polled interrupts, the interrupt controller has no effect on operation. For more information about interrupt servicing by the eZ8 CPU, refer to the eZ8 CPU Core User Manual (UM0128), available on www.zilog.com.

8.1. Interrupt Vector Listing

Table 36 lists all of the interrupts available in order of priority. The interrupt vector is stored with the most-significant byte (MSB) at the even Program Memory address and the least-significant byte (LSB) at the following odd Program Memory address.

Note: Some port interrupts are not available on the 20-pin and 28-pin packages. The ADC interrupt is unavailable on devices not containing an ADC.

Chapter 9. Timers

The Z8 Encore! XP F1680 Series products contain three 16-bit reloadable timers that can be used for timing, event counting, or generation of pulse-width modulated signals. The timers' features include:

- 16-bit reload counter
- Programmable prescaler with prescale values ranging from 1 to 128
- PWM output generation
- Capture and compare capability
- Two independent capture/compare channels which reference the common timer
- External input pin for timer input, clock gating, or capture signal. External input pin signal frequency is limited to a maximum of one-fourth the timer clock frequency
- Timer output pin
- Timer interrupt
- Noise Filter on Timer input signal
- Operation in any mode with 32kHz secondary oscillator

In addition to the timers described in this chapter, the Baud Rate Generator (BRG) of unused UART peripheral can also be used to provide basic timing functionality. For more information about using the Baud Rate Generator as additional timers, see the LIN-UART chapter on page 144.

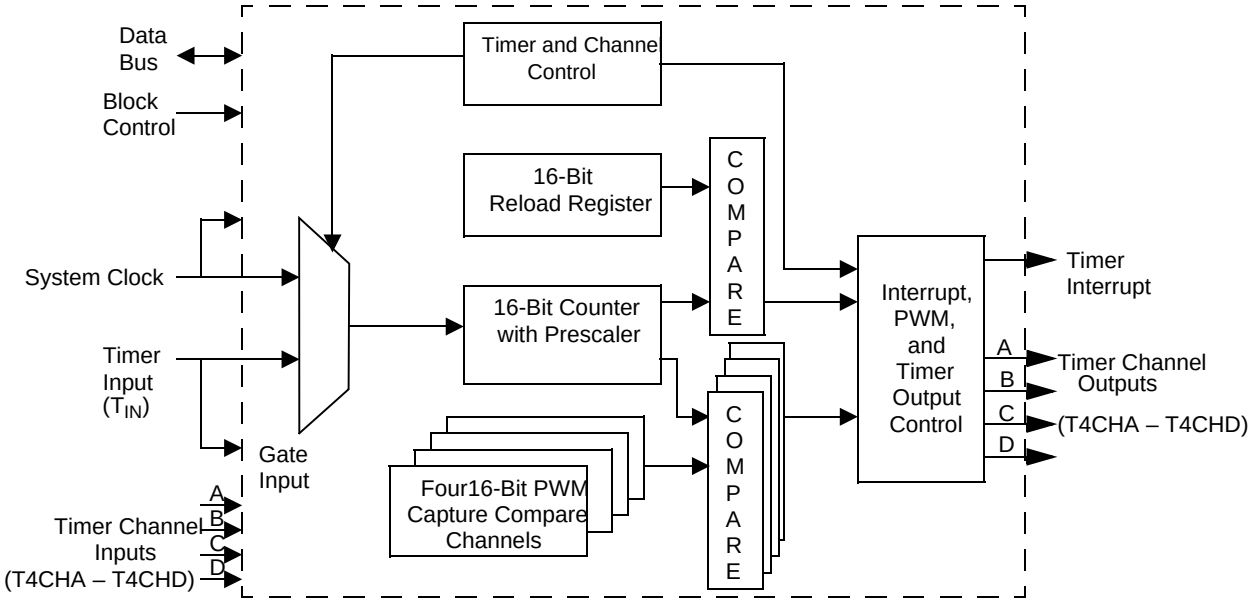

Figure 14. Multi-Channel Timer Block Diagram

10.2. Timer Operation

This section discusses the key features of the Multi-Channel Timer, including its counter, clock source, prescaler and counting modes.

