# E·XFL



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

#### Details

Product Status	Obsolete
Core Processor	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	IrDA, UART/USART
Peripherals	Brown-out Detect/Reset, LED, LVD, POR, PWM, Temp Sensor, WDT
Number of I/O	17
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 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/z8f042ahh020eg

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

# **LED Drive Enable Register**

The LED Drive Enable Register, shown in Table 31, activates the controlled current drive. The Alternate Function Register has no control over the LED function; therefore, setting the Alternate Function Register to select the LED function is not required. LEDEN bits [7:0] correspond to Port C bits [7:0], respectively.

Bit	7	7 6 5 4 3 2 1 0								
Field		LEDEN[7:0]								
RESET	0	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				F8	2H					

Table 31		Drive	Enable	
Table ST.	LEV	Dire	Ellable	(LEDEN)

#### Bit Description

[7:0] LED Drive Enable

LEDENx These bits determine which Port C pins are connected to an internal current sink.

0 = Tristate the Port C pin.

1 = Enable controlled current sink on the Port C pin.

**Note:** *x* indicates the specific GPIO port pin number (7–0).

# LED Drive Level High Register

The LED Drive Level registers contain two control bits for each Port C pin, as shown in Table 32. These two bits select between four programmable drive levels. Each pin is individually programmable.

Bit	7	6	5	4	3	2	1	0		
Field		LEDLVLH[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		F83H								
Bit	Descrip	tion								
[7:0] LEDLVLHx	<b>LED Lev</b> {LEDLVL 00 = 3m	<b>/el High Bit</b> .H, LEDLVLL A	_} select one	of four prog	ammable cu	urrent drive le	evels for eac	n Port C pin.		

Tabla '	22 I EI	סעויים ר		Linh	Dogistor	٨
i abie .	32. LEI	JUIIVe	Levei	піціі	Redister	

01 = 7mA 10 = 13mA

11 = 20 mA

Note: x indicates the specific GPIO port pin number (7–0).

Table 46.	<b>IRQ2</b> Enable	Low Bit	Register	(IRQ2ENL)
			riogiotoi	(

Bit	7	6	5	4	3	2	1	0
Field		Rese	erved		C3ENL	C2ENL	C1ENL	C0ENL
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
Address				FC	8H			

Bit	Description
[7:4]	<b>Reserved</b> These bits are reserved and must be programmed to 0000.
[3] C3ENL	Port C3 Interrupt Request Enable Low Bit
[2] C2ENL	Port C2 Interrupt Request Enable Low Bit
[1] C1ENL	Port C1 Interrupt Request Enable Low Bit
[0] C0ENL	Port C0 Interrupt Request Enable Low Bit

# Interrupt Edge Select Register

The Interrupt Edge Select (IRQES) Register, shown in Table 47, determines whether an interrupt is generated for the rising edge or falling edge on the selected GPIO Port A input pin.

Bit	7	6	5	4	3	2	1	0	
Field	IES7	IES6	IES5	IES4	IES3	IES2	IES1	IES0	
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				FC	DH				

	Table 47	. Interrupt	Edge	Select	Register	(IRQES)
--	----------	-------------	------	--------	----------	---------

Bit	Description
[7:0]	Interrupt Edge Select x
IESx	0 = An interrupt request is generated on the falling edge of the PAx input.
	1 = An interrupt request is generated on the rising edge of the PAx input.
Note:	x indicates the specific GPIO port pin number (0–7).

## **Timer Reload High and Low Byte Registers**

The Timer 0–1 Reload High and Low Byte (TxRH and TxRL) registers, shown in Tables 54 and 55, store a 16-bit reload value, {TRH[7:0], TRL[7:0]}. Values written to the Timer Reload High Byte Register are stored in a temporary holding register. When a write to the Timer Reload Low Byte Register occurs, the temporary holding register value is written to the Timer High Byte Register. This operation allows simultaneous updates of the 16-bit Timer reload value.

In COMPARE Mode, the Timer Reload High and Low Byte registers store the 16-bit Compare value.

