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	24MHz
Connectivity	DALI, DMX, I²C, LINbus, SPI, UART/USART, USB
Peripherals	DMA, LCD, LVD, POR, PWM, WDT
Number of I/O	67
Program Memory Size	16KB (16K 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 12x12b; D/A 1x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-LQFP
Supplier Device Package	80-LQFP
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f1682at024xk2246

Table 19. Port Alternate Function Mapping (Z8Fxx81 64-Pin Parts) (Continued)

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Subregisters
Port F[1]	PF0	ESOUT[0]	Event System Out 0	N/A
		Reserved		
	PF1	ESOUT[1]	Event System Out 1	
		Reserved		
	PF2	ESOUT[2]	Event System Out 2	
		Reserved		
	PF3	ESOUT[3]	Event System Out 3	
		Reserved		
	PF4	Reserved		
		Reserved		
	PF5	Reserved		
		Reserved		
	PF6	$\overline{SS1}$	SPI 1 Slave Select	
		Reserved		
	PF7	MISO1	SPI 1 Master In Slave Out	
		Reserved		
Port G	PG0	SCK1	SPI 1 Serial Clock	N/A
		Reserved		
	PG1	MOSI1	SPI 1 Master Out Slave In	
		Reserved		
	PG2	Reserved		
		Reserved		
	PG3	Reserved		
		Reserved		
	PG4	Reserved		
		Reserved		

Notes
1. Because there are at most two choices of alternate function for some pins of Ports A, B, D, E and F, the Alternate Function Set register AFS2 is not implemented. Additionally, alternate function selection, as described in the Port A–J Alternate Function Subregisters (see page 88), must also be enabled.
2. The alternate function selection for Port C, as described in the Port A–J Alternate Function Subregisters (see page 88), must also be enabled.

Bit	Description (Continued)
[1:0] FLLNDIVL	**Frequency Locked Loop (FLL) N-Divider Low Byte** 00–11: Least significant bits of the FLL N-divider control word, bits 1:0. To meet minimum and maximum FLL operating frequency requirements, {FLLNDIVH, FLLNDIVL} should not be less than 01Fh (1MHz) nor exceed 2DCh (24MHz). {FLLNDIVH, FLLNDIV} function as follows: 000h–01Eh=Reserved, 01Fh=Multiply PCLK by 31. 020h=Multiply PCLK by 32. 021h=Multiply PCLK by 33. • • • 2DAh: Multiply PCLK by 730. 2DBh: Multiply PCLK by 731. 2DCh: Multiply PCLK by 732. 2FBh–3FFh: Reserved.

8.11.6. Clock Control 5 Register

The Clock Control 5 Register (CLKCTL5), shown in Table 43, enables/disables various clock sources and selects controls the DCO and FLL. Entry into Stop Mode and Stop-Mode Recovery (SMR) affects the state of some bits in this register and leaves others unchanged. Before writing CLKCTL5, the clock control registers must be unlocked as described in the Clock System Control Register Unlocking/Locking section on page 103.

Table 43. Clock Control 5 Register (CLKCTL5)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	FLLIRQE	FLLLL	FLLRDY	FLADONE	DCOEN	SEEDSEL	FLLEN
Reset	0	0	0	0	0	1	0	1
STOP*	no change	no change	0	0	no change	0	no change	0
SMR*	no change	no change	0	0	no change	1	no change	0
R/W	R	R/W	R	R	R	R/W	R/W	R/W
Address	F87h							
Notes: *STOP = Effect of entering Stop Mode; SMR = Effect of Stop-Mode Recovery.								

Bit	Description
[7]	**Reserved** This bit is reserved and must be programmed to 0.

9.4.10. Shared Interrupt Select Register 0

The Shared Interrupt Select 0 (IRQSS0) Register, shown in Table 69, determines the source of the PAD*x*S interrupts. The Shared Interrupt Select Register selects between Port A and alternate sources for the individual interrupts.

