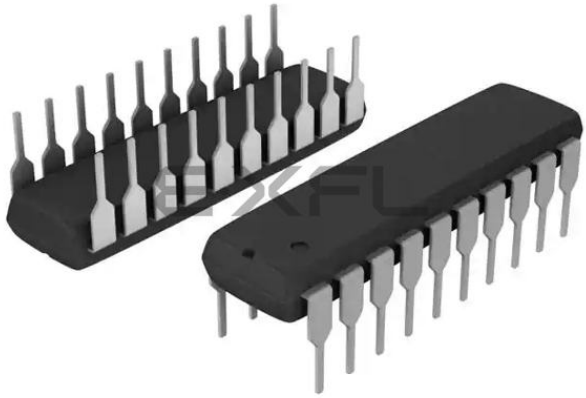


Welcome to E-XFL.COM

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	Obsolete
Core Processor	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	IrDA, UART/USART
Peripherals	Brown-out Detect/Reset, LED, LVD, POR, PWM, WDT
Number of I/O	17
Program Memory Size	2KB (2K x 8)
Program Memory Type	FLASH
EEPROM Size	64 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f021aph020eg

Table 119. Additional Symbols 207
Table 120. Arithmetic Instructions 208
Table 121. Bit Manipulation Instructions 209
Table 122. Block Transfer Instructions 209
Table 123. CPU Control Instructions 209
Table 124. Load Instructions 210
Table 125. Logical Instructions 210
Table 126. Program Control Instructions 211
Table 127. Rotate and Shift Instructions 211
Table 128. eZ8 CPU Instruction Summary 212
Table 129. Opcode Map Abbreviations 223
Table 130. Absolute Maximum Ratings 226
Table 131. DC Characteristics 227
Table 132. Power Consumption 229
Table 133. AC Characteristics 232
Table 134. Internal Precision Oscillator Electrical Characteristics 232
Table 135. Power-On Reset and Voltage Brown-Out Electrical Characteristics and Timing 233
Table 136. Flash Memory Electrical Characteristics and Timing 234
Table 137. Watchdog Timer Electrical Characteristics and Timing 235
Table 138. Non-Volatile Data Storage 235
Table 139. Analog-to-Digital Converter Electrical Characteristics and Timing 236
Table 140. Low Power Operational Amplifier Electrical Characteristics 238
Table 141. Comparator Electrical Characteristics 238
Table 142. Temperature Sensor Electrical Characteristics 239
Table 143. GPIO Port Input Timing 240
Table 144. GPIO Port Output Timing 241
Table 145. On-Chip Debugger Timing 242
Table 146. UART Timing With CTS 243
Table 147. UART Timing Without CTS 244
Table 148. Z8 Encore! XP F082A Series Ordering Matrix 246

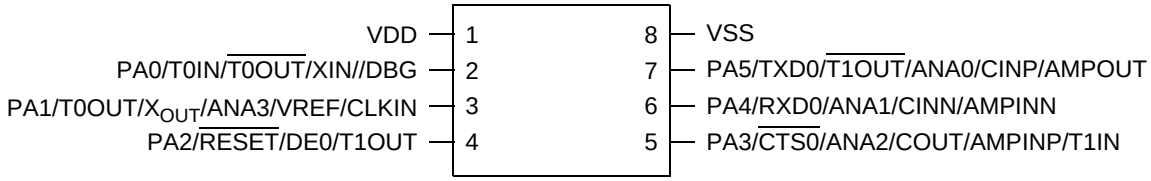


Figure 2. Z8F08xA, Z8F04xA, Z8F02xA and Z8F01xA in 8-Pin SOIC, QFN/MLF-S, or PDIP Package

