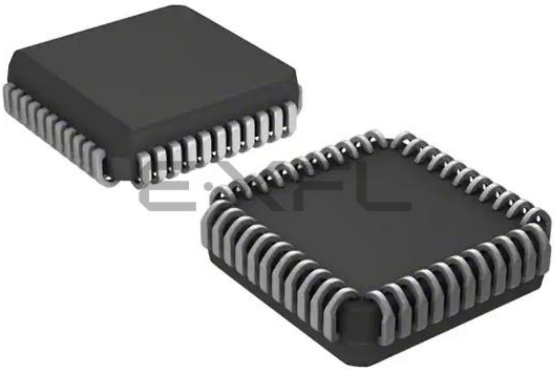
Zilog - Z8F4821VN020SC00TR Datasheet



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	I²C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	31
Program Memory Size	48KB (48K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LCC (J-Lead)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f4821vn020sc00tr

Part Selection Guide

Table 1 identifies the basic features and package styles available for each device within the Z8 Encore! XP product line.

Table 1. Z8 Encore! XP 64K Series Flash Microcontrollers Part Selection Guide

Part Number	Flash (KB)	RAM (KB)	I/O	16-bit Timers with PWM	ADC Inputs	UARTs with IrDA	I^{2}C	SPI	40/44-pin packages	64/68-pin packages	80-pin package
Z8F1621	16	2	31	3	8	2	1	1	X		
Z8F1622	16	2	46	4	12	2	1	1		X	
Z8F2421	24	2	31	3	8	2	1	1	X		
Z8F2422	24	2	46	4	12	2	1	1		X	
Z8F3221	32	2	31	3	8	2	1	1	X		
Z8F3222	32	2	46	4	12	2	1	1		X	
Z8F4821	48	4	31	3	8	2	1	1	X		
Z8F4822	48	4	46	4	12	2	1	1		X	
Z8F4823	48	4	60	4	12	2	1	1			X
Z8F6421	64	4	31	3	8	2	1	1	X		
Z8F6422	64	4	46	4	12	2	1	1		X	
Z8F6423	64	4	60	4	12	2	1	1			X
Die Form Sales	Contact Zilog®										

I^{2}C

The I^{2}C controller makes the Z8 Encore! XP compatible with the I^{2}C protocol. The I^{2}C controller consists of two bidirectional bus lines, a serial data (SDA) line and a serial clock (SCL) line.

Serial Peripheral Interface

The serial peripheral interface allows the Z8 Encore! XP to exchange data between other peripheral devices such as EEPROMs, A/D converters and ISDN devices. The SPI is a full-duplex, synchronous, character-oriented channel that supports a four-wire interface.

Timers

Up to four 16-bit reloadable timers can be used for timing/counting events or for motor control operations. These timers provide a 16-bit programmable reload counter and operate in One-Shot, Continuous, Gated, Capture, Compare, Capture and Compare, and PWM modes. Only 3 timers (Timers 0-2) are available in the 44-pin packages.

Interrupt Controller

The 64K Series products support up to 24 interrupts. These interrupts consist of 12 internal and 12 GPIO pins. The interrupts have 3 levels of programmable interrupt priority.

Reset Controller

The Z8 Encore! can be reset using the $\overline{\text{RESET}}$ pin, Power-On Reset, Watchdog Timer, STOP mode exit, or Voltage Brownout (VBO) warning signal.

On-Chip Debugger

The Z8 Encore! XP features an integrated On-Chip Debugger. The OCD provides a rich set of debugging capabilities, such as reading and writing registers, programming the Flash, setting breakpoints and executing code. A single-pin interface provides communication to the OCD.

DMA Controller

The 64K Series features three channels of DMA. Two of the channels are for register RAM to and from I/O operations. The third channel automatically controls the transfer of data from the ADC to the memory.