Table 69. Shared Interrupt Select Register 0 (IRQSS0)

Bit	7	6	5	4	3	2	1	0
Field	PA7VS	PA6CS	PA5CS	PAD4S	PAD3S	PAD2S	PAD1S	Reserved
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	FCDh							

Bit	Description
[7] PA7VS	**PA7/LVD Selection** 0: PA7 is used for the interrupt for PA7VS interrupt request. 1: The LVD is used for the interrupt for PA7VS interrupt request.
[6] PA6CS	**PA6/Comparator 0 Selection** 0: PA6 is used for the interrupt for PA6CS interrupt request. 1: The Comparator 0 is used for the interrupt for PA6CS interrupt request.
[5] PA5CS	**PA5/Comparator 1 Selection** 0: PA5 is used for the interrupt for PA5CS interrupt request. 1: The Comparator 1 is used for the interrupt for PA5CS interrupt request.
[4:1] PAD*x*S	**PAx/PDx Selection** *x* indicates the specific Port A bit number (4–1). 0: PAx is used for the interrupt for PAx/PDx interrupt request. 1: PDx is used for the interrupt for PAx/PDx interrupt request.
[0]	**Reserved** This bit is reserved and must be programmed to 0.

Observe the following steps to configure a timer for Capture/Compare Mode and initiate the count:

1. Write to the Timer Control 1 Register to:
   - Disable the timer
   - Configure the timer for Capture/Compare Mode
   - Set the prescale value
   - Set the Capture edge (rising or falling) for the Timer Input 0
2. Write to the Timer High and Low Byte registers to set the starting count value (typically `0001h`).
3. Write to the Timer Control 2 Register to choose the timer clock source.
4. Write to the Timer Control 0 Register to set the timer interrupt configuration field TICONFIG.
5. Write to the Timer Reload High and Low Byte registers to set the Compare value.
6. 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.
7. Configure the associated GPIO port pin for the Timer Input alternate function or configure the desired Event System Timer Input 0.
8. Write to the Timer Control 1 Register to enable the timer.
9. Counting begins on the first transition of the Timer Input 0 signal. No interrupt is generated by this first edge.

In Capture/Compare Mode, the elapsed time from timer start to Capture event is calculated using the following equation:

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

10.1.3.13.Demodulation Mode

In Demodulation Mode (TMODE=1100), the timer begins counting on the first external Timer Input 0 transition. The appropriate transition (rising edge or falling edge or both) is set by the TPOL bit in the Timer Control 1 Register and TPOLHI bit in the Timer Control 2 Register. The Timer counts timer clocks up to the 16-bit reload value.

Every subsequent appropriate transition (after the first) of the Timer Input 0 signal captures the current count value. The Capture value is written to the Timer PWM0 High and

Chapter 11. Multi-Channel Timer

The Multi-Channel timer has a 16-bit up/down counter and a 4-channel Capture/Compare/PWM channel array. This timer provides multiple synchronous Capture/Compare/PWM channels based on a single timer. The Multi-Channel Timer features include:

- 16-bit up/down timer counter with programmable prescale
- Selectable clock source (system clock or external input pin)
- Count Modulo and Count up/down counter modes
- Four independent capture/compare channels which reference the common timer
- Channel modes:
  - One-Shot Compare Mode
  - Continuous Compare Mode
  - PWM Output Mode
  - Capture Mode
- Event System and external input pin for timer input
- DMA request source

11.1. Architecture

Figure 21 shows the Multi-Channel Timer architecture.