Bit	7	6	5	4	3	2	1	0		
Field				TF	RH					
RESET	1	1 1 1 1 1 1 1 1								
R/W	R/W	R/W R/W R/W R/W R/W R/W R/W								
Address				F02H,	F0AH					

#### Table 54. Timer 0–1 Reload High Byte Register (TxRH)

Bit	7	6	5	4	3	2	1	0
Field				TF	RL			
RESET	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address		F03H, F0BH						

Bit	Description
[7:0]	Timer Reload Register High and Low
TRH, TRL	These two bytes form the 16-bit reload value, {TRH[7:0], TRL[7:0]}. This value sets the max- imum count value which initiates a timer reload to 0001H. In COMPARE Mode, these two bytes form the 16-bit Compare value.

# **Analog-to-Digital Converter**

The analog-to-digital converter (ADC) converts an analog input signal to its digital representation. The features of this sigma-delta ADC include:

- 11-bit resolution in DIFFERENTIAL Mode
- 10-bit resolution in SINGLE-ENDED Mode
- Eight single-ended analog input sources are multiplexed with general-purpose I/O ports
- 9<sup>th</sup> analog input obtained from temperature sensor peripheral
- 11 pairs of differential inputs also multiplexed with general-purpose I/O ports
- Low-power operational amplifier (LPO)
- Interrupt on conversion complete
- Bandgap generated internal voltage reference with two selectable levels
- Manual in-circuit calibration is possible employing user code (offset calibration)
- Factory calibrated for in-circuit error compensation

# Architecture

Figure 19 displays the major functional blocks of the ADC. An analog multiplexer network selects the ADC input from the available analog pins, ANA0 through ANA7.

The input stage of the ADC allows both differential gain and buffering. The following input options are available:

- Unbuffered input (SINGLE-ENDED and DIFFERENTIAL modes)
- Buffered input with unity gain (SINGLE-ENDED and DIFFERENTIAL modes)
- LPO output with full pin access to the feedback path

- Write to BUFMODE[2:0] to select SINGLE-ENDED or DIFFERENTIAL mode, plus unbuffered or buffered mode.
- Write the REFSELH bit of the pair {REFSELH, REFSELL} to select the internal voltage reference level or to disable the internal reference. The REFSELL bit is contained in the ADC Control Register 0.
- 3. Write to the ADC Control Register 0 to configure the ADC for continuous conversion. The bit fields in the ADC Control Register may be written simultaneously:
  - Write to the ANAIN[3:0] field to select from the available analog input sources (different input pins available depending on the device).
  - Set CONT to 1 to select continuous conversion.
  - If the internal VREF must be output to a pin, set the REFEXT bit to 1. The internal voltage reference must be enabled in this case.
  - Write the REFSELL bit of the pair {REFSELH, REFSELL} to select the internal voltage reference level or to disable the internal reference. The REFSELH bit is contained in ADC Control/Status Register 1.
  - Set CEN to 1 to start the conversions.
- 4. When the first conversion in continuous operation is complete (after 5129 system clock cycles, plus the 40 cycles for power-up, if necessary), the ADC control logic performs the following operations:
  - CEN resets to 0 to indicate the first conversion is complete. CEN remains 0 for all subsequent conversions in continuous operation
  - An interrupt request is sent to the Interrupt Controller to indicate the conversion is complete
- 5. The ADC writes a new data result every 256 system clock cycles. For each completed conversion, the ADC control logic performs the following operations:
  - Writes the 13-bit two's complement result to {ADCD\_H[7:0], ADCD\_L[7:3]}
  - Sends an interrupt request to the Interrupt Controller denoting conversion complete
- 6. To disable continuous conversion, clear the CONT bit in the ADC Control Register to 0.

#### Interrupts

The ADC is able to interrupt the CPU when a conversion has been completed. When the ADC is disabled, no new interrupts are asserted; however, an interrupt pending when the ADC is disabled is not cleared.

# **Calibration and Compensation**

The Z8 Encore! XP F082A Series ADC is factory calibrated for offset error and gain error, with the compensation data stored in Flash memory. Alternatively, you can perform your own calibration, storing the values into Flash themselves. Thirdly, the user code can perform a manual offset calibration during DIFFERENTIAL Mode operation.