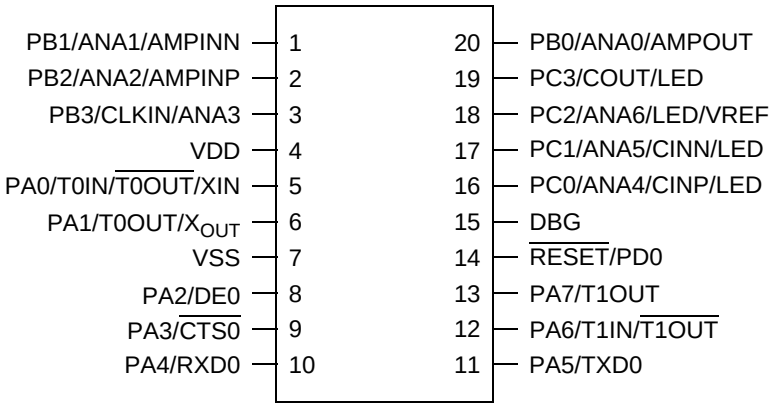


Figure 3. Z8F08xA, Z8F04xA, Z8F02xA and Z8F01xA in 20-Pin SOIC, SSOP or PDIP Package

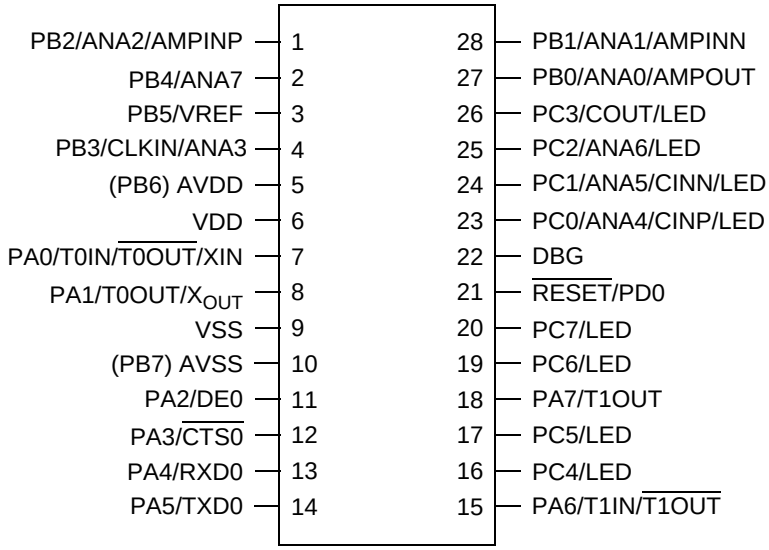


Figure 4. Z8F08xA, Z8F04xA, Z8F02xA and Z8F01xA in 28-Pin SOIC, SSOP or PDIP Package

Reset Sources

Table 9 lists the possible sources of a system reset.

Table 9. Reset Sources and Resulting Reset Type

Operating Mode	Reset Source	Special Conditions
NORMAL or HALT modes	Power-On Reset/Voltage Brown-Out	Reset delay begins after supply voltage exceeds POR level.
	Watchdog Timer time-out when configured for Reset	None.
	$\overline{\text{RESET}}$ pin assertion	All reset pulses less than three system clocks in width are ignored.
	On-Chip Debugger initiated Reset (OCDCTL[0] set to 1)	System Reset, except the On-Chip Debugger is unaffected by the reset.
STOP Mode	Power-On Reset/Voltage Brown-Out	Reset delay begins after supply voltage exceeds POR level.
	$\overline{\text{RESET}}$ pin assertion	All reset pulses less than the specified analog delay are ignored. See Table 131 on page 229.
	DBG pin driven Low	None.

Power-On Reset

Z8 Encore! XP F082A Series devices contain an internal Power-On Reset circuit. The POR circuit monitors the supply voltage and holds the device in the Reset state until the supply voltage reaches a safe operating level. After the supply voltage exceeds the POR voltage threshold (V_{POR}), the device is held in the Reset state until the POR Counter has timed out. If the crystal oscillator is enabled by the option bits, this time-out is longer.