Table 89. Multi-Channel Timer Address Map (Continued)

Address/ Subaddress	Register/ Subregister Name
FA6	Subregister 1
FA7	Subregister 2
Subregister 0	
0	Timer Control 0
1	Channel Status 0
2	Channel A Capture/Compare High
3	Channel B Capture/Compare High
4	Channel C Capture/Compare High
5	Channel D Capture/Compare High
Subregister 1	
0	Timer Control 1
1	Channel Status 1
2	Channel A Capture/Compare Low
3	Channel B Capture/Compare Low
4	Channel C Capture/Compare Low
5	Channel D Capture/Compare High
Subregister 2	
0	Reserved
1	Reserved
2	Channel A Control
3	Channel B Control
4	Channel C Control
5	Channel D Control

11.7.2. Multi-Channel Timer High and Low Byte Registers

The High and Low Byte (MCTH and MCTL) registers, shown in Tables 90 and 91, contain the current 16-bit MCT count value. Zilog does not recommend writing to the MCT High and Low Byte registers while the MCT is enabled. If either or both of the MCT High or Low Byte registers are written to during counting, the 8-bit written value is placed in the counter (High and/or Low byte) at the next system clock edge. The counter continues counting from the new value.

When MCT is enabled, a read from MCTH causes the value in MCTL to be stored in a temporary holding register. A read from MCTL returns this temporary register when MCT

Collision detection is enabled by setting CLSNE in the DALI Control Register. This setting is useful for DALI masters in systems with more than one master, because it is possible for more than one master to start a transmission at the same time. When CLSNE is set, the UART-LDD monitors its own transmission. Because Low (i.e., 0) is the dominant state on the DALI bus, collision detection effectively checks to determine if High (i.e., 1) state transmissions are not corrupted. If a collision is detected, CLSN is set in the Status Register and an interrupt request is generated.

14.1.11.4. DALI Receive Operation

The UART-LDD Control 0 Register must be initialized. For DALI receive operation, configure REN=1, STOP=1 and all other bits=0. When the MSEL= `100b` (Mode Select) in the DALI Control Register, DALI receive operation can be configured for single-byte or multiple-byte reception. If MULTRXE=0 in the DALI Control Register, single-byte receive is selected, and stop bits will be expected after each transmitted byte. This setting is typically selected for a DALI master that will receive a slave response message. Address match checking is not performed when MULTRXE=0.

If MULTRXE=1 in the DALI Control Register, multiple-byte receive is selected, and stop bits will be received to signal the end of the transmission. This setting is typically selected for DALI slave operation to receive an address byte followed by one or more data bytes.

If PARTRXE is set, and if a partial byte has been received, it will be loaded into the UART-LDD Receive Data Register upon receiving the number of stop bits selected by STOP in the UART-LDD Control 0 Register. Software should determine which bits in the received byte are valid.

When MULTRXE=1 in the DALI Control Register, the start bit is detected as the beginning of a new message. The UART-LDD decodes the first byte received as an address, and a status is provided with MODESTAT in the UART-LDD Mode Select and Status Register, as follows:

0–7Fh. Short address, each DALI slave is assigned a short address (MODESTAT=001).

80–9Fh. Group address (MODESTAT=010).

A0–FDh. Special or unrecognized command (MODESTAT=101).

FE–FFh. Broadcast (MODESTAT=100).

Address matching for short addresses is performed by hardware, which compares the short address in the received address byte to the value of COMP_ADDR[5:0] stored in the Comparison Address Register. If a short address is received that does not match the value of COMP_ADDR[5:0], the message is ignored.

Each DALI slave can belong to as many as 4 groups of the 16 available groups. Software should determine whether the slave belongs to the group for which the message is intended, and whether to process the message.

16.2.6.5. Slave Receive Transaction with 7-Bit Address

The data transfer format for writing data from a master to a slave in 7-bit address mode is shown in Figure 50. The procedure that follows describes the I²C Master/Slave Controller operating as a slave in 7-bit addressing mode and receiving data from the bus master.

