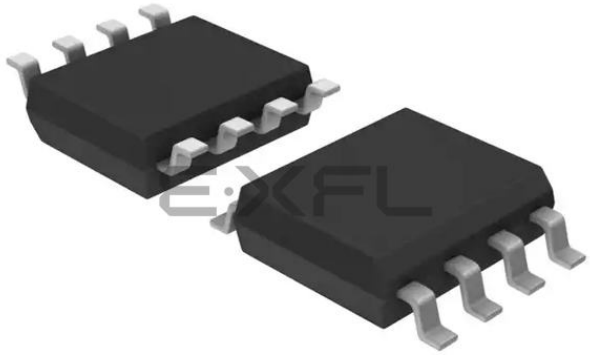


Welcome to E-XFL.COM

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	5MHz
Connectivity	IrDA, UART/USART
Peripherals	Brown-out Detect/Reset, LED, POR, PWM, WDT
Number of I/O	6
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	8-SOIC (0.154", 3.90mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f0413sb005ec

Table 8. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page No
FF2	Watchdog Timer Reload High Byte	WDTH	FF	91
FF3	Watchdog Timer Reload Low Byte	WDTL	FF	91
FF4–FF5	Reserved	—	XX	
Trim Bit Control				
FF6	Trim Bit Address	TRMADR	00	143
FF7	Trim Data	TRMDR	XX	144
Flash Memory Controller				
FF8	Flash Control	FCTL	00	137
FF8	Flash Status	FSTAT	00	137
FF9	Flash Page Select	FPS	00	138
	Flash Sector Protect	FPROT	00	139
FFA	Flash Programming Frequency High Byte	FFREQH	00	140
FFB	Flash Programming Frequency Low Byte	FFREQL	00	140
eZ8 CPU				
FFC	Flags	—	XX	Refer to *eZ8 CPU Core User Manual (UM0128)*
FFD	Register Pointer	RP	XX	
FFE	Stack Pointer High Byte	SPH	XX	
FFF	Stack Pointer Low Byte	SPL	XX	

XX=Undefined

vector address. Following Stop Mode Recovery, the STOP bit in the Watchdog Timer Control Register is set to 1. Table 11 lists the Stop Mode Recovery sources and resulting actions. The section following the table provides more detailed information on each of the Stop Mode Recovery sources.

Table 11. Stop Mode Recovery Sources and Resulting Action

Operating Mode	Stop Mode Recovery Source	Action
STOP mode	Watchdog Timer time-out when configured for Reset	Stop Mode Recovery
	Watchdog Timer time-out when configured for interrupt	Stop Mode Recovery followed by interrupt (if interrupts are enabled)
	Data transition on any GPIO port pin enabled as a Stop Mode Recovery source	Stop Mode Recovery
	Assertion of external $\overline{\text{RESET}}$ Pin	System Reset
	Debug Pin driven Low	System Reset

Stop Mode Recovery Using Watchdog Timer Time-Out

If the Watchdog Timer times out during STOP mode, the device undergoes a Stop Mode Recovery sequence. In the Watchdog Timer Control register, the WDT and STOP bits are set to 1. If the Watchdog Timer is configured to generate an interrupt upon time-out and Z8 Encore! XP® F0823 Series device is configured to respond to interrupts, the eZ8 CPU services the Watchdog Timer interrupt request following the normal Stop Mode Recovery sequence.

Stop Mode Recovery Using a GPIO Port Pin Transition

Each of the GPIO port pins can be configured as a Stop Mode Recovery input source. On any GPIO pin enabled as a Stop Mode Recovery source, a change in the input pin value (from High to Low or from Low to High) initiates Stop Mode Recovery.

Note: *The SMR pulses shorter than specified does not trigger a recovery. When this happens, the STOP bit in the Reset Status (RSTSTAT) register is set to 1.*

Caution: *In STOP mode, the GPIO Port Input Data registers (PxIN) are disabled. The Port Input Data registers record the port transition only if the signal stays on the port pin through the end of the Stop Mode Recovery delay. As a result, short pulses on the port pin can initiate Stop Mode Recovery without being written to the Port Input Data register or without initiating an interrupt (if enabled for that pin).*

Interrupt Request 1 Register

The Interrupt Request 1 (IRQ1) register (Table 35) stores interrupt requests for both vectored and polled interrupts. When a request is presented to the interrupt controller, the corresponding bit in the IRQ1 register becomes 1. If interrupts are globally enabled (vectored interrupts), the interrupt controller passes an interrupt request to the eZ8 CPU. If interrupts are globally disabled (polled interrupts), the eZ8 CPU reads the Interrupt Request 1 register to determine if any interrupt requests are pending.