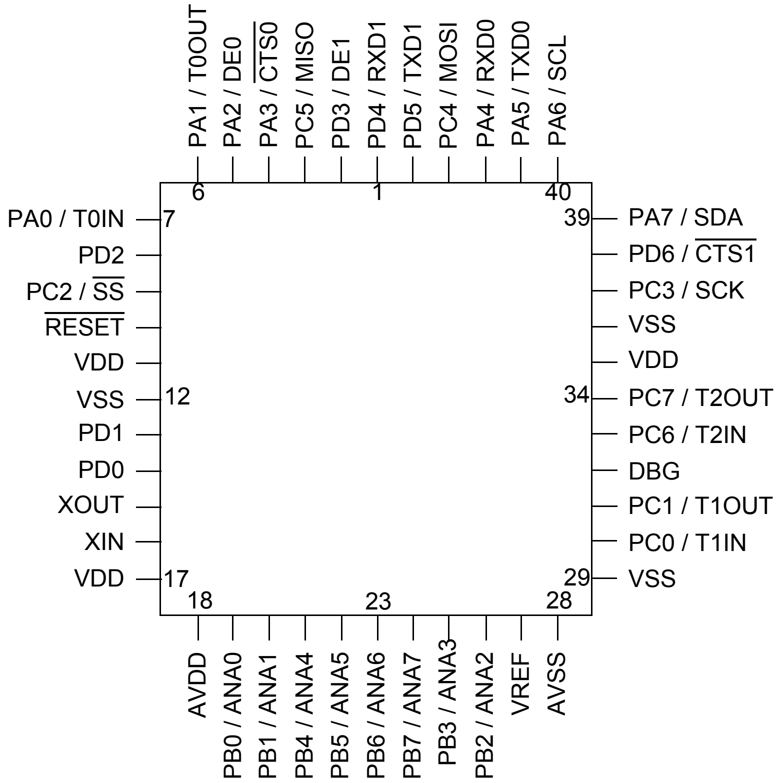


Figure 3. Z8 Encore! XP 64K Series Flash Microcontrollers in 44-Pin Plastic Leaded Chip Carrier (PLCC)

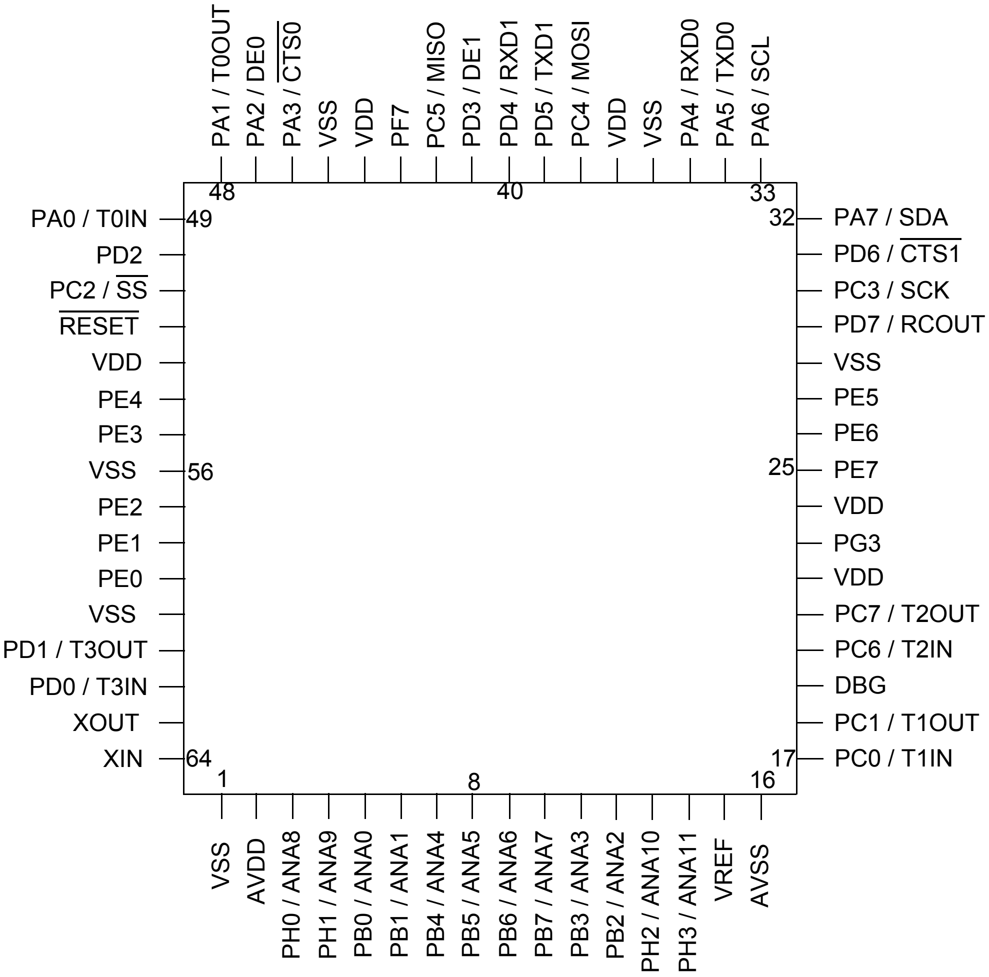


Figure 5. Z8 Encore! XP 64K Series Flash Microcontrollers in 64-Pin Low-Profile Quad Flat Package (LQFP)