S	Slave Address	W=0	A	Data	A	Data	A	Data	A/$\overline{A}$	P/S

Figure 50. Data Transfer Format, Slave Receive Transaction with 7-Bit Address

1. The software configures the controller for operation as a slave in 7-bit addressing mode, as follows:
   a. Initialize the MODE field in the I²C Mode Register for either SLAVE ONLY Mode or MASTER/SLAVE Mode with 7-bit addressing.
   b. Optionally set the GCE bit.
   c. Initialize the SLA[6:0] bits in the I²C Slave Address Register.
   d. Set IEN=1 in the I²C Control Register. Set NAK=0 in the I²C Control Register.

2. The bus master initiates a transfer, sending the address byte. In SLAVE Mode, the I²C controller recognizes its own address and detects that R/$\overline{W}$ bit=0 (written from the master to the slave). The I²C controller acknowledges, indicating it is available to accept the transaction. The SAM bit in the I2CISTAT Register is set to 1, causing an interrupt. The RD bit in the I2CISTAT Register is cleared to 0, indicating a write to the slave. The I²C controller holds the SCL signal Low, waiting for the software to load the first data byte.
3. The software responds to the interrupt by reading the I2CISTAT Register (which clears the SAM bit). After seeing the SAM bit to 1, the software checks the RD bit. Because RD=0, no immediate action is required until the first byte of data is received. If software is only able to accept a single byte, it sets the NAK bit in the I2CCTL Register at this time.
4. The master detects the Acknowledge and sends the byte of data.
5. The I²C controller receives the data byte and responds with an Acknowledge or a Not Acknowledge, depending on the state of the NAK bit in the I2CCTL Register. The I²C controller generates the receive data interrupt by setting the RDRF bit in the I2CISTAT Register.
6. The software responds by reading the I2CISTAT Register, finding the RDRF bit=1 and reading the I2CDATA Register clearing the RDRF bit. If software can accept only one more data byte, it sets the NAK bit in the I2CCTL Register.
7. The master and slave loops through Step 4 to Step 6 until the master detects a Not Acknowledge instruction or runs out of data to send.

- When the DMA interrupt occurs, poll the I2CSTAT Register until the TDRE bit=1. This polling sequence ensures that the I^2C master/slave hardware has commenced transmitting the last byte written by the DMA.
- Set the stop bit in the I2CCTL Register. The stop bit is polled by software to determine when the transaction is actually completed.
- Clear the DMAIF bit in the I2CMODE Register.

Note: If the slave sends a Not Acknowledge prior to the last byte, a Not Acknowledge interrupt occurs. Software must respond to this interrupt by clearing the DMAIF bit and setting the stop bit to end the transaction.

16.2.7.2. Master Read Transaction with Data DMA

In master read transactions, the master is responsible for the Acknowledge for each data byte transferred. The master software must set the NAK bit after the next to the last data byte has been received, or while the last byte is being received. The DMA supports these actions by setting the DMA watermark to 1, which results in a DMA interrupt when the next-to-the-last byte has been received. A DMA interrupt also occurs when the last byte is received. Otherwise, the sequence is similar to the sequence for the master write transaction described in the previous subsection.

- Configure the selected DMA channel for I^2C receive. The IEOB bit must be set in the DMAxCTL0 Register for the last buffer to be transferred. Typically, one buffer is defined with a transfer length of N, in which N bytes are expected to be read from the slave. The watermark is set to 1 by setting WMCNT to 0001 in the DMAxCNTH Register.
- The I^2C interrupt must be enabled in the interrupt controller to alert software of any I^2C error conditions. A Not Acknowledge interrupt occurs on the last byte transferred.
- The I^2C master/slave must be configured as defined in a previous section describing MASTER Mode transactions. The TXI bit in the I2CCTL Register must be cleared.
- Initiate the I^2C transaction as described in the Master Address-Only Transactions section on page 312, using the ACKV and ACK bits in the I2CSTATE Register to determine if the slave acknowledges. Do not set the stop bit unless ACKV=1 and ACK=0 (i.e., slave did not acknowledge).
- Set the DMAIF bit in the I2CMODE Register.
- The DMA transfers the data to memory as it is received from the slave.
- When the first DMA interrupt occurs indicating the (N–1)st byte has been received, the NAK bit must be set in the I2CCTL Register.