After the Z8 Encore! XP F082A Series device exits the Power-On Reset state, the eZ8 CPU fetches the Reset vector. Following Power-On Reset, the POR status bit in the Reset Status (RSTSTAT) Register is set to 1.

Figure 5 displays Power-On Reset operation. See Electrical Characteristics on page 221 for the POR threshold voltage (V_{POR}).

Low-Power Modes

The Z8 Encore! XP F082A Series products contain power-saving features. The highest level of power reduction is provided by the STOP Mode, in which nearly all device functions are powered down. The next lower level of power reduction is provided by the HALT Mode, in which the CPU is powered down.

Further power savings can be implemented by disabling individual peripheral blocks while in Active mode (defined as being in neither STOP nor HALT Mode).

STOP Mode

Executing the eZ8 CPU's STOP instruction places the device into STOP Mode, powering down all peripherals except the Voltage Brown-Out detector, the Low-power Operational Amplifier and the Watchdog Timer. These three blocks may also be disabled for additional power savings. Specifically, the operating characteristics are:

- Primary crystal oscillator and internal precision oscillator are stopped; X_{IN} and X_{OUT} (if previously enabled) are disabled and PA0/PA1 revert to the states programmed by the GPIO registers
- System clock is stopped
- eZ8 CPU is stopped
- Program counter (PC) stops incrementing
- Watchdog Timer's internal RC oscillator continues to operate if enabled by the Oscillator Control Register
- If enabled, the Watchdog Timer logic continues to operate
- If enabled for operation in STOP Mode by the associated Flash option bit, the Voltage Brown-Out protection circuit continues to operate
- Low-power operational amplifier continues to operate if enabled by the Power Control Register
- All other on-chip peripherals are idle

To minimize current in STOP Mode, all GPIO pins that are configured as digital inputs must be driven to one of the supply rails (V_{CC} or GND). Additionally, any GPIOs configured as outputs must also be driven to one of the supply rails. The device can be brought out of STOP Mode using Stop Mode Recovery. For more information about Stop Mode Recovery, see the Reset, Stop Mode Recovery and Low Voltage Detection chapter on page 22.

HALT Mode

Executing the eZ8 CPU's HALT instruction places the device into HALT Mode, which powers down the CPU but leaves all other peripherals active. In HALT Mode, the operating characteristics are:

- Primary oscillator is enabled and continues to operate
- System clock is enabled and continues to operate
- eZ8 CPU is stopped
- Program counter (PC) stops incrementing
- Watchdog Timer's internal RC oscillator continues to operate
- If enabled, the Watchdog Timer continues to operate
- All other on-chip peripherals continue to operate, if enabled

The eZ8 CPU can be brought out of HALT Mode by any of the following operations:

- Interrupt
- Watchdog Timer time-out (interrupt or reset)
- Power-On Reset
- Voltage Brown-Out reset
- External $\overline{\text{RESET}}$ pin assertion

To minimize current in HALT Mode, all GPIO pins that are configured as inputs must be driven to one of the supply rails (V_{CC} or GND).

Peripheral-Level Power Control

In addition to the STOP and HALT modes, it is possible to disable each peripheral on each of the Z8 Encore! XP F082A Series devices. Disabling a given peripheral minimizes its power consumption.

Power Control Register Definitions

The following sections define the Power Control registers.

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

Watchdog Timer

The Watchdog Timer (WDT) protects against corrupt or unreliable software, power faults and other system-level problems which may place the Z8 Encore! XP F082A Series devices into unsuitable operating states. The features of Watchdog Timer include:

- On-chip RC oscillator
- A selectable time-out response: reset or interrupt
- 24-bit programmable time-out value

Operation

The Watchdog Timer is a one-shot timer that resets or interrupts the Z8 Encore! XP F082A Series devices when the WDT reaches its terminal count. The Watchdog Timer uses a dedicated on-chip RC oscillator as its clock source. The Watchdog Timer operates in only two modes: ON and OFF. Once enabled, it always counts and must be refreshed to prevent a time-out. Perform an enable by executing the WDT instruction or by setting the WDT_AO Flash option bit. The WDT_AO bit forces the Watchdog Timer to operate immediately upon reset, even if a WDT instruction has not been executed.