Table 5. Z8 Encore! XP 64K Series Flash Microcontrollers Program Memory Maps (Continued)

Program Memory Address (Hex)	Function
0000-0001	Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-BFFF	Program Memory
Z8F642x Products	
0000-0001	Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-FFFF	Program Memory

*See Table 23 on page 68 for a list of the interrupt vectors.

Data Memory

The Z8 Encore! XP 64K Series Flash Microcontrollers does not use the eZ8 CPU's 64 KB Data Memory address space.

Information Area

Table 6 on page 22 describes the Z8 Encore! XP 64K Series Flash Microcontrollers Information Area. This 512 byte Information Area is accessed by setting bit 7 of the Page Select Register to 1. When access is enabled, the Information Area is mapped into the Program Memory and overlays the 512 bytes at addresses FE00H to FFFFH. When the Information Area access is enabled, execution of LDC and LDCI instruction from these Program Memory addresses return the Information Area data rather than the Program Memory data. Reads of these addresses through the On-Chip Debugger also returns the Information Area data. Execution of code from these addresses continues to correctly use the Program Memory. Access to the Information Area is read-only.

Table 7. Z8 Encore! XP 64K Series Flash Microcontrollers Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page No
F61	SPI Control	SPICTL	00	137
F62	SPI Status	SPISTAT	01	139
F63	SPI Mode	SPIMODE	00	140
F64	SPI Diagnostic State	SPIDST	00	141
F65	Reserved	—	XX	
F66	SPI Baud Rate High Byte	SPIBRH	FF	142
F67	SPI Baud Rate Low Byte	SPIBRL	FF	142
F68-F6F	Reserved	—	XX	
Analog-to-Digital Converter				
F70	ADC Control	ADCCTL	20	179
F71	Reserved	—	XX	
F72	ADC Data High Byte	ADCD_H	XX	180
F73	ADC Data Low Bits	ADCD_L	XX	180
F74-FAF	Reserved	—	XX	
DMA 0				
FB0	DMA0 Control	DMA0CTL	00	167
FB1	DMA0 I/O Address	DMA0IO	XX	169
FB2	DMA0 End/Start Address High Nibble	DMA0H	XX	169
FB3	DMA0 Start Address Low Byte	DMA0START	XX	170
FB4	DMA0 End Address Low Byte	DMA0END	XX	170
DMA 1				
FB8	DMA1 Control	DMA1CTL	00	167
FB9	DMA1 I/O Address	DMA1IO	XX	169
FBA	DMA1 End/Start Address High Nibble	DMA1H	XX	169
FBB	DMA1 Start Address Low Byte	DMA1START	XX	170
FBC	DMA1 End Address Low Byte	DMA1END	XX	170
DMA ADC				
FBD	DMA_ADC Address	DMAA_ADDR	XX	171
FBE	DMA_ADC Control	DMAACTL	00	172
FBF	DMA_ADC Status	DMAASTAT	00	173
Interrupt Controller				
FC0	Interrupt Request 0	IRQ0	00	71
FC1	IRQ0 Enable High Bit	IRQ0ENH	00	74
FC2	IRQ0 Enable Low Bit	IRQ0ENL	00	74
FC3	Interrupt Request 1	IRQ1	00	72
FC4	IRQ1 Enable High Bit	IRQ1ENH	00	75
FC5	IRQ1 Enable Low Bit	IRQ1ENL	00	75
FC6	Interrupt Request 2	IRQ2	00	73
FC7	IRQ2 Enable High Bit	IRQ2ENH	00	76
FC8	IRQ2 Enable Low Bit	IRQ2ENL	00	76
FC9-FCC	Reserved	—	XX	

(BRG[15:0]) that sets the data transmission rate (baud rate) of the UART. The UART data rate is calculated using the following equation:

$$\text{UART Data Rate (bits/s)} = \frac{\text{System Clock Frequency (Hz)}}{16 \times \text{UART Baud Rate Divisor Value}}$$

When the UART is disabled, the Baud Rate Generator can function as a basic 16-bit timer with interrupt on time-out. To configure the Baud Rate Generator as a timer with interrupt on time-out, complete the following procedure:

1. Disable the UART by clearing the REN and TEN bits in the UART Control 0 register to 0.
2. Load the desired 16-bit count value into the UART Baud Rate High and Low Byte registers.
3. Enable the Baud Rate Generator timer function and associated interrupt by setting the BRGCTL bit in the UART Control 1 register to 1.