16.3.2. I²C Interrupt Status Register

The read-only I²C Interrupt Status Register, shown in Table 155, indicates the cause of any current I²C interrupt and provides the status of the I²C controller. When an interrupt occurs, one or more of the TDRE, RDRF, SAM, ARBLST, SPRS or NCKI bits is set. The GCA and RD bits do not generate an interrupt, but instead provide the status associated with the SAM bit interrupt.

Table 155. I²C Interrupt Status Register (I2CISTAT=F51h)

Bit	7	6	5	4	3	2	1	0
Field	TDRE	RDRF	SAM	GCA	RD	ARBLST	SPRS	NCKI
Reset	1	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Address	F51h							

Bit	Description
[7] TDRE	**Transmit Data Register Empty** When the I²C controller is enabled, this bit is 1 when the I²C Data Register is empty. When set, this bit causes the I²C controller to generate an interrupt, except when the I²C controller is shifting in data during the reception of a byte or when shifting an address and the RD bit is set. This bit clears by writing to the I2CDATA Register.
[6] RDRF	**Receive Data Register Full** This bit is set=1 when the I²C controller is enabled and the I²C controller has received a byte of data. When asserted, this bit causes the I²C controller to generate an interrupt. This bit clears by reading the I2CDATA Register.
[5] SAM	**Slave Address Match** This bit is set=1 if the I²C controller is enabled in SLAVE Mode and an address is received that matches the unique slave address or General Call Address (if enabled by the GCE bit in the I²C Mode Register). In 10-bit addressing mode, this bit is not set until a match is achieved on both address bytes. When this bit is set, the RD and GCA bits are also valid. This bit clears by reading the I2CISTAT Register.
[4] GCA	**General Call Address** This bit is set in SLAVE Mode when the General Call Address or start byte is recognized (in either 7 or 10 bit SLAVE Mode). The GCE bit in the I²C Mode Register must be set to enable recognition of the General Call Address and start byte. This bit clears when IEN=0 and is updated following the first address byte of each SLAVE Mode transaction. A General Call Address is distinguished from a start byte by the value of the RD bit (RD=0 for General Call Address, 1 for start byte).
[3] RD	**Read** This bit indicates the direction of transfer of the data. It is set when the master is reading data from the slave. This bit matches the least-significant bit of the address byte after the start condition occurs (for both MASTER and SLAVE modes). This bit clears when IEN=0, and is updated following the first address byte of each transaction.

17.3.21. USB OUT 0–3 Byte Count Subregisters

The USB OUT 0–3 Byte Count subregisters, shown in Table 190, contain the USB OUT endpoint byte counts.

Table 190. USB OUT 0–3 Byte Count Subregisters (USBOxBC)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	BC						
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Address	If USBSA = 45h, 47h, 49h, 4Bh in the USB Subaddress Register, accessible through the USB Subdata Register							

Bit	Description
[7]	**Reserved** This bit is reserved and must be programmed to 0.
[6:0] BC	**OUT Byte Count** 00–40: BC contains the number of bytes sent during the last OUT transfer from the host to the OUT endpoint x. 41-7F: Reserved.

20.3.4. AES Control Register

The AES Control Register, shown in Table 232, configures the AES for operation.