Table 35. Interrupt Request 1 Register (IRQ1)

BITS	7	6	5	4	3	2	1	0
FIELD	PA7VI	PA6CI	PA5I	PA4I	PA3I	PA2I	PA1I	PA0I
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
ADDR	FC3H							

PA7VI—Port A7 Interrupt Request
0 = No interrupt request is pending for GPIO Port A
1 = An interrupt request from GPIO Port A

PA6CI—Port A6 or Comparator Interrupt Request
0 = No interrupt request is pending for GPIO Port A or Comparator
1 = An interrupt request from GPIO Port A or Comparator

PA*x*I—Port A Pin *x* Interrupt Request
0 = No interrupt request is pending for GPIO Port A pin *x*
1 = An interrupt request from GPIO Port A pin *x* is awaiting service

where *x* indicates the specific GPIO Port pin number (0–5)

Interrupt Request 2 Register

The Interrupt Request 2 (IRQ2) register (Table 36) stores interrupt requests for both vectored and polled interrupts. When a request is presented to the interrupt controller, the corresponding bit in the IRQ2 register becomes 1. If interrupts are globally enabled (vectored interrupts), the interrupt controller passes an interrupt request to the eZ8 CPU. If interrupts are globally disabled (polled interrupts), the eZ8 CPU can read the Interrupt Request 2 register to determine if any interrupt requests are pending.

Table 36. Interrupt Request 2 Register (IRQ2)

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved				PC3I	PC2I	PC1I	PC0I
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
ADDR	FC6H							

Reserved—Must be 0

PC*x*I—Port C Pin *x* Interrupt Request
0 = No interrupt request is pending for GPIO Port C pin *x*
1 = An interrupt request from GPIO Port C pin *x* is awaiting service

where *x* indicates the specific GPIO Port C pin number (0–3)

IRQ0 Enable High and Low Bit Registers

Table 37 describes the priority control for IRQ0. The IRQ0 Enable High and Low Bit registers (Table 38 and Table 39) form a priority encoded enabling for interrupts in the Interrupt Request 0 register. Priority is generated by setting bits in each register.

Table 37. IRQ0 Enable and Priority Encoding

IRQ0ENH[*x*]	IRQ0ENL[*x*]	Priority	Description
0	0	Disabled	Disabled
0	1	Level 1	Low
1	0	Level 2	Nominal
1	1	Level 3	High

where *x* indicates the register bits from 0–7.

Table 38. IRQ0 Enable High Bit Register (IRQ0ENH)

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved	T1ENH	T0ENH	U0RENH	U0TENH	Reserved	Reserved	ADCENH
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
ADDR	FC1H							

Table 46. Interrupt Edge Select Register (IRQES)

BITS	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
ADDR	FCDH							

IES*x*—Interrupt Edge Select *x*
0 = An interrupt request is generated on the falling edge of the PA*x* input or PDx
1 = An interrupt request is generated on the rising edge of the PA*x* input PDx
where *x* indicates the specific GPIO port pin number (0 through 7)

Shared Interrupt Select Register

The Shared Interrupt Select (IRQSS) register (Table 47) determines the source of the PADxS interrupts. The Shared Interrupt Select register selects between Port A and alternate sources for the individual interrupts.

Because these shared interrupts are edge-triggered, it is possible to generate an interrupt just by switching from one shared source to another. For this reason, an interrupt must be disabled before switching between sources.

Table 47. Shared Interrupt Select Register (IRQSS)

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved	PA6CS	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
ADDR	FCEH							

PA6CS—PA6/Comparator Selection
0 = PA6 is used for the interrupt for PA6CS interrupt request
1 = The Comparator is used for the interrupt for PA6CS interrupt request

Reserved—Must be 0

Interrupt Control Register

The Interrupt Control (IRQCTL) register (Table 48) contains the master enable bit for all interrupts.

PWM SINGLE OUTPUT Mode

In PWM SINGLE OUTPUT mode, the timer outputs a PWM output signal through a GPIO port pin. The timer input is the system clock. The timer first counts up to the 16-bit PWM match value stored in the Timer PWM High and Low Byte registers. When the timer count value matches the PWM value, the Timer Output toggles. The timer continues counting until it reaches the Reload value stored in the Timer Reload High and Low Byte registers. Upon reaching the Reload value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes.