When configured as a general purpose timer, the interrupt interval is calculated using the following equation:

$$\text{Interrupt Interval}(s) = \text{System Clock Period (s)} \times \text{BRG}[15:0]$$

UART Control Register Definitions

The UART control registers support the UART and the associated Infrared Encoder/Decoders. For more information on the infrared operation, see Infrared Encoder/Decoder on page 125.

UART Transmit Data Register

Data bytes written to the UART Transmit Data register (Table 52) are shifted out on the TXD*x* pin. The Write-only UART Transmit Data register shares a Register File address with the Read-only UART Receive Data register.

Table 52. UART Transmit Data Register (U*x*TXD)

BITS	7	6	5	4	3	2	1	0
FIELD	TXD							
RESET	X							
R/W	W							
ADDR	F40H and F48H							

Table 61. UART Baud Rates (Continued)

1.20	868	1.20	0.01	1.20	576	1.20	0.00
0.60	1736	0.60	0.01	0.60	1152	0.60	0.00
0.30	3472	0.30	0.01	0.30	2304	0.30	0.00
10.0 MHz System Clock				**5.5296 MHz System Clock**			
Desired Rate	**BRG Divisor**	**Actual Rate**	**Error**	**Desired Rate**	**BRG Divisor**	**Actual Rate**	**Error**
(kHz)	**(Decimal)**	**(kHz)**	**(%)**	**(kHz)**	**(Decimal)**	**(kHz)**	**(%)**
1250.0	N/A	N/A	N/A	1250.0	N/A	N/A	N/A
625.0	1	625.0	0.00	625.0	N/A	N/A	N/A
250.0	3	208.33	-16.67	250.0	1	345.6	38.24
115.2	5	125.0	8.51	115.2	3	115.2	0.00
57.6	11	56.8	-1.36	57.6	6	57.6	0.00
38.4	16	39.1	1.73	38.4	9	38.4	0.00
19.2	33	18.9	0.16	19.2	18	19.2	0.00
9.60	65	9.62	0.16	9.60	36	9.60	0.00
4.80	130	4.81	0.16	4.80	72	4.80	0.00
2.40	260	2.40	-0.03	2.40	144	2.40	0.00
1.20	521	1.20	-0.03	1.20	288	1.20	0.00
0.60	1042	0.60	-0.03	0.60	576	0.60	0.00
0.30	2083	0.30	0.2	0.30	1152	0.30	0.00
3.579545 MHz System Clock				**1.8432 MHz System Clock**			
Desired Rate	**BRG Divisor**	**Actual Rate**	**Error**	**Desired Rate**	**BRG Divisor**	**Actual Rate**	**Error**
(kHz)	**(Decimal)**	**(kHz)**	**(%)**	**(kHz)**	**(Decimal)**	**(kHz)**	**(%)**
1250.0	N/A	N/A	N/A	1250.0	N/A	N/A	N/A
625.0	N/A	N/A	N/A	625.0	N/A	N/A	N/A
250.0	1	223.72	-10.51	250.0	N/A	N/A	N/A
115.2	2	111.9	-2.90	115.2	1	115.2	0.00
57.6	4	55.9	-2.90	57.6	2	57.6	0.00
38.4	6	37.3	-2.90	38.4	3	38.4	0.00
19.2	12	18.6	-2.90	19.2	6	19.2	0.00

SPI Status Register

The SPI Status register (Table 65) indicates the current state of the SPI. All bits revert to their reset state if the SPIEN bit in the SPICTL register = 0.

Table 65. SPI Status Register (SPISTAT)

BITS	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FIELD	IRQ	OVR	COL	ABT	Reserved		TXST	SLAS
RESET	0							1
R/W	R/W*				R			
ADDR	F62H							
Note: R/W* = Read access. Write a 1 to clear the bit to 0.								