Table 232. AES Control Register (AESCTL)

Bit	7	6	5	4	3	2	1	0
Field	AUTODIS	Reserved	IRQ	MODE		IVEN	DECRYPT	AESEN
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
Address	FBAh							

Bit	Description
[7] AUTODIS	**Auto-Start Mode Disable** 0: Enable Auto-Start Mode. 1: Disable Auto-Start Mode.
[6]	**Reserved** This bit is reserved and must be programmed to 0.
[5] IRQ	**Interrupt Control** 0: Interrupt on ERROR only. 1: Enables interrupt on DONE (BUSY=0) and ERROR.
[4:3] MODE	**Confidentiality Mode Select** 00: Electronic Codebook (ECB) Mode. 01: Output Feedback (OFB) Mode. 10: Cipher Block (CBC) Mode. 11: Decrypt Key Derivation.
[2] IVEN	**Initialization Vector Enable** 0: Disable writing to the Initialization Vector Register. Writing to the AES Data Register is enabled. 1: Enable writing to the Initialization Vector Register. Writing to the AES Data Register is disabled.
[1] DECRYPT	**Decryption/Encryption Select** 0: Encryption. 1: Decryption.
[0] AESEN	**AES Enable** 0: AES accelerator disabled and AESSTAT Register is reset. The Key and Initialization Vector must be loaded after AES enabled. 1: AES accelerator enabled for operation.

Bit	Description (Continued)
[4:3] INMODE	**Input Mode** 00: Single-ended. 01: Balanced differential. 10: Unbalanced differential with translation buffer. 11: Single-ended with translation buffer.
[2] DFORMAT	**Data Format** 0: Data is unsigned (binary). 1: Data is signed (two's complement). Negative values are sign-extended.
[1] RESOLUT	**ADC Conversion Resolution** 0: 12-bit resolution. 1: 2-pass 14-bit resolution.
[0]	**Reserved** This bit is reserved and must be programmed to 1 which changes the reset value.

21.3.3. ADC Control 2 Register

The ADC Control 2 Register, shown in Table 237, contains control for the ADC prescaler, reference selection and power consumption.

Table 237. ADC Control 2 Register (ADCCTL2)

Bits	7	6	5	4	3	2	1	0
Field	REFSEL		REFLVL		PRESCALE			
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	F72h							

Bit	Description
[7:6] REFSEL	**ADC Positive Voltage Reference Select** If REFSEL = 11 and the DAC is also configured to drive V_{REF}+, the DAC voltage reference buffer selection is used. 00: Internal connection to AV_{DD}. 01: V_{REF+} pin driven by an external source. 10: Buffered VBIAS from the Reference System using an internal connection. 11: Buffered VBIAS from the Reference System drives the V_{REF}+ pin.
[5:4] REFLVL	**VBIAS Level Select** 00: 1.25V. 01: 1.5V. 10: 2.0V. 11: 2.5V.

Bit	Description (Continued)
[5:3] REFSEL	**DAC Positive Voltage Reference Select** If REFSEL = 1xx and the ADC is also configured to drive V_{REF}+, the DAC voltage reference buffer selection is used. 000: Internal connection to AV_{DD}. 001: Reserved. 010: Reserved. 011: V_{REF}+ pin driven by an external source. 100: 1.25V internal voltage reference from the Reference System is buffered and drives the V_{REF}+ pin. 101: 1.5V internal voltage reference from the Reference System is buffered and drives the V_{REF}+ pin. 110: 2.0V internal voltage reference from the Reference System is buffered and drives the V_{REF}+ pin. 111: 2.5V internal voltage reference from the Reference System is buffered and drives the V_{REF}+ pin.
[2] DFORMAT	**Data Format** 0: Data is unsigned (binary). 1: Data is signed (two's complement).
[1] DACTRIG	**DAC Triggering** 0: DAC conversion is triggered by a software or DMA write to the DACD_H register. 1: DAC conversion is triggered by an Event System input. While converting the data, interrupt request and DMA request will be asserted allowing transfer of the next data word to be converted.
[0] JUSTIFY	**Data Register Justification** 0: Data is left-justified. 1: Data is right-justified.