10.2.1. Multi-Channel Timer Counter

The Multi-Channel Timer is based around a 16-bit up/down counter. The counter, depending on the TIMER mode counts up or down with each rising edge of the clock signal. Timer Counter registers MCTH and MCTL can be read/written by software.

10.2.2. Clock Source

The Multi-Channel Timer clock source can come from either the system clock or the alternate function T_{IN} pin when the system clock is the clock source; the alternate function T_{IN} input pin can perform a clock gating function. The TCLKS field in the MCTCTL0 Register selects the timer clock source. When using the T_{IN} pin, the associated GPIO pin, T4CH, must be configured as an input. The T_{IN} frequency cannot exceed one-fourth the system clock frequency.

The window remains open until the count again reaches 8 (in other words, 24 baud clock periods since the previous pulse is detected), giving the endec a sampling window of minus 4 baud rate clocks to plus 8 baud rate clocks around the expected time of an incoming pulse. If an incoming pulse is detected inside this window this process is repeated. If the incoming data is a logical 1 (no pulse), the endec returns to the initial state and waits for the next falling edge. As each falling edge is detected, the endec clock counter is reset, resynchronizing the endec to the incoming signal, allowing the endec to tolerate jitter and baud rate errors in the incoming datastream. Resynchronizing the endec does not alter the operation of the UART, which ultimately receives the data. The UART is only synchronized to the incoming data stream when a start bit is received.

13.3. Infrared Encoder/Decoder Control Register Definitions

All infrared endec configuration and status information is set by the UART control registers as defined beginning on page 163.

Caution: To prevent spurious signals during IrDA data transmission, set the IREN bit in the UART Control 1 Register to 1 to enable the infrared encoder/decoder before enabling the GPIO Port alternate function for the corresponding pin of UART. See Tables 17 through 19 on pages 49–54 for details.

Chapter 16. Enhanced Serial Peripheral Interface

The Enhanced Serial Peripheral Interface (ESPI) supports the Serial Peripheral Interface (SPI) and other synchronous serial interface modes, such as Inter-IC Sound (I^2S) and time division multiplexing (TDM). ESPI includes the following features:

- Full-duplex, synchronous, character-oriented communication
- Four-wire interface ($\overline{SS}$, SCK, MOSI and MISO)
- Data Shift Register is buffered to enable high throughput
- MASTER Mode transfer rates up to a maximum of one-half the system clock frequency
- SLAVE Mode transfer rates up to a maximum of one-eighth the system clock frequency
- Error detection
- Dedicated Programmable Baud Rate Generator
- Data transfer control via polling, interrupt

16.1. Architecture

The ESPI is a full-duplex, synchronous, character-oriented channel that supports a four-wire interface (serial clock, transmit data, receive data and slave select). The ESPI block consists of a shift register, data buffer register, a Baud Rate (clock) Generator, control/status registers and a control state machine. Transmit and receive transfers are in synch as there is a single shift register for both transmitting and receiving data. Figure 33 displays a diagram of the ESPI block.

17.2.5.1. Master Arbitration

If a Master loses arbitration during the address byte it releases the SDA line, switches to SLAVE Mode and monitors the address to determine if it is selected as a Slave. If a Master loses arbitration during the transmission of a data byte, it releases the SDA line and waits for the next stop or start condition.

The Master detects a loss of arbitration when a 1 is transmitted but a 0 is received from the bus in the same bit-time. This loss occurs if more than one Master is simultaneously accessing the bus. Loss of arbitration occurs during the address phase (two or more Masters accessing different slaves) or during the data phase, when the masters are attempting to Write different data to the same Slave.

When a Master loses arbitration, the software is informed by means of the Arbitration Lost interrupt. The software can repeat the same transaction at a later time.