The Watchdog Timer is a 24-bit reloadable downcounter that uses three 8-bit registers in the eZ8 CPU register space to set the reload value. The nominal WDT time-out period is described by the following equation:

$$\text{WDT Time-out Period (ms)} = \frac{\text{WDT Reload Value}}{10}$$

where the WDT reload value is the decimal value of the 24-bit value given by {WDTU[7:0], WDTH[7:0], WDTL[7:0]} and the typical Watchdog Timer RC oscillator frequency is 10kHz. The Watchdog Timer cannot be refreshed after it reaches 000002H. The WDT reload value must not be set to values below 000004H. Table 58 provides information about approximate time-out delays for the minimum and maximum WDT reload values.

Table 58. Watchdog Timer Approximate Time-Out Delays

WDT Reload Value (Hex)	WDT Reload Value (Decimal)	Approximate Time-Out Delay (with 10kHz typical WDT oscillator frequency)	
		Typical	Description
000004	4	400 µs	Minimum time-out delay
FFFFFF	16,777,215	28 minutes	Maximum time-out delay

Watchdog Timer Reload Upper, High and Low Byte Registers

The Watchdog Timer Reload Upper, High and Low Byte (WDTU, WDTH, WDTL) registers, shown in Tables 60 through 62, form the 24-bit reload value that is loaded into the Watchdog Timer when a WDT instruction executes. The 24-bit reload value ranges across bits [23:0] to encompass the three bytes {WDTU[7:0], WDTH[7:0], WDTL[7:0]}. Writing to these registers sets the appropriate reload value. Reading from these registers returns the current Watchdog Timer count value.

! **Caution:** The 24-bit WDT reload value must not be set to a value less than 000004H.

Table 60. Watchdog Timer Reload Upper Byte Register (WDTU)

Bit	7	6	5	4	3	2	1	0
Field	WDTU							
RESET	00H							
R/W	R/W*							
Address	FF1H							

Note: A read returns the current WDT count value; a write sets the appropriate reload value.

Bit	Description
[7:0] WDTU	**WDT Reload Upper Byte** Most-significant byte (MSB); bits[23:16] of the 24-bit WDT reload value.

Table 61. Watchdog Timer Reload High Byte Register (WDTH)

Bit	7	6	5	4	3	2	1	0
Field	WDTH							
RESET	04H							
R/W	R/W*							
Address	FF2H							

Note: A read returns the current WDT count value; a write sets the appropriate reload value.

Bit	Description
[7:0] WDTH	**WDT Reload High Byte** Middle byte; bits[15:8] of the 24-bit WDT reload value.

Universal Asynchronous Receiver/ Transmitter

The universal asynchronous receiver/transmitter (UART) is a full-duplex communication channel capable of handling asynchronous data transfers. The UART uses a single 8-bit data mode with selectable parity. Features of the UART include:

- 8-bit asynchronous data transfer
- Selectable even- and odd-parity generation and checking
- Option of one or two STOP bits
- Separate transmit and receive interrupts
- Framing, parity, overrun and break detection
- Separate transmit and receive enables
- 16-bit baud rate generator (BRG)
- Selectable MULTIPROCESSOR (9-bit) Mode with three configurable interrupt schemes
- Baud rate generator (BRG) can be configured and used as a basic 16-bit timer
- Driver enable (DE) output for external bus transceivers

Architecture

The UART consists of three primary functional blocks: transmitter, receiver and baud rate generator. The UART's transmitter and receiver function independently, but employ the same baud rate and data format. Figure 10 displays the UART architecture.