Bit	Description (Continued)
[5] PREFSRC	**Programmable Reference Source Selection** 0: VBIAS is the highest tap of the Programmable Reference. 1: AV_{DD} is the highest tap of the Programmable Reference.
[4:0] PREFLVL	**Programmable Reference Level Selection** 0000 to 1111: Programmable reference level=(PREFSRC selection) * (PREFLVL + 1) ÷ 32.

Bit	Description (Continued)
[2:1] BMODE	**Blinking Mode** BMODE has no effect if DMMODE = 1x. 00: No Blinking. 01: One display segment blinks, the LCD display segment accessed by SEG0, COM0. 10: Up to 4 display segments blink, the LCD display segments accessed by SEG0, COM[3:0]. 11: All segments blink.
[0]	**Reserved** This bit is reserved and must be programmed to 0.

26.3.5. LCD Control 1 Register

The LCDCTL1 Register, shown in Table 271, controls the internal charge pump and bias generators. Writes to this register take effect at the end of the current waveform.

Table 271. LCD Control 1 Register (LCDCTL1)

Bits	7	6	5	4	3	2	1	0
Field	HBDDUR			CPEN	BIASGSEL	CONTRAST		
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	FB5h							

Bit	Description
[7:5] HBDDUR	**Higher Bias Drive Duration** Determines the higher bias drive duration at waveform transitions. BIASGSEL selects the higher-bias drive for the waveform transition. 000: Continuous low bias drive. 001: Higher bias drive for 1 prescaler output clock period, low bias drive otherwise. 010: Higher bias drive for 2 prescaler output clock periods, low bias drive otherwise. 011: Higher bias drive for 3 prescaler output clock periods, low bias drive otherwise. 100: Higher bias drive for 4 prescaler output clock periods, low bias drive otherwise. 101: Higher bias drive for 5 prescaler output clock periods, low bias drive otherwise. 110: Higher bias drive for 6 prescaler output clock periods, low bias drive otherwise. 111: Continuous higher bias drive. Supported only if CPEN=0 as the current consumption of the resistor network in continuous high bias drive exceeds the internal charge pump drive.
[4] CPEN	**Charge Pump Enable** 0: The internal LCD Controller charge pump is disabled. 1: The internal LCD Controller charge pump is enabled and is active in all operating modes including Stop Mode. The internal charge pump output current is selected with BIASGSEL.

Bit	Description (Continued)
[1] TXD	**Transmit Drive** 0: Pin is only driven Low during transmission (Open-Drain). 1: Pin is always driven during transmission.
[0] TXHD	**Transmit High Drive Strength** 0: Pin output driver is low drive strength. 1: Pin output driver is high drive strength.

30.4.4. Baud Reload Register

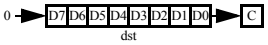
The Baud Reload Register contains the measured auto-baud value.

Table 311. Baud Reload Register

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

Bit	Description
[15:12]	**Reserved** These bits are reserved and must be set to 0000.
[11:0] RELOAD	**Baud Reload Value** This value is the measured Auto-Baud value. Its value can be calculated using the following formula: $\text{RELOAD} = \frac{\text{SYSCLK}}{\text{BAUDRATE}} \times 8$

Table 325. eZ8 CPU Instruction Summary (Continued)