If the TPOL bit in the Timer Control register is set to 1, the Timer Output signal begins as a High (1) and transitions to a Low (0) when the timer value matches the PWM value. The Timer Output signal returns to a High (1) after the timer reaches the Reload value and is reset to 0001H.

If the TPOL bit in the Timer Control register is set to 0, the Timer Output signal begins as a Low (0) and transitions to a High (1) when the timer value matches the PWM value. The Timer Output signal returns to a Low (0) after the timer reaches the Reload value and is reset to 0001H.

Follow the steps below for configuring a timer for PWM Single Output mode and initiating the PWM operation:

1. Write to the Timer Control register to:
   - Disable the timer
   - Configure the timer for PWM mode
   - Set the prescale value
   - Set the initial logic level (High or Low) and PWM High/Low transition for the Timer Output alternate function
2. Write to the Timer High and Low Byte registers to set the starting count value (typically 0001H). This only affects the first pass in PWM mode. After the first timer reset in PWM mode, counting always begins at the reset value of 0001H.
3. Write to the PWM High and Low Byte registers to set the PWM value.
4. Write to the Timer Reload High and Low Byte registers to set the Reload value (PWM period). The Reload value must be greater than the PWM value.
5. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
6. Configure the associated GPIO port pin for the Timer Output alternate function.
7. Write to the Timer Control register to enable the timer and initiate counting.

The PWM period is represented by the following equation:

$$\text{PWM Period (s)} = \frac{\text{Reload Value} \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

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

Table 51. Timer 0–1 Reload High Byte Register (TxRH)

BITS	7	6	5	4	3	2	1	0
FIELD	TRH							
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
ADDR	F02H, F0AH							

Table 52. Timer 0–1 Reload Low Byte Register (TxRL)

BITS	7	6	5	4	3	2	1	0
FIELD	TRL							
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
ADDR	F03H, F0BH							

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

Timer 0-1 PWM High and Low Byte Registers

The Timer 0-1 PWM High and Low Byte (TxPWMH and TxPWML) registers (Table 53 and Table 54) control pulse-width modulator (PWM) operations. These registers also store the Capture values for the CAPTURE and CAPTURE/COMPARE modes.

Table 53. Timer 0–1 PWM High Byte Register (TxPWMH)

BITS	7	6	5	4	3	2	1	0
FIELD	PWMH							
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
ADDR	F04H, F0CH							

Watchdog Timer Reload Byte registers (WDTU, WDTH, and WDTL) to allow changes to the time-out period. These write operations to the WDTCTL register address produce no effect on the bits in the WDTCTL register. The locking mechanism prevents spurious writes to the Reload registers.

This register address is shared with the read-only Reset Status Register.

Table 58. Watchdog Timer Control Register (WDTCTL)

BITS	7	6	5	4	3	2	1	0
FIELD	WDTUNLK							
RESET	X	X	X	X	X	X	X	X
R/W	W	W	W	W	W	W	W	W
ADDR	FF0H							

WDTUNLK—Watchdog Timer Unlock
The software must write the correct unlocking sequence to this register before it is allowed to modify the contents of the Watchdog Timer reload registers.

Watchdog Timer Reload Upper, High and Low Byte Registers

The Watchdog Timer Reload Upper, High and Low Byte (WDTU, WDTH, WDTL) registers (Tables 59 through Table 61) form the 24-bit reload value that is loaded into the Watchdog Timer when a `WDT` instruction executes. The 24-bit reload value is {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`.

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. The features of 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
- BRG can be configured and used as a basic 16-bit timer
- Driver Enable 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.

REFSELL—Voltage Reference Level Select Low Bit; in conjunction with the High bit (REFSELH) in ADC Control/Status Register 1, this determines the level of the internal voltage reference; the following details the effects of {REFSELH, REFSELL};

Note: *This reference is independent of the Comparator reference.*

00= Internal Reference Disabled, reference comes from external pin.
01= Internal Reference set to 1.0 V
10= Internal Reference set to 2.0 V (default)

REFEXT—External Reference Select
0 = External reference buffer is disabled; V_{ref} pin is available for GPIO functions
1 = The internal ADC reference is buffered and connected to the V_{ref} pin

CONT
0 = Single-shot conversion. ADC data is output once at completion of the 5129 system clock cycles.
1 = Continuous conversion. ADC data updated every 256 system clock cycles.