IRQ—Interrupt Request
If SPIEN = 1, this bit is set if the STR bit in the SPICTL register is set, or upon completion of an SPI master or slave transaction. This bit does not set if SPIEN = 0 and the SPI Baud Rate Generator is used as a timer to generate the SPI interrupt.
0 = No SPI interrupt request pending.
1 = SPI interrupt request is pending.

OVR—Overrun
0 = An overrun error has not occurred.
1 = An overrun error has been detected.

COL—Collision
0 = A multi-master collision (mode fault) has not occurred.
1 = A multi-master collision (mode fault) has been detected.

ABT—Slave mode transaction abort
This bit is set if the SPI is configured in slave mode, a transaction is occurring and $\overline{SS}$ deasserts before all bits of a character have been transferred as defined by the NUMBITS field of the SPIMODE register. The IRQ bit also sets, indicating the transaction has completed.
0 = A slave mode transaction abort has not occurred.
1 = A slave mode transaction abort has been detected.

Reserved—Must be 0.

TXST—Transmit Status
0 = No data transmission currently in progress.
1 = Data transmission currently in progress.

SLAS—Slave Select
If SPI enabled as a Slave,

16. If the I^2C slave sends an acknowledge by pulling the SDA signal low during the next high period of SCL, the I^2C Controller sets the ACK bit in the I^2C Status register. Continue with step 17.

    If the slave does not acknowledge the second address byte or one of the data bytes, the I^2C Controller sets the NCKI bit and clears the ACK bit in the I^2C Status register. Software responds to the Not Acknowledge interrupt by setting the STOP and FLUSH bits and clearing the TXI bit. The I^2C Controller sends the STOP condition on the bus and clears the STOP and NCKI bits. The transaction is complete (ignore the following steps).
17. The I^2C Controller shifts the data out by the SDA signal. After the first bit is sent, the Transmit interrupt is asserted.
18. If more bytes remain to be sent, return to step 14.
19. If the last byte is currently being sent, software sets the STOP bit of the I^2C Control register (or START bit to initiate a new transaction). In the STOP case, software also clears the TXI bit of the I^2C Control register at the same time.
20. The I^2C Controller completes transmission of the last data byte on the SDA signal.
21. The slave may either Acknowledge or Not Acknowledge the last byte. Because either the STOP or START bit is already set, the NCKI interrupt does not occur.
22. The I^2C Controller sends the STOP (or RESTART) condition to the I^2C bus and clears the STOP (or START) bit.

Read Transaction with a 7-Bit Address

Figure 32 displays the data transfer format for a read operation to a 7-bit addressed slave. The shaded regions indicate data transferred from the I^2C Controller to slaves and unshaded regions indicate data transferred from the slaves to the I^2C Controller.

S	Slave Address	R = 1	A	Data	A	Data	$\overline{\text{A}}$	P/S

Figure 32. Receive Data Transfer Format for a 7-Bit Addressed Slave

Follow the steps below for a read operation to a 7-bit addressed slave:

1. Software writes the I^2C Data register with a 7-bit slave address plus the read bit (=1).
2. Software asserts the START bit of the I^2C Control register.
3. If this is a single byte transfer, Software asserts the NAK bit of the I^2C Control register so that after the first byte of data has been read by the I^2C Controller, a Not Acknowledge is sent to the I^2C slave.

DMA_ADC Address Register

The DMA_ADC Address register (Table 83) points to a block of the Register File to store ADC conversion values as displayed in Table 82. This register contains the seven most-significant bits of the 12-bit Register File addresses. The five least-significant bits are calculated from the ADC Analog Input number (5-bit base address is equal to twice the ADC Analog Input number). The 10-bit ADC conversion data is stored as two bytes with the most significant byte of the ADC data stored at the even numbered Register File address.

Table 82 provides an example of the Register File addresses if the DMA_ADC Address register contains the value 72H.

Table 82. DMA_ADC Register File Address Example

ADC Analog Input	Register File Address (Hex)[1]
0	720H-721H
1	722H-723H
2	724H-725H
3	726H-727H
4	728H-729H
5	72AH-72BH
6	72CH-72DH
7	72EH-72FH
8	730H-731H
9	732H-733H
10	734H-735H
11	736H-737H

[1] DMAA_ADDR set to 72H.