### **Factory Calibration**

Devices that have been factory calibrated contain 30 bytes of calibration data in the Flash option bit space. This data consists of 3 bytes for each input mode, one for offset and two for gain correction. For a list of input modes for which calibration data exists, see the <u>Zilog Calibration Data</u> section on page 168.

### **User Calibration**

If you have precision references available, its own external calibration can be performed using any input modes. This calibration data takes into account buffer offset and nonlinearity; therefore Zilog recommends that this calibration be performed separately for each of the ADC input modes planned for use.

### **Manual Offset Calibration**

When uncalibrated, the ADC has significant offset (see <u>Table 139</u> on page 236). Subsequently, manual offset calibration capability is built into the block. When the ADC Control Register 0 sets the input mode (ANAIN[2:0]) to MANUAL OFFSET CALIBRATION Mode, the differential inputs to the ADC are shorted together by an internal switch. Reading the ADC value at this point produces 0 in an ideal system. The value actually read is the ADC offset. This value can be stored in nonvolatile memory (see the <u>Nonvolatile Data Storage</u> chapter on page 176) and accessed by user code to compensate for the input offset error. There is no provision for manual gain calibration.

# Software Compensation Procedure Using Factory Calibration Data

The value read from the ADC high and low byte registers is uncompensated. The user mode software must apply gain and offset correction to this uncompensated value for maximum accuracy. The following equation yields the compensated value:

$$ADC_{comp} = (ADC_{uncomp} - OFFCAL) + ((ADC_{uncomp} - OFFCAL) \times GAINCAL)/2$$

where GAINCAL is the gain calibration value, OFFCAL is the offset calibration value and  $ADC_{uncomp}$  is the uncompensated value read from the ADC. All values are in two's complement format.

#### 150

#### Table 79. Flash Code Protection Using the Flash Option Bits

FWP	Flash Code Protection Description
0	Programming and erasing disabled for all of Flash Program Mem-
	ory. In user code programming, Page Erase and Mass Erase are all
	disabled. Mass Erase is available through the On-Chip Debugger.
1	Programming, Page Erase and Mass Erase are enabled for all of
	Flash Program Memory.

#### Flash Code Protection Using the Flash Controller

At Reset, the Flash Controller locks to prevent accidental program or erasure of the Flash memory. To program or erase the Flash memory, first write the Page Select Register with the target page. Unlock the Flash Controller by making two consecutive writes to the Flash Control Register with the values 73H and 8CH, sequentially. The Page Select Register must be rewritten with the target page. If the two Page Select writes do not match, the controller reverts to a locked state. If the two writes match, the selected page becomes active. See Figure 22 on page 148 for details.

After unlocking a specific page, you can enable either Page Program or Erase. Writing the value 95H causes a Page Erase only if the active page resides in a sector that is not protected. Any other value written to the Flash Control Register locks the Flash Controller. Mass Erase is not allowed in the user code but only in through the Debug Port.

After unlocking a specific page, you can also write to any byte on that page. After a byte is written, the page remains unlocked, allowing for subsequent writes to other bytes on the same page. Further writes to the Flash Control Register cause the active page to revert to a locked state.

#### **Sector-Based Flash Protection**

The final protection mechanism is implemented on a per-sector basis. The Flash memories of Z8 Encore! XP devices are divided into maximum number of 8 sectors. A sector is 1/8 of the total Flash memory size unless this value is smaller than the page size – in which case, the sector and page sizes are equal. On Z8 Encore! F082A Series devices, the sector size is varied according to the Flash memory configuration shown in <u>Table 78</u> on page 146.

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

The Flash Sector Protect Register shares its Register File address with the Page Select Register. The Flash Sector Protect Register is accessed by writing the Flash Control Regis**Caution:** The byte at each address of the Flash memory cannot be programmed (any bits written to 0) more than twice before an erase cycle occurs. Doing so may result in corrupted data at the target byte.