A special case can occur when a Slave transaction starts just before the software attempts to start a new master transaction by setting the start bit. In this case, the state machine enters its Slave states before the start bit is set and as a result the I^2C controller will not arbitrate. If a Slave address match occurs and the I^2C controller receives/transmits data, the start bit is cleared and an Arbitration Lost interrupt is asserted. The software can minimize the chance of this instance occurring by checking the busy bit in the I2CSTATE Register before initiating a Master transaction. If a slave address match does not occur, the Arbitration Lost interrupt will not occur and the start bit will not be cleared. The I^2C controller will initiate the master transaction after the I^2C bus is no longer busy.

17.2.5.2. Master Address-Only Transactions

It is sometimes preferable to perform an address-only transaction to determine if a particular slave device is able to respond. This transaction can be performed by monitoring the ACKV bit in the I2CSTATE Register after the address has been written to the I2CDATA Register and the start bit has been set. After the ACKV bit is set, the ACK bit in the I2CSTATE Register determines if the slave is able to communicate. The stop bit must be set in the I2CCTL Register to terminate the transaction without transferring data. For a 10-bit slave address, if the first address byte is acknowledged, the second address byte should also be sent to determine if the preferred Slave is responding.

Another approach is to set both the stop and start bits (for sending a 7-bit address). After both bits have been cleared (7-bit address has been sent and transaction is complete), the ACK bit can be read to determine if the Slave has acknowledged. For a 10-bit Slave, set the stop bit after the second TDRE interrupt (which indicates that the second address byte is being sent).

17.2.5.3. Master Transaction Diagrams

In the following transaction diagrams, the shaded regions indicate the data that is transferred from the Master to the Slave and the unshaded regions indicate the data that is

Bit	Description (Continued)
[5:2] REFLVL	**Comparator 1 Internal Reference Voltage Level** This reference is independent of the ADC voltage reference. 0000 = 0.0 V 0001 = 0.2 V 0010 = 0.4 V 0011 = 0.6 V 0100 = 0.8 V 0101 = 1.0 V (Default) 0110 = 1.2 V 0111 = 1.4 V 1000 = 1.6 V 1001 = 1.8 V 1010–1111 = Reserved
[1:0] TIMTRG	**Timer Trigger (Comparator Counter Mode)** Enable/disable timer operation. 00 = Disable Timer Trigger. 01 = Comparator 1 output works as Timer 0 Trigger. 10 = Comparator 1 output works as Timer 1 Trigger. 11 = Comparator 1 output works as Timer 2 Trigger.

20.2.8. Flash Controller Behavior in Debug Mode

The following changes in behavior of the Flash Controller occur when the Flash Controller is accessed using the On-Chip Debugger:

- The Flash Write Protect option bit is ignored
- The Flash Sector Protect register is ignored for programming and erase operations
- Programming operations are not limited to the page selected in the Page Select Register
- Bits in the Flash Sector Protect register can be written to 1 or 0
- The second write of the Page Select register to unlock the Flash Controller is not necessary
- The Page Select register can be written when the Flash Controller is unlocked
- The Mass Erase command is enabled through the Flash Control Register

! Caution: For security reasons, the Flash controller allows only a single page to be opened for write/erase. When writing multiple Flash pages, the Flash controller must go through the unlock sequence again to select another page.

20.3. Flash Control Register Definitions

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

Flash Control Register: see page 271

Flash Status Register: see page 272

Flash Page Select Register: see page 273

Flash Sector Protect Register: see page 274

Flash Frequency High and Low Byte Registers: see page 274

20.3.1. Flash Control Register

The Flash Controller must be unlocked using the Flash Control Register (see Table 134) before programming or erasing Flash memory. The Flash Controller is unlocked by writing to the Flash Page Select Register, then 73H 8CH, sequentially, to the Flash Control Register, and finally again to the Flash Page Select Register with the same value as the previous write. When the Flash Controller is unlocked, Mass Erase or Page Erase can be initiated by writing the appropriate command to the FCTL. Erase applies only to the active page selected in the Flash Page Select Register. Mass Erase is enabled only through the

21.1.2. Option Bit Types

This section describes the User, Trim and Calibration option bit types.