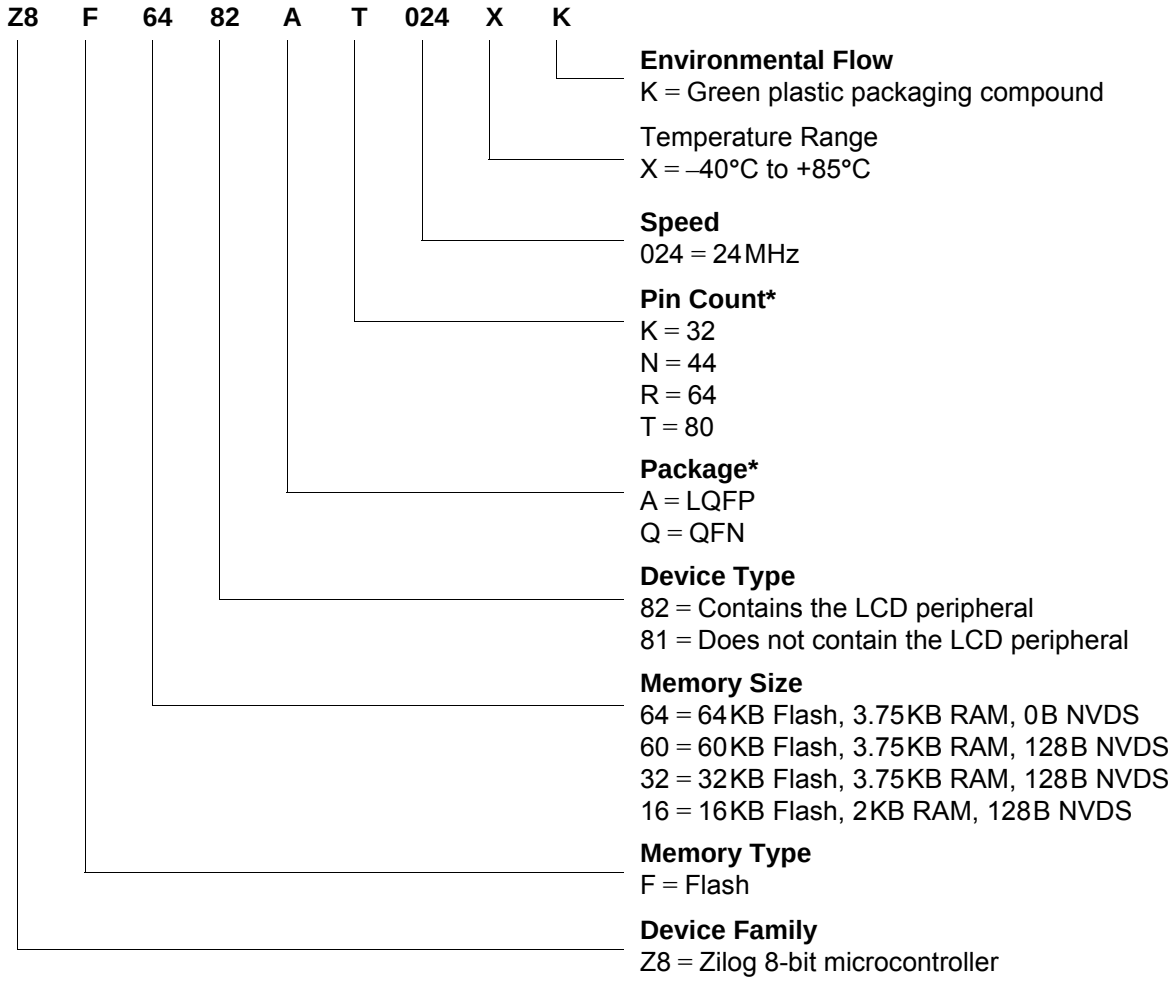
Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Op Code(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
SRA dst		R		D0	*	*	*	0	–	–	2	2
		IR		D1							2	3
SRL dst		R		1F C0	*	*	0	*	–	–	3	2
		IR		1F C1							3	3
SRP src	RP ← src		IM	01	–	–	–	–	–	–	2	2
STOP	Stop Mode			6F	–	–	–	–	–	–	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

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

35.2. Part Number Suffix Designations

Zilog part numbers consist of a number of components, as indicated in the following example.

Example. Part number Z8F6482AT024XK is an 8-bit, 24 MHz Flash microcontroller with 64 KB of Flash memory and containing an LCD peripheral in an 80-pin LQFP package, operating within a –40°C to +85°C temperature range and built using lead-free solder.



Note: * See Table 358 for the combination of package and pin count.