- Set or clear the CTSE bit to enable or disable control from the remote receiver using the $\overline{\text{CTS}}$ pin

6. Check the TDRE bit in the UART Status 0 Register to determine if the Transmit Data Register is empty (indicated by a 1). If empty, continue to Step 7. If the Transmit Data Register is full (indicated by a 0), continue to monitor the TDRE bit until the Transmit Data Register becomes available to receive new data.
7. Write the UART Control 1 Register to select the outgoing address bit.
8. Set the Multiprocessor Bit Transmitter (MPBT) if sending an address byte, clear it if sending a data byte.
9. Write the data byte to the UART Transmit Data Register. The transmitter automatically transfers the data to the Transmit Shift Register and transmits the data.
10. Make any changes to the Multiprocessor Bit Transmitter (MPBT) value, if appropriate and MULTIPROCESSOR Mode is enabled.
11. To transmit additional bytes, return to Step 5.

Transmitting Data using the Interrupt-Driven Method

The UART Transmitter interrupt indicates the availability of the Transmit Data Register to accept new data for transmission. Observe the following steps to configure the UART for interrupt-driven data transmission:

1. Write to the UART Baud Rate High and Low Byte registers to set the appropriate baud rate.
2. Enable the UART pin functions by configuring the associated GPIO port pins for alternate function operation.
3. Execute a DI instruction to disable interrupts.
4. Write to the Interrupt control registers to enable the UART Transmitter interrupt and set the acceptable priority.
5. Write to the UART Control 1 Register to enable MULTIPROCESSOR (9-bit) Mode functions, if MULTIPROCESSOR Mode is appropriate.
6. Set the MULTIPROCESSOR Mode Select (MPEN) to Enable MULTIPROCESSOR Mode.
7. Write to the UART Control 0 Register to:
   - Set the transmit enable bit (TEN) to enable the UART for data transmission
   - Enable parity, if appropriate and if MULTIPROCESSOR Mode is not enabled and select either even or odd parity

Table 69. UART Address Compare Register (U0ADDR)

Bit	7	6	5	4	3	2	1	0
Field	COMP_ADDR							
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	F45H							

Bit	Description
[7:0] COMP_ADDR	**Compare Address** This 8-bit value is compared to incoming address bytes.

UART Baud Rate High and Low Byte Registers

The UART Baud Rate High (UxBRH) and Low Byte (UxBRL) registers, shown in Tables 70 and 71, combine to create a 16-bit baud rate divisor value (BRG[15:0]) that sets the data transmission rate (baud rate) of the UART.

Table 70. UART Baud Rate High Byte Register (U0BRH)

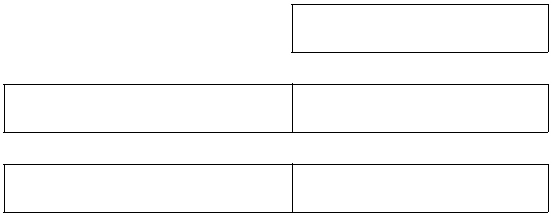
Bit	7	6	5	4	3	2	1	0
Field	BRH							
RESET	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F46H							

Bit	Description
[7:0] BRH	**UART Baud Rate High Byte**

Table 71. UART Baud Rate Low Byte Register (U0BRL)

Bit	7	6	5	4	3	2	1	0
Field	BRL							
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	F47H							

Bit	Description
[7:0] BRL	**UART Baud Rate Low Byte**



Compensation Steps:

1. Correct for Offset:

ADC MSB	ADC LSB

–

Offset MSB	Offset LSB

=

#1 MSB	#1 LSB

2. Compute the absolute value of the offset-corrected ADC value *if negative*; the gain correction factor is computed assuming positive numbers, with sign restoration afterward.

#2 MSB	#2 LSB

Also compute the absolute value of the gain correction word, if negative.

AGain MSB	AGain LSB

3. Multiply by the Gain Correction Word. If operating in DIFFERENTIAL Mode, there are two gain correction values: one for positive ADC values, another for negative ADC values. Use the appropriate Gain Correction Word based on the sign computed by byte #2.

#2 MSB	#2 LSB

*

AGain MSB	AGain LSB

=

Output Data

The output format of the corrected ADC value is shown below.