21.1.2.1. User Option Bits

The user option bits are contained in the first two bytes of Program Memory. User access to these bits has been provided because these locations contain application-specific device configurations. The information contained here is lost when page 0 of the Program Memory is erased.

21.1.2.2. Trim Option Bits

The trim option bits are contained in the Flash memory information page. These bits are factory programmed values required to optimize the operation of onboard analog circuitry and cannot be permanently altered by the user. Program Memory can be erased without endangering these values. It is possible to alter working values of these bits by accessing the Trim Bit Address and Data registers, but these working values are lost after a power loss.

There are 32 bytes of trim data. To modify one of these values the user code must first write a value between 00H and 1FH into the Trim Bit Address Register. The next write to the Trim Bit Data Register changes the working value of the target trim data byte.

Reading the trim data requires the user code to write a value between 00H and 1FH into the Trim Bit Address Register. The next read from the Trim Bit Data Register returns the working value of the target trim data byte.

Note: The trim address ranges from information address 20-3F only. The remainder of the information page is not accessible via the trim bit address and data registers.

21.1.2.3. Calibration Option Bits

The calibration option bits are also contained in the information page. These bits are factory programmed values intended for use in software correcting the device's analog performance. To read these values, the user code must employ the LDC instruction to access the information area of the address space as defined in the Flash Information Area section on page 21.

The following code example shows how to read the calibration data from the Flash Information Area.

```
; get value at info address 60 (FE60h)
ldx FPS, #%80 ; enable access to flash info page
ld R0, #%FE
```

21.2.5. Zilog Calibration Option Bits

This section describes the calibration of the temperature sensor's Low and High bytes.

21.2.5.1. Temperature Sensor Calibration High and Low Byte Registers

Tables 157 and 158 present the Temperature Sensor Calibration High and Low Byte registers.

Table 157. Temperature Sensor Calibration High Byte at FE60H (TEMPCALH)

Bits	7	6	5	4	3	2	1	0
Field	TEMPCALH							
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 FE60H							

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

Bit	Description
[7:0] TEMPCALH	**Temperature Sensor Calibration High Byte** Bits [7:3] of this register are not used. Bit 2 indicates whether the calibration data is added to or subtracted from the measured ADC data. If bit 2 is 0, the calibration data is added; if bit 2 is 1, the calibration data is subtracted. Bits 1 and 0 are the High two bits of the 10-bit calibration data.

Table 158. Temperature Sensor Calibration Low Byte at FE61H (TEMPCALL)

Bits	7	6	5	4	3	2	1	0
Field	TEMPCALL							
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 FE61H							

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

Bit	Description
[7:0] TEMPCALL	**Temperature Sensor Calibration Low Byte** TEMPCALL is the low eight bits of 10-bit calibration data. The entire 10-bit calibration data field is {TEMPCALH[1:0], TEMPCALL}.

Table 162. OCD Baud-Rate Limits

System Clock Frequency	Maximum Asynchronous Baud Rate (bits/s)	Minimum Baud Rate (bits/s)
20.0 MHz	2.5 M	39.1 k
1.0MHz	125 k	1.96K
32kHz	4096	64

If the OCD receives a Serial Break (ten or more continuous bits Low), the Autobaud Detector/Generator resets. The Autobaud Detector/Generator can then be reconfigured by sending 80H. If the Autobaud Detector overflows while measuring the Autobaud character, the Autobaud Detector will remain reset.

23.2.4. High Speed Synchronous

It is possible to operate the serial On-Chip Debugger at high speeds. To operate at high speeds, data must be synchronized with an external clock. High speed synchronous communication will only work when using an external clock source. To operate in high-speed synchronous mode, simply Autobaud to the appropriate speed. The Autobaud generator will automatically run at the appropriate baud rate.