ANAIN[3:0]—Analog Input Select
These bits select the analog input for conversion. Not all port pins in this list are available in all packages for Z8 Encore! XP® F0823 Series. For information on the port pins available with each package style, see Pin Description on page 7. Do not enable unavailable analog inputs. Usage of these bits changes depending on the buffer mode selected in ADC Control/Status Register 1.

For the reserved values, all input switches are disabled to avoid leakage or other undesirable operation. ADC samples taken with reserved bit settings are undefined.

Single-Ended:

0000 = ANA0
0001 = ANA1
0010 = ANA2
0011 = ANA3
0100 = ANA4
0101 = ANA5
0110 = ANA6
0111 = ANA7
1000 = Reserved
1001 = Reserved
1010 = Reserved
1011 = Reserved
1100 = Reserved
1101 = Reserved
1110 = Reserved
1111 = Reserved

The randomized lot identifier is a 32 byte binary value, stored in the flash information page (for more details, see Reading the Flash Information Page on page 143 and Randomized Lot Identifier on page 149) and is unaffected by mass erasure of the device's flash memory.

Reading the Flash Information Page

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

```
; get value at info address 60 (FE60h)
ldx FPS, #%80 ; enable access to flash info page
ld R0, #%FE
ld R1, #%60
ldc R2, @RR0 ; R2 now contains the calibration value
```

Flash Option Bit Control Register Definitions

Trim Bit Address Register

The Trim Bit Address (TRMADR) register contains the target address for an access to the trim option bits.

Table 85. Trim Bit Address Register (TRMADR)

BITS	7	6	5	4	3	2	1	0
FIELD	TRMADR - Trim Bit Address (00H to 1FH)							
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
ADDR	FF6H							

Table 106 lists additional symbols that are used throughout the Instruction Summary and Instruction Set Description sections.

Table 106. 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. For example,

dst ← dst + src

indicates the source data is added to the destination data and the result is stored in the destination location.

eZ8 CPU Instruction Classes

eZ8 CPU instructions are divided functionally into the following groups:

- Arithmetic
- Bit Manipulation
- Block Transfer
- CPU Control
- Load
- Logical
- Program Control

Table 110. CPU Control Instructions (Continued)

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

Table 111. 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 112. 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

Table 125. Analog-to-Digital Converter Electrical Characteristics and Timing

Symbol	Parameter	V_{DD} = 3.0 V to 3.6 V T_A = 0 °C to +70 °C (unless otherwise stated)			Units	Conditions
		Minimum	Typical	Maximum		
	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		Ω	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
	Continuous Conversion Time	–	256	–	System clock cycles	All measurements but temperature sensor
			512			Temperature sensor measurement
	Signal Input Bandwidth	–	10		kHz	As defined by -3 dB point
R_S	Analog Source Impedance[4]	–	–	10	kΩ	In unbuffered mode

UART Timing

Figure 32 and Table 130 provide timing information for UART pins for the case where CTS is used for flow control. The CTS to DE assertion delay (T1) assumes the transmit data register has been loaded with data prior to CTS assertion.