### **Page Erase**

The Flash memory can be erased one page (512 bytes) at a time. Page Erasing the Flash memory sets all bytes in that page to the value FFH. The Flash Page Select Register identifies the page to be erased. Only a page residing in an unprotected sector can be erased. With the Flash Controller unlocked and the active page set, writing the value 95h to the Flash Control Register initiates the Page Erase operation. While the Flash Controller executes the Page Erase operation, the eZ8 CPU idles but the system clock and on-chip peripherals continue to operate. The eZ8 CPU resumes operation after the Page Erase operation completes. If the Page Erase operation is performed using the On-Chip Debugger, poll the Flash Status Register to determine when the Page Erase operation is complete. When the Page Erase is complete, the Flash Controller returns to its locked state.

### **Mass Erase**

The Flash memory can also be Mass Erased using the Flash Controller, but only by using the On-Chip Debugger. Mass Erasing the Flash memory sets all bytes to the value FFH. With the Flash Controller unlocked and the Mass Erase successfully enabled, writing the value 63H to the Flash Control Register initiates the Mass Erase operation. While the Flash Controller executes the Mass Erase operation, the eZ8 CPU idles but the system clock and on-chip peripherals continue to operate. Using the On-Chip Debugger, poll the Flash Status Register to determine when the Mass Erase operation is complete. When the Mass Erase is complete, the Flash Controller returns to its locked state.

### **Flash Controller Bypass**

The Flash Controller can be bypassed and the control signals for the Flash memory brought out to the GPIO pins. Bypassing the Flash Controller allows faster Row Programming algorithms by controlling the Flash programming signals directly.

Row programming is recommended for gang programming applications and large volume customers who do not require in-circuit initial programming of the Flash memory. Page Erase operations are also supported when the Flash Controller is bypassed.

For more information about bypassing the Flash Controller, refer to the <u>Third-Party Flash</u> <u>Programming Support for Z8 Encore! MCUs Application Note (AN0117)</u>, which is available for download on <u>www.zilog.com</u>.

# Flash Page Select Register

The Flash Page Select (FPS) Register shares address space with the Flash Sector Protect Register. Unless the Flash controller is unlocked and written with 5EH, writes to this address target the Flash Page Select Register.

The register is used to select one of the available Flash memory pages to be programmed or erased. Each Flash Page contains 512 bytes of Flash memory. During a Page Erase operation, all Flash memory having addresses with the most significant 7 bits given by FPS[6:0] are chosen for program/erase operation.

Bit	7	6	5	4	3	2	1	0
Field	INFO_EN				PAGE			
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		FF9H						

#### Table 82. Flash Page Select Register (FPS)

#### Bit Description

#### [7] Information Area Enable

INFO\_EN 0 = Information Area us not selected.

1 = Information Area is selected. The Information Area is mapped into the Program Memory address space at addresses FE00H through FFFFH.

#### [6:0] Page Select

PAGE This 7-bit field identifies the Flash memory page for Page Erase and page unlocking. Program Memory Address[15:9] = PAGE[6:0]. For the Z8F08xx devices, the upper 3 bits must be zero. For the Z8F04xx devices, the upper 4 bits must be zero. For Z8F02xx devices, the upper 5 bits must always be 0. For the Z8F01xx devices, the upper 6 bits must always be 0.

**Caution:** The Flash Frequency High and Low Byte registers must be loaded with the correct value to ensure proper operation of the device. Also, Flash programming and erasure is not supported for system clock frequencies below 20kHz or above 20MHz.

Table 84. Flash Frequency High Byte Register (FFREQH)	

Bit	7	6	5	4	3	2	1	0
Field				FFR	EQH			
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	FFAH							

Bit	Description
[7:0]	Flash Frequency High Byte
FFREQH	High byte of the 16-bit Flash Frequency value.

#### Table 85. Flash Frequency Low Byte Register (FFREQL)

Bit	7	6	5	4	3	2	1	0
Field	FFREQL							
RESET		0						
R/W		R/W						
Address		FFBH						

Bit	Description
[7:0]	Flash Frequency Low Byte
FFREQL	Low byte of the 16-bit Flash Frequency value.