Slow bus rise times due to the pullup resistor become a limiting factor when operating at high speeds. To compensate for slow rise times, the output driver can be configured to drive the line High. If the TXD (Transmit Drive) bit is set, the line will be driven both High and Low during transmission. The line starts being driven at the beginning of the start bit and stops being driven at the middle of the stop bit. If the TXDH (Transmit Drive High) bit is set, the line will be driven High until the input is High or the center of the bit occurs, whichever is first. If both TXD and TXDH are set, the pin will be driven High for one clock period at the beginning of each 0 to 1 transition. An example of a high-speed synchronous interface is displayed in Figure 60.

enabled, the OCD ignores the BRK signal and the BRK instruction operates as an NOP instruction.

If breakpoints are enabled, the OCD can be configured to automatically enter DEBUG mode, or to loop on the break instruction. If the OCD is configured to loop on the BRK instruction, then the CPU remains able to service interrupt requests.

The loop on a BRK instruction can service interrupts in the background. For interrupts to be serviced in the background, there cannot be any breakpoints in the interrupt service routine. Otherwise, the CPU stops on the breakpoint in the interrupt routine. For interrupts to be serviced in the background, interrupts must also be enabled. Interrupts are typically disabled during critical sections of code where interrupts do not occur (such as adjusting the stack pointer or modifying shared data).

Through the OCD, host debugger software can poll the IDLE bit of the OCDSTAT Register to determine if the OCD is looping on a BRK instruction. When the host must stop the CPU on the BRK instruction on which it is looping, the host must not set the DBGMODE bit of the OCDCTL register. The CPU may have vectored to an interrupt service routine. Instead, the host clears the BRKLOOP bit, thereby allowing the CPU to finish the interrupt service routine and return to the BRK instruction. When the CPU returns to the BRK instruction on which it was previously looping, it automatically sets the DBGMODE bit and enters DEBUG mode.

The majority of the OCD commands remain disabled when the eZ8 CPU is looping on a BRK instruction. The eZ8 CPU must be in DEBUG mode before these commands can be issued.

23.2.8.1. Breakpoints in Flash Memory

The BRK instruction is op code `00H`, which corresponds to the fully programmed state of a byte in Flash memory. To implement a breakpoint, write `00H` to the appropriate address, overwriting the current instruction. To remove a breakpoint, erase the corresponding page of Flash memory and reprogram with the original data.

23.2.9. OCDCNTR Register

The On-Chip Debugger contains a multipurpose 16-bit Counter Register. It can be used for the following:

- Count system clock cycles between breakpoints
- Generate a BRK when it counts down to 0
- Generate a BRK when its value matches the Program Counter

When configured as a counter, the OCDCNTR Register starts counting when the On-Chip Debugger exits DEBUG mode and stops counting when it enters DEBUG mode again or

Table 184. Program Control Instructions

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

Table 185. Rotate and Shift Instructions

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

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
CLR dst	dst ← 00H	R		B0	–	–	–	–	–	–	2	2
		IR		B1							2	3
COM dst	dst ← ~dst	R		60	–	*	*	0	–	–	2	2
		IR		61							2	3
CP dst, src	dst – src	r	r	A2	*	*	*	*	–	–	2	3
		r	Ir	A3							2	4
		R	R	A4							3	3
		R	IR	A5							3	4
		R	IM	A6							3	3
		IR	IM	A7							3	4
CPC dst, src	dst – src – C	r	r	1F A2	*	*	*	*	–	–	3	3
		r	Ir	1F A3							3	4
		R	R	1F A4							4	3
		R	IR	1F A5							4	4
		R	IM	1F A6							4	3
		IR	IM	1F A7							4	4
CPCX dst, src	dst – src – C	ER	ER	1F A8	*	*	*	*	–	–	5	3
		ER	IM	1F A9							5	3
CPX dst, src	dst – src	ER	ER	A8	*	*	*	*	–	–	4	3
		ER	IM	A9							4	3
DA dst	dst ← DA(dst)	R		40	*	*	*	X	–	–	2	2
		IR		41							2	3
DEC dst	dst ← dst – 1	R		30	–	*	*	*	–	–	2	2
		IR		31							2	3
DECW dst	dst ← dst – 1	RR		80	–	*	*	*	–	–	2	5
		IRR		81							2	6
DI	IRQCTL[7] ← 0			8F	–	–	–	–	–	–	1	2

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

Figure 73 displays the STOP Mode supply current versus ambient temperature and V_{DD} level with all peripherals disabled.