Table 83. DMA_ADC Address Register (DMAA_ADDR)

BITS	7	6	5	4	3	2	1	0
FIELD	DMAA_ADDR							Reserved
RESET	X							
R/W	R/W							
ADDR	FBDH							

DMAA_ADDR—DMA_ADC Address
These bits specify the seven most-significant bits of the 12-bit Register File addresses used for storing the ADC output data. The ADC Analog Input Number defines the five least-significant bits of the Register File address. Full 12-bit address is {DMAA_ADDR[7:1], 4-bit ADC Analog Input Number, 0}.

Reserved
This bit is reserved and must be 0.

DMA_ADC Control Register

The DMA_ADC Control register (Table 84 on page 172) enables and sets options (DMA enable and interrupt enable) for ADC operation.

Table 84. DMA_ADC Control Register (DMAACTL)

BITS	7	6	5	4	3	2	1	0
FIELD	DAEN	IRQEN	Reserved		ADC_IN			
RESET	0							
R/W	R/W							
ADDR	FBEH							

DAEN—DMA_ADC Enable
0 = DMA_ADC is disabled and the ADC Analog Input Number (ADC_IN) is reset to 0.
1 = DMA_ADC is enabled.

IRQEN—Interrupt Enable
0 = DMA_ADC does not generate any interrupts.
1 = DMA_ADC generates an interrupt after transferring data from the last ADC Analog Input specified by the ADC_IN field.

Reserved
These bits are reserved and must be 0.

ADC_IN—ADC Analog Input Number
These bits set the number of ADC Analog Inputs to be used in the continuous update (data conversion followed by DMA data transfer). The conversion always begins with ADC Analog Input 0 and then progresses sequentially through the other selected ADC Analog Inputs.
0000 = ADC Analog Input 0 updated.
0001 = ADC Analog Inputs 0-1 updated.
0010 = ADC Analog Inputs 0-2 updated.
0011 = ADC Analog Inputs 0-3 updated.
0100 = ADC Analog Inputs 0-4 updated.

0100 = ANA4
0101 = ANA5
0110 = ANA6
0111 = ANA7
1000 = ANA8
1001 = ANA9
1010 = ANA10
1011 = ANA11
11XX = Reserved.

ADC Data High Byte Register

The ADC Data High Byte register (Table 87) contains the upper eight bits of the 10-bit ADC output. During a single-shot conversion, this value is invalid. Access to the ADC Data High Byte register is read-only. The full 10-bit ADC result is given by {ADCD_H[7:0], ADCD_L[7:6]}. Reading the ADC Data High Byte register latches data in the ADC Low Bits register.

Table 87. ADC Data High Byte Register (ADCD_H)

BITS	7	6	5	4	3	2	1	0
FIELD	ADCD_H							
RESET	X							
R/W	R							
ADDR	F72H							

ADCD_H—ADC Data High Byte
This byte contains the upper eight bits of the 10-bit ADC output. These bits are not valid during a single-shot conversion. During a continuous conversion, the last conversion output is held in this register. These bits are undefined after a Reset.

ADC Data Low Bits Register

The ADC Data Low Bits register (Table 88) contains the lower two bits of the conversion value. The data in the ADC Data Low Bits register is latched each time the ADC Data High Byte register is read. Reading this register always returns the lower two bits of the conversion last read into the ADC High Byte register. Access to the ADC Data Low Bits register is read-only. The full 10-bit ADC result is given by {ADCD_H[7:0], ADCD_L[7:6]}.

Electrical Characteristics

Absolute Maximum Ratings

Stresses greater than those listed in Table 105 may cause permanent damage to the device. These ratings are stress ratings only. Operation of the device at any condition outside those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. For improved reliability, unused inputs must be tied to one of the supply voltages (V_{DD} or V_{SS}).

Table 105. Absolute Maximum Ratings

Parameter	Minimum	Maximum	Units	Notes
Ambient temperature under bias	-40	+125	C	
Storage temperature	-65	+150	C	
Voltage on any pin with respect to V_{SS}	-0.3	+5.5	V	1
Voltage on V_{DD} pin with respect to V_{SS}	-0.3	+3.6	V	
Maximum current on input and/or inactive output pin	-5	+5	µA	
Maximum output current from active output pin	-25	+25	mA	
80-Pin QFP Maximum Ratings at –40 °C to 70 °C				
Total power dissipation		550	mW	
Maximum current into V_{DD} or out of V_{SS}		150	mA	
80-Pin QFP Maximum Ratings at 70 °C to 125 °C				
Total power dissipation		200	mW	
Maximum current into V_{DD} or out of V_{SS}		56	mA	
68-Pin PLCC Maximum Ratings at –40 °C to 70 °C				
Total power dissipation		1000	mW	
Maximum current into V_{DD} or out of V_{SS}		275	mA	
68-Pin PLCC Maximum Ratings at 70 °C to 125 °C				
Total power dissipation		500	mW	

Figure 44 displays the maximum active mode current consumption across the full operating temperature range of the device and versus the system clock frequency. All GPIO pins are configured as outputs and driven High.