MSB								LSB							
s	v	b	a	9	8	7	6	5	4	3	2	1	0	–	–

The overflow bit in the corrected output indicates that the computed value was greater than the maximum logical value (+1023) or less than the minimum logical value (–1024). Unlike the hardware overflow bit, this is not a simple binary flag. For a normal (nonoverflow) sample, the sign and the overflow bit match. If the sign bit and overflow bit do not match, a computational overflow has occurred.

Input Buffer Stage

Many applications require the measurement of an input voltage source with a high output impedance. This ADC provides a buffered input for such situations. The drawback of the buffered input is a limitation of the input range. When using unity gain buffered mode, the input signal must be prevented from coming too close to either V_{SS} or V_{DD}. See Table 139 on page 236 for details.

This condition applies only to the input voltage level (with respect to ground) of each differential input signal. The actual differential input voltage magnitude may be less than 300 mV.

The input range of the unbuffered ADC swings from V_{SS} to V_{DD}. Input signals smaller than 300 mV must use the unbuffered input mode. If these signals do not contain low output impedances, they might require off-chip buffering.

Signals outside the allowable input range can be used without instability or device damage. Any ADC readings made outside the input range are subject to greater inaccuracy than specified.

ADC Control Register Definitions

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

ADC Control Register 0 (ADCCTL0): see page 134

ADC Control/Status Register 1 (ADCCTL1): see page 136

ADC Data High Byte Register (ADCD_H): see page 137

ADC Data Low Byte Register (ADCD_L): see page 137

Table 105. Randomized Lot ID Locations (Continued)

Info Page Address	Memory Address	Usage
6A	FE6A	Randomized Lot ID Byte 13.
6B	FE6B	Randomized Lot ID Byte 12.
6D	FE6D	Randomized Lot ID Byte 11.
6E	FE6E	Randomized Lot ID Byte 10.
70	FE70	Randomized Lot ID Byte 9.
71	FE71	Randomized Lot ID Byte 8.
73	FE73	Randomized Lot ID Byte 7.
74	FE74	Randomized Lot ID Byte 6.
76	FE76	Randomized Lot ID Byte 5.
77	FE77	Randomized Lot ID Byte 4.
79	FE79	Randomized Lot ID Byte 3.
7A	FE7A	Randomized Lot ID Byte 2.
7C	FE7C	Randomized Lot ID Byte 1.
7D	FE7D	Randomized Lot ID Byte 0 (least significant).

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

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

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

Read Program Memory CRC (0EH). The Read Program Memory CRC command computes and returns the Cyclic Redundancy Check (CRC) of Program Memory using the 16-bit CRC-CCITT polynomial. If the device is not in DEBUG Mode, this command returns FFFFH for the CRC value. Unlike most other OCD Read commands, there is a delay from issuing of the command until the OCD returns the data. The OCD reads the Program Memory, calculates the CRC value and returns the result. The delay is a function of the Program Memory size and is approximately equal to the system clock period multiplied by the number of bytes in the Program Memory.

```
DBG ← 0EH
DBG → CRC[15:8]
DBG → CRC[7:0]
```

Step Instruction (10H). The Step Instruction command steps one assembly instruction at the current Program Counter (PC) location. If the device is not in DEBUG Mode or the Flash Read Protect Option bit is enabled, the OCD ignores this command.

```
DBG ← 10H
```

Stuff Instruction (11H). The Stuff Instruction command steps one assembly instruction and allows specification of the first byte of the instruction. The remaining 0-4 bytes of the instruction are read from Program Memory. This command is useful for stepping over instructions where the first byte of the instruction has been overwritten by a Breakpoint. If the device is not in DEBUG Mode or the Flash Read Protect Option bit is enabled, the OCD ignores this command.

```
DBG ← 11H
DBG ← opcode[7:0]
```

Execute Instruction (12H). The Execute Instruction command allows sending an entire instruction to be executed to the eZ8 CPU. This command can also step over Breakpoints. The number of bytes to send for the instruction depends on the opcode. If the device is not

Table 118. Notational Shorthand (Continued)

Notation	Description	Operand	Range
Vector	Vector address	Vector	Vector represents a number in the range of 00H to FFH.
X	Indexed	#Index	The register or register pair to be indexed is offset by the signed Index value (#Index) in a +127 to –128 range.