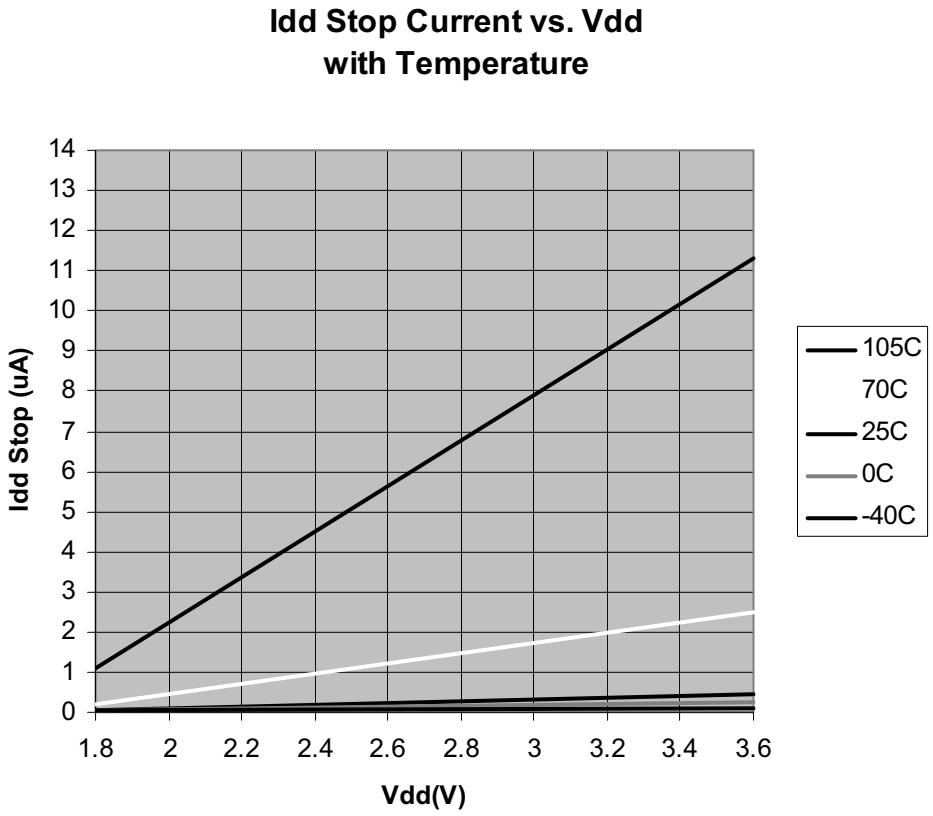

Figure 73. STOP Mode Current Consumption as a Function of V_{DD} with Temperature as a Parameter; all Peripherals Disabled

Table 200. IPO Electrical Characteristics (Continued)

		V_{DD} = 1.8 to 3.6V T_A = –40°C to +105°C			V_{DD} = 2.7 to 3.6V T_A = 0°C to +70°C				
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
F_{IPO}	Output Frequency	10.6168	11.0592	11.5016	10.78272	11.0592	11.33568	MHz	±2.5% 2.7 to 3.6 V, 0–70°C; ±4% 1.8 to 2.7 V, 0–70°C ±4% 1.8 to 3.6 V, –40–105°C
	Divided-by-2 Output Frequency	5.3084	5.5296	5.7508	5.39136	5.5296	5.66784		
	Divided-by-4 Output Frequency	2.6542	2.7648	2.8754	2.69568	2.7648	2.83392		
	Divided-by-8 Output Frequency	1.3271	1.3824	1.4377	1.34784	1.3824	1.41696		
	Divided-by-16 Output Frequency	0.6636	0.6912	0.7188	0.67392	0.6912	0.70848		
	Divided-by-32 Output Frequency	0.3318	0.3456	0.3594	0.33696	0.3456	0.35424		
	Divided-by-128 Output Frequency	0.0829	0.0864	0.0899	0.08424	0.0864	0.08856		
	Divided-by-256 Output Frequency	0.0415	0.0432	0.0449	0.04212	0.0432	0.04428		
	Duty Cycle of Output	45		55	45		55	%	