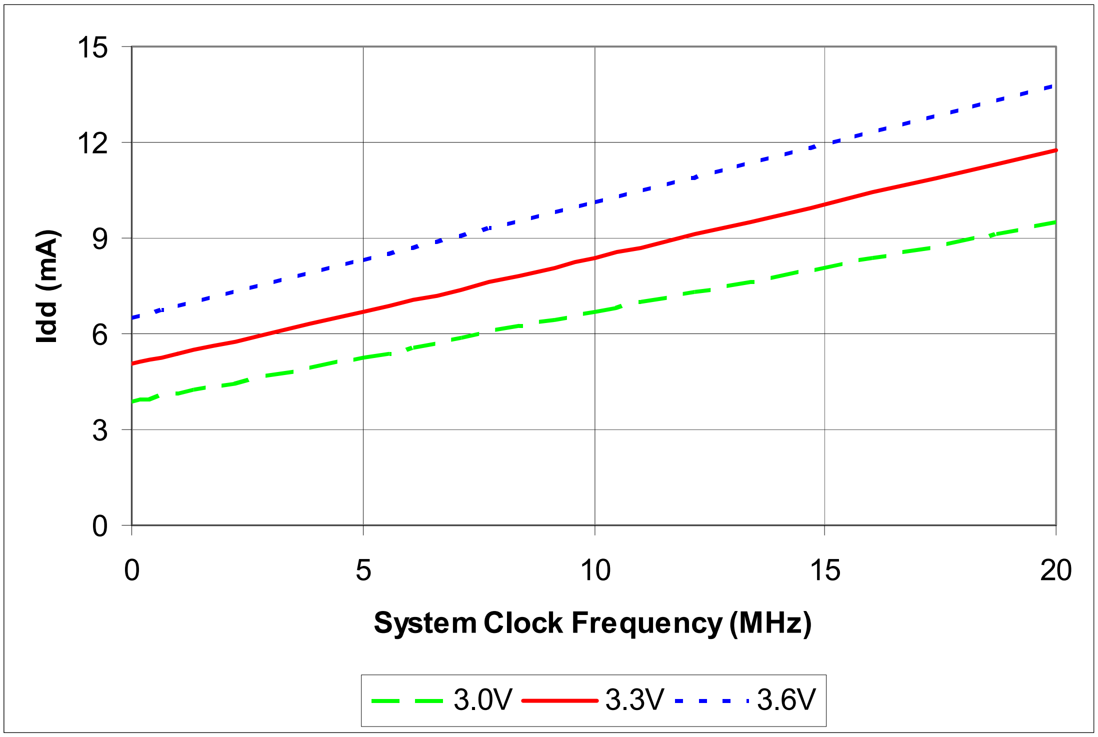


Figure 44. Maximum Active Mode Idd Versus System Clock Frequency

UART Timing

Figure 56 and Table 120 provide timing information for UART pins for the case where the Clear To Send input pin ($\overline{\text{CTS}}$) is used for flow control. In this example, it is assumed that the Driver Enable polarity has been configured to be Active Low and is represented here by $\overline{\text{DE}}$. The $\overline{\text{CTS}}$ to $\overline{\text{DE}}$ assertion delay (T1) assumes the UART Transmit Data register has been loaded with data prior to $\overline{\text{CTS}}$ assertion.