### Flash Program Memory Address 0001H

#### Table 89. Flash Options Bits at Program Memory Address 0001H

Bit	7	6	5	4	3	2	1	0
Field		Reserved		XTLDIS	Reserved			
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		Program Memory 0001H						

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

Bit	Description						
[7:5]	Reserved						
	These bits are reserved and must be programmed to 111.						
[4]	State of the Crystal Oscillator at Reset						
XTLDIS	This bit only enables the crystal oscillator. Its selection as a system clock must be performed manually.						
	0 = Crystal oscillator is enabled during reset, resulting in longer reset timing.						
	1 = Crystal oscillator is disabled during reset, resulting in shorter reset timing.						
	<b>Caution:</b> Programming the XTLDIS bit to zero on 8-pin versions of this device prevents any further communication via the debug pin due to the fact that the XIN and DBG functions are shared on pin 2 of this package. Do not program this bit to zero on 8-pin devices unless further debugging or Flash programming is not required.						
[3:0]	Reserved						

These bits are reserved and must be programmed to 1111.

## Trim Bit Address 0002H

#### Table 92. Trim Option Bits at 0002H (TIPO)

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

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

# Trim Bit Address 0003H

**Note:** The LVD is available on 8-pin devices only.

#### Table 93. Trim Option Bits at Address 0003H (TLVD)

Bit	7	6	5	4	3	2	1	0
Field	Reserved			LVD_TRIM				
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	is Information Page Memory 0023H							
Note: U = Unchanged by Reset, R/W = Read/Write.								

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

Bit	Description
[7:5]	<b>Reserved</b> These bits are reserved and must be programmed to 111.
[4:0] LVD_TRIM	<b>Low Voltage Detect Trimm</b> This trimming affects the low voltage detection threshold. Each LSB represents a 50mV change in the threshold level. Alternatively, the low voltage threshold may be computed from the options bit value by the following equation:
	$LVD_LVL = 3.6 V - LVD_TRIM \times 0.05 V$
	These values are tabulated in Table 94.



Figure 25. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface; #2 of 2

## **DEBUG Mode**

The operating characteristics of the devices in DEBUG Mode are:

- The eZ8 CPU fetch unit stops, idling the eZ8 CPU, unless directed by the OCD to execute specific instructions
- The system clock operates unless in STOP Mode
- All enabled on-chip peripherals operate unless in STOP Mode
- Automatically exits HALT Mode
- Constantly refreshes the Watchdog Timer, if enabled

### **Entering DEBUG Mode**

The operating characteristics of the devices entering DEBUG Mode are:

- The device enters DEBUG Mode after the eZ8 CPU executes a BRK (Breakpoint) instruction
- If the DBG pin is held Low during the final clock cycle of system reset, the part enters DEBUG Mode immediately (20-/28-pin products only)

**Note:** Holding the DBG pin Low for an additional 5000 (minimum) clock cycles after reset (making sure to account for any specified frequency error if using an internal oscillator) prevents a false interpretation of an Autobaud sequence (see the <u>OCD Auto-Baud Detector/Generator</u> section on page 183).

When selecting a new clock source, the system clock oscillator failure detection circuitry and the Watchdog Timer oscillator failure circuitry must be disabled. If SOFEN and WOFEN are not disabled prior to a clock switch-over, it is possible to generate an interrupt for a failure of either oscillator. The Failure detection circuitry can be enabled anytime after a successful write of OSCSEL in the OSCCTL Register.

The internal precision oscillator is enabled by default. If the user code changes to a different oscillator, it may be appropriate to disable the IPO for power savings. Disabling the IPO does not occur automatically.

### **Clock Failure Detection and Recovery**

Should an oscillator or timer fail, there are methods of recovery, as this section describes.

#### System Clock Oscillator Failure

The Z8F04xA family devices can generate nonmaskable interrupt-like events when the primary oscillator fails. To maintain system function in this situation, the clock failure recovery circuitry automatically forces the Watchdog Timer oscillator to drive the system clock. The Watchdog Timer oscillator must be enabled to allow the recovery. Although this oscillator runs at a much slower speed than the original system clock, the CPU continues to operate, allowing execution of a clock failure vector and software routines that either remedy the oscillator failure or issue a failure alert. This automatic switch-over is not available if the Watchdog Timer oscillator is disabled, though it is not necessary to enable the Watchdog Timer reset function (see the <u>Watchdog Timer</u> chapter on page 93).