Table 201. Low Voltage Detect Electrical Characteristics

		T_A = 0°C to +70°C T_A = –40°C to +105°C V_{DD} = 1.8 to 3.6 V				
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
$I_{DD}LVD$	LVD Active Current	–	–	50	µA	
$I_{DDQ}LVD$	LVD Quiescent Current	–	5	–	nA	
V_{TH}	Detected Source Voltage	V_{TP} – 10%	V_{TP}[1]	V_{TP} + 10%	V	

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.

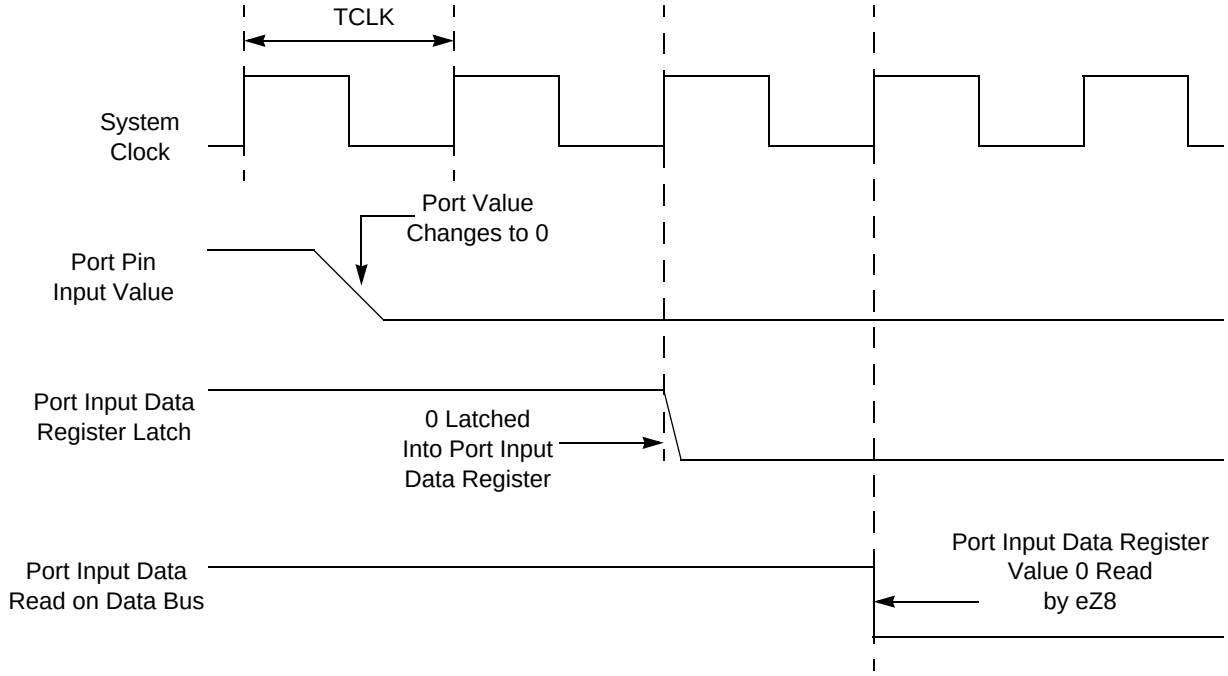

Figure 75. Port Input Sample Timing

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	

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.