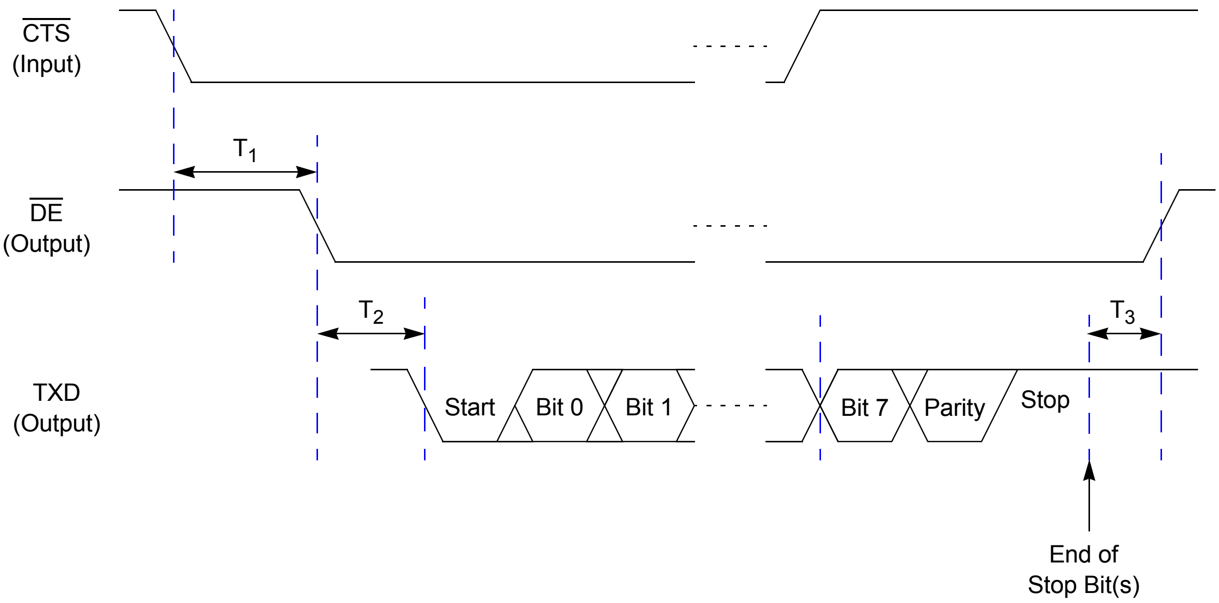


Figure 56. UART Timing with $\overline{\text{CTS}}$

Table 120. UART Timing with $\overline{\text{CTS}}$

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
T_1	$\overline{\text{CTS}}$ Fall to $\overline{\text{DE}}$ Assertion Delay	2 * XIN period	2 * XIN period + 1 Bit period
T_2	$\overline{\text{DE}}$ Assertion to TXD Falling Edge (Start) Delay	1 Bit period	1 Bit period + 1 * XIN period
T_3	End of Stop Bit(s) to $\overline{\text{DE}}$ Deassertion Delay	1 * XIN period	2 * XIN period

; value 01H, is the source. The value 01H is written into the
; Register at address 234H.

Assembly Language Syntax

For proper instruction execution, eZ8 CPU assembly language syntax requires that the operands be written as 'destination, source'. After assembly, the object code usually has the operands in the order 'source, destination', but ordering is opcode-dependent. The following instruction examples illustrate the format of some basic assembly instructions and the resulting object code produced by the assembler. This binary format must be followed if you prefer manual program coding or intend to implement your own assembler.

Example 1: If the contents of Registers 43H and 08H are added and the result is stored in 43H, the assembly syntax and resulting object code is:

Assembly Language Syntax Example 1

Assembly Language Code	ADD	43H,	08H	(ADD dst, src)
Object Code	04	08	43	(OPC src, dst)

Example 2: In general, when an instruction format requires an 8-bit register address, that address can specify any register location in the range 0–255 or, using Escaped Mode Addressing, a Working Register R0 - R15. If the contents of Register 43H and Working Register R8 are added and the result is stored in 43H, the assembly syntax and resulting object code is:

Assembly Language Syntax Example 2

Assembly Language Code	ADD	43H,	R8	(ADD dst, src)
Object Code	04	E8	43	(OPC src, dst)

Refer to the device-specific Product Specification to determine the exact register file range available. The register file size varies, depending on the device type.

eZ8 CPU Instruction Notation

In the eZ8 CPU Instruction Summary and Description sections, the operands, condition codes, status Flags, and address modes are represented by a notational shorthand that is described in Table 122.

Table 130. Logical Instructions (Continued)

Mnemonic	Operands	Instruction
XOR	dst, src	Logical Exclusive OR
XORX	dst, src	Logical Exclusive OR using Extended Addressing

Table 131. 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 132. 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