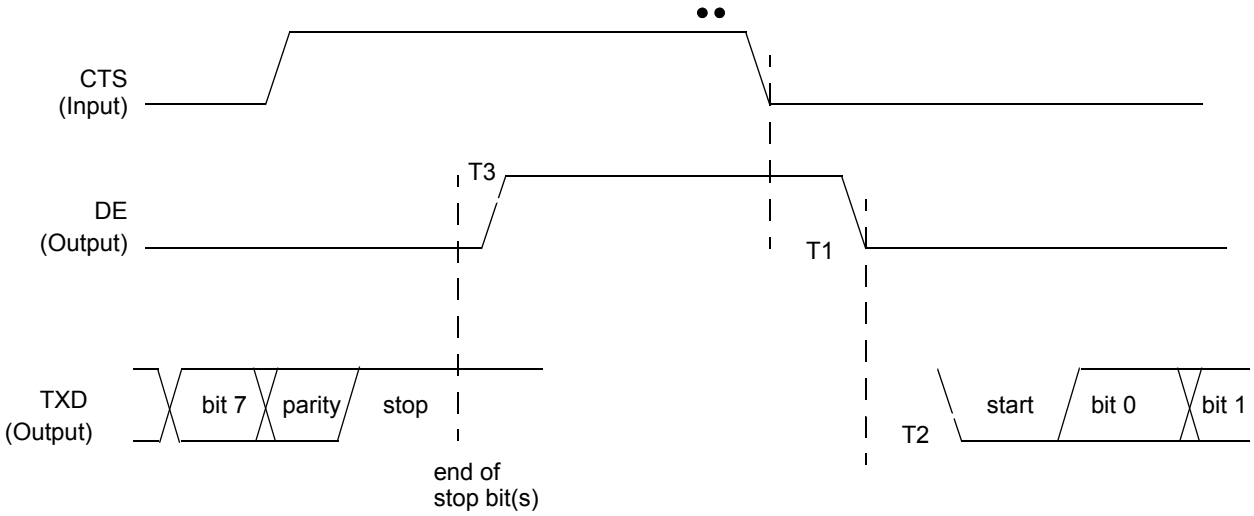
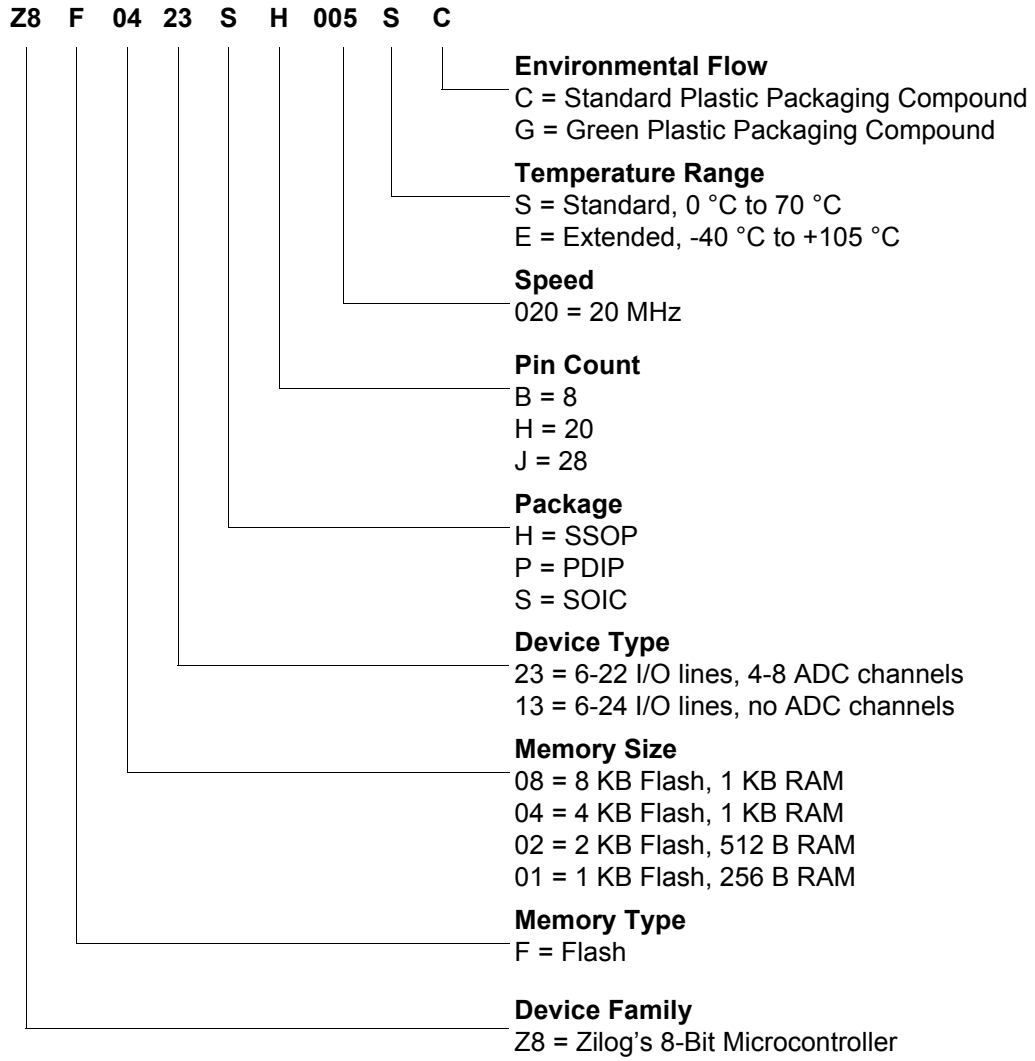


Figure 32. UART Timing With CTS

Table 130. UART Timing With CTS

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
UART			
T_1	CTS Fall to DE output delay	2 * XIN period	2 * XIN period + 1 bit time
T_2	DE assertion to TXD falling edge (start bit) delay	± 5	
T_3	End of Stop Bit(s) to DE deassertion delay	± 5	

Part Number Suffix Designations



D

E

F

X

Z