The primary oscillator failure detection circuitry asserts if the system clock frequency drops below  $1 \text{ kHz} \pm 50\%$ . If an external signal is selected as the system oscillator, it is possible that a very slow but nonfailing clock can generate a failure condition. Under these conditions, do not enable the clock failure circuitry (SOFEN must be deasserted in the OSCCTL Register).

#### Watchdog Timer Failure

In the event of a Watchdog Timer oscillator failure, a similar nonmaskable interrupt-like event is issued. This event does not trigger an attendant clock switch-over, but alerts the CPU of the failure. After a Watchdog Timer failure, it is no longer possible to detect a primary oscillator failure. The failure detection circuitry does not function if the Watchdog Timer is used as the system clock oscillator or if the Watchdog Timer oscillator has been disabled. For either of these cases, it is necessary to disable the detection circuitry by deasserting the WDFEN bit of the OSCCTL Register.

The Watchdog Timer oscillator failure detection circuit counts system clocks while looking for a Watchdog Timer clock. The logic counts 8004 system clock cycles before determining that a failure has occurred. The system clock rate determines the speed at which

			• · ·
Parameter	Value	Units	Comments
Frequency	20	MHz	
Resonance	Parallel		
Mode	Fundamental		
Series Resistance (R <sub>S</sub> )	60	W	Maximum
Load Capacitance (C <sub>L</sub> )	30	pF	Maximum
Shunt Capacitance (C <sub>0</sub> )	7	pF	Maximum
Drive Level	1	mW	Maximum

Table 114. Recommended Crystal Oscillator Specifications

Table 115.	Transconductance	Values for Low.	Medium and Hic	ah Gain Op	erating Modes

Mode	Crystal Frequency Range	Function	Transco (U៖ for	nductanc se this rai calculatio	e (mA/V) nge ons)
Low Gain*	32kHz-1MHz	Low Power/Frequency Applications	0.02	0.04	0.09
Medium Gain*	0.5MHz-10MHz	Medium Power/Frequency Applications	0.84	1.7	3.1
High Gain*	8MHz–20MHz	High Power/Frequency Applications	1.1	2.3	4.2
Note: *Printed circuit board layouts must not add more than 4pF of stray capacitance to either the X <sub>IN</sub> or X <sub>OUT</sub> pins.					

if no oscillation occurs, reduce the values of the capacitors C1 and C2 to decrease the loading.

# **Oscillator Operation with an External RC Network**

Figure 28 displays a recommended configuration for connection with an external resistorcapacitor (RC) network.



#### Figure 28. Connecting the On-Chip Oscillator to an External RC Network

An external resistance value of  $45 \text{ k}\Omega$  is recommended for oscillator operation with an external RC network. The minimum resistance value to ensure operation is 40 k $\Omega$ . The typical oscillator frequency can be estimated from the values of the resistor (*R* in k $\Omega$ ) and capacitor (*C* in pF) elements using the following equation:

Oscillator Frequency (kHz) =  $\frac{1 \times 10^6}{(0.4 \times R \times C) + (4 \times C)}$ 

Figure 29 displays the typical  $(3.3 \text{ V} \text{ and } 25^{\circ}\text{C})$  oscillator frequency as a function of the capacitor (C, in pF) employed in the RC network assuming a  $45 \text{ K}\Omega$  external resistor. For very small values of C, the parasitic capacitance of the oscillator X<sub>IN</sub> pin and the printed circuit board must be included in the estimation of the oscillator frequency.

It is possible to operate the RC oscillator using only the parasitic capacitance of the package and printed circuit board. To minimize sensitivity to external parasitics, external capacitance values in excess of 20pF are recommended.

# **Internal Precision Oscillator**

The internal precision oscillator (IPO) is designed for use without external components. You can either manually trim the oscillator for a nonstandard frequency or use the automatic factory-trimmed version to achieve a 5.53MHz frequency. IPO features include:

- On-chip RC oscillator that does not require external components
- Output frequency of either 5.53 MHz or 32.8 kHz (contains both a fast and a slow mode)
- Trimmed through Flash option bits with user override
- Elimination of crystals or ceramic resonators in applications where very high timing accuracy is not required

# Operation

An 8-bit trimming register, incorporated into the design, compensates for absolute variation of oscillator frequency. Once trimmed the oscillator frequency is stable and does not require subsequent calibration. Trimming is performed during manufacturing and is not necessary for you to repeat unless a frequency other than 5.53 MHz (fast mode) or 32.8 kHz (slow mode) is required. This trimming is done at +30°C and a supply voltage of 3.3 V, so accuracy of this operating point is optimal.