Table 119 lists additional symbols that are used throughout the Instruction Summary and Instruction Set Description sections.

Table 119. Additional Symbols

Symbol	Definition
dst	Destination Operand
src	Source Operand
@	Indirect Address Prefix
SP	Stack Pointer
PC	Program Counter
FLAGS	Flags Register
RP	Register Pointer
#	Immediate Operand Prefix
B	Binary Number Suffix
%	Hexadecimal Number Prefix
H	Hexadecimal Number Suffix

Assignment of a value is indicated by an arrow, as shown in the following example.

dst ← dst + src

This example indicates that the source data is added to the destination data; the result is stored in the destination location.

eZ8 CPU Instruction Classes

eZ8 CPU instructions can be divided functionally into the following groups:

- Arithmetic
- Bit Manipulation

Table 123. CPU Control Instructions (Continued)

Mnemonic	Operands	Instruction
RCF	—	Reset Carry Flag
SCF	—	Set Carry Flag
SRP	src	Set Register Pointer
STOP	—	STOP Mode
WDT	—	Watchdog Timer Refresh

Table 124. Load Instructions

Mnemonic	Operands	Instruction
CLR	dst	Clear
LD	dst, src	Load
LDC	dst, src	Load Constant to/from Program Memory
LDCI	dst, src	Load Constant to/from Program Memory and Auto-Increment Addresses
LDE	dst, src	Load External Data to/from Data Memory
LDEI	dst, src	Load External Data to/from Data Memory and Auto-Increment Addresses
LDWX	dst, src	Load Word using Extended Addressing
LDX	dst, src	Load using Extended Addressing
LEA	dst, X(src)	Load Effective Address
POP	dst	Pop
POPX	dst	Pop using Extended Addressing
PUSH	src	Push
PUSHX	src	Push using Extended Addressing

Table 125. Logical Instructions

Mnemonic	Operands	Instruction
AND	dst, src	Logical AND
ANDX	dst, src	Logical AND using Extended Addressing
COM	dst	Complement
OR	dst, src	Logical OR
ORX	dst, src	Logical OR using Extended Addressing
XOR	dst, src	Logical Exclusive OR
XORX	dst, src	Logical Exclusive OR using Extended Addressing

Table 128. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
AND dst, src	dst ← dst AND src	r	r	52	–	*	*	0	–	–	2	3
		r	Ir	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 ← 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	X	*	*	0	–	–	2	2
BTJ p, bit, src, dst	if src[bit] = p PC ← PC + X		r	F6	–	–	–	–	–	–	3	3
			Ir	F7							3	4
BTJNZ bit, src, dst	if src[bit] = 1 PC ← PC + X		r	F6	–	–	–	–	–	–	3	3
			Ir	F7							3	4
BTJZ bit, src, dst	if src[bit] = 0 PC ← PC + X		r	F6	–	–	–	–	–	–	3	3
			Ir	F7							3	4

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

Table 128. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
TMX dst, src	dst AND src	ER	ER	78	–	*	*	0	–	–	4	3
		ER	IM	79							4	3
TRAP Vector	SP ← SP – 2 @SP ← PC SP ← SP – 1 @SP ← FLAGS PC ← @Vector		Vector	F2	–	–	–	–	–	–	2	6
WDT				5F	–	–	–	–	–	–	1	2
XOR dst, src	dst ← dst XOR src	r	r	B2	–	*	*	0	–	–	2	3
		r	Ir	B3							2	4
		R	R	B4							3	3
		R	IR	B5							3	4
		R	IM	B6							3	3
		IR	IM	B7							3	4
XORX dst, src	dst ← dst XOR src	ER	ER	B8	–	*	*	0	–	–	4	3
		ER	IM	B9							4	3

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

Table 139. Analog-to-Digital Converter Electrical Characteristics and Timing

		V_{DD} = 3.0 V to 3.6 V T_A = 0°C to +70°C (unless otherwise stated)				
Symbol	**Parameter**	**Minimum**	**Typical**	**Maximum**	**Units**	**Conditions**
	Resolution	10		–	bits	
	Differential Nonlinearity (DNL)	–1.0	–	1.0	LSB[3]	External V_{REF} = 2.0 V; $R_S \leftarrow$ 3.0 kΩ
	Integral Nonlinearity (INL)	–3.0	–	3.0	LSB[3]	External V_{REF} = 2.0 V; $R_S \leftarrow$ 3.0 kΩ
	Offset Error with Calibration		±1		LSB[3]	
	Absolute Accuracy with Calibration		±3		LSB[3]	
V_{REF}	Internal Reference Voltage	1.0 2.0	1.1 2.2	1.2 2.4	V	REFSEL=01 REFSEL=10
V_{REF}	Internal Reference Variation with Temperature		±1.0		%	Temperature variation with V_{DD} = 3.0
V_{REF}	Internal Reference Voltage Variation with V_{DD}		±0.5		%	Supply voltage variation with T_A = 30°C
R_{REFOUT}	Reference Buffer Output Impedance		850		W	When the internal reference is buffered and driven out to the VREF pin (REFOUT = 1)
	Single-Shot Conversion Time	–	5129	–	System clock cycles	All measurements but temperature sensor
			10258			Temperature sensor measurement

Notes:
1. Analog source impedance affects the ADC offset voltage (because of pin leakage) and input settling time.
2. Devices are factory calibrated at V_{DD} = 3.3V and T_A = +30°C, so the ADC is maximally accurate under these conditions.
3. LSBs are defined assuming 10-bit resolution.
4. This is the maximum recommended resistance seen by the ADC input pin.
5. The input impedance is inversely proportional to the system clock frequency.

General Purpose I/O Port Output Timing

Figure 35 and Table 144 provide timing information for GPIO port pins.

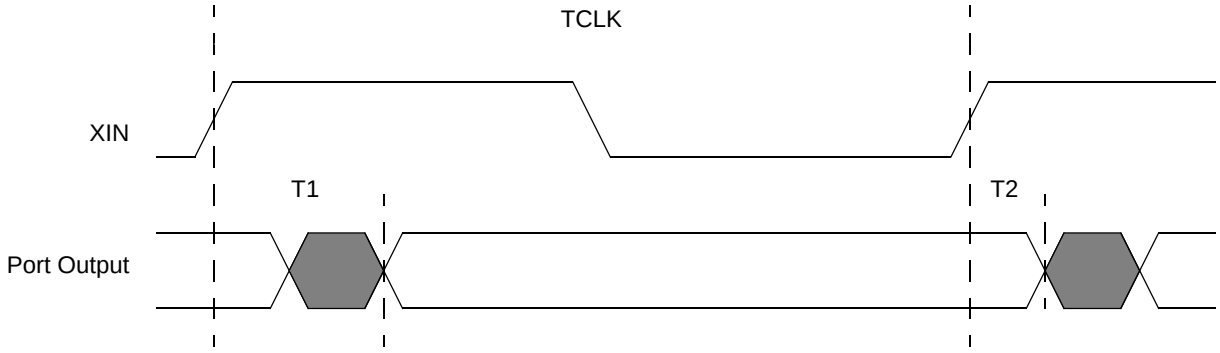


Figure 35. GPIO Port Output Timing

Table 144. GPIO Port Output Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
GPIO port pins			
T_1	X_{IN} Rise to Port Output Valid Delay	–	15
T_2	X_{IN} Rise to Port Output Hold Time	2	–