If not used, the IPO can be disabled by the Oscillator Control Register (see the <u>Oscillator</u> <u>Control Register Definitions section on page 196</u>).

By default, the oscillator frequency is set by the factory trim value stored in the write-protected Flash information page. However, the user code can override these trim values as described in the <u>Trim Bit Address Space</u> section on page 165.

Select one of two frequencies for the oscillator (5.53MHz and 32.8kHz) using the OSC-SEL bits in the the <u>Oscillator Control</u> chapter on page 193.

Mnemonic	Operands	Instruction
BCLR	bit, dst	Bit Clear
BIT	p, bit, dst	Bit Set or Clear
BSET	bit, dst	Bit Set
BSWAP	dst	Bit Swap
CCF	—	Complement Carry Flag
RCF	_	Reset Carry Flag
SCF	—	Set Carry Flag
ТСМ	dst, src	Test Complement Under Mask
ТСМХ	dst, src	Test Complement Under Mask using Extended Addressing
ТМ	dst, src	Test Under Mask
ТМХ	dst, src	Test Under Mask using Extended Addressing

#### Table 121. Bit Manipulation Instructions

#### Table 122. Block Transfer Instructions

Mnemonic	Operands	Instruction
LDCI	dst, src	Load Constant to/from Program Memory and Auto- Increment Addresses
LDEI	dst, src	Load External Data to/from Data Memory and Auto- Increment Addresses

#### Table 123. CPU Control Instructions

Mnemonic	Operands	Instruction
ATM		Atomic Execution
CCF		Complement Carry Flag
DI	_	Disable Interrupts
EI	_	Enable Interrupts
HALT	_	Halt Mode
NOP		No Operation

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 126. Program Control Instructions

#### Table 127. 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

213
-----

Assembly Mnemonic	Symbolic Operation	Address Mode		Oncode(s)	Flags						Fetch	Instr.
		dst	src	(Hex)	С	Ζ	S	۷	D	Н	S	S
AND dst, src	$dst \gets dst \ AND \ src$	r	r	52	-	*	*	0	_	_	2	3
		r	lr	53	-						2	4
		R	R	54	-						3	3
		R	IR	55	-						3	4
		R	IM	56	-						3	3
		IR	IM	57	-						3	4
ANDX dst, src	$dst \leftarrow dst \ AND \ src$	ER	ER	58	_	*	*	0	-	-	4	3
		ER	IM	59	-						4	3
ATM	Block all interrupt and DMA requests during execution of the next 3 instructions			2F	_	_	_	_	_	_	1	2
BCLR bit, dst	dst[bit] ← 0	r		E2	-	_	_	-	_	_	2	2
BIT p, bit, dst	dst[bit] ← p	r		E2	-	-	-	-	-	-	2	2
BRK	Debugger Break			00	-	-	-	-	_	_	1	1
BSET bit, dst	dst[bit] ← 1	r		E2	-	-	-	-	_	_	2	2
BSWAP dst	dst[7:0] ← dst[0:7]	R		D5	Х	*	*	0	-	-	2	2
BTJ p, bit, src, dst	if src[bit] = p PC $\leftarrow$ PC + X		r	F6	-	-	-	-	_	_	3	3
			lr	F7	-						3	4
BTJNZ bit, src, dst	if src[bit] = 1 PC $\leftarrow$ PC + X		r	F6	_	-	-	-	-	-	3	3
			lr	F7							3	4
BTJZ bit, src, dst	if src[bit] = 0 PC $\leftarrow$ PC + X		r	F6	_	-	-		-	-	3	3
			lr	F7							3	4

#### Table 128. eZ8 CPU Instruction Summary (Continued)

Note: Flags Notation:

\* = Value is a function of the result of the operation.

- = Unaffected.

X = Undefined.

0 = Reset to 0.

1 = Set to 1.