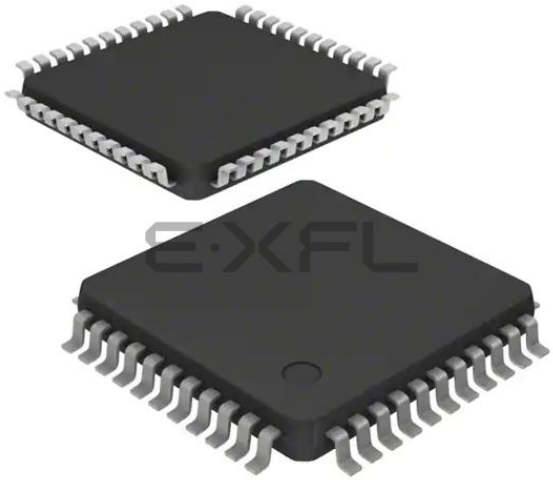

Welcome to E-XFL.COM

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	20MHz
Connectivity	I²C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	31
Program Memory Size	64KB (64K 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-LQFP
Supplier Device Package	44-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f6421an020sg

Table 5. Z8 Encore! XP F64xx Series Program Memory Maps (Continued)

Program Memory Address (Hex)	Function
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

Note: *See Table 23 on page 48 for a list of the interrupt vectors.

Data Memory

The Z8 Encore! XP F64xx Series does not use the eZ8 CPU's 64KB data memory address space.

Information Area

Table 6 describes the Z8 Encore! XP F64xx Series' 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 program memory and overlays the 512 bytes at addresses FE00H to FFFFH. When the Information Area access is enabled, execution of the LDC and LDCI instructions 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 program memory. Access to the Information Area is read-only.

Table 7. Z8 Encore! XP F64xx Series Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page
Timer 2 (continued)				
F14	Timer 2 PWM High Byte	T2PWMH	00	75
F15	Timer 2 PWM Low Byte	T2PWML	00	75
F16	Timer 2 Control 0	T2CTL0	00	76
F17	Timer 2 Control 1	T2CTL1	00	77
Timer 3 (Unavailable in the 44-Pin Package)				
F18	Timer 3 High Byte	T3H	00	72
F19	Timer 3 Low Byte	T3L	01	72
F1A	Timer 3 Reload High Byte	T3RH	FF	74
F1B	Timer 3 Reload Low Byte	T3RL	FF	74
F1C	Timer 3 PWM High Byte	T3PWMH	00	75
F1D	Timer 3 PWM Low Byte	T3PWML	00	75
F1E	Timer 3 Control 0	T3CTL0	00	76
F1F	Timer 3 Control 1	T3CTL1	00	77
20–3F	Reserved	—	XX	
UART 0				
F40	UART0 Transmit Data	U0TXD	XX	98
	UART0 Receive Data	U0RXD	XX	99
F41	UART0 Status 0	U0STAT0	0000011Xb	100
F42	UART0 Control 0	U0CTL0	00	102
F43	UART0 Control 1	U0CTL1	00	102
F44	UART0 Status 1	U0STAT1	00	100
F45	UART0 Address Compare Register	U0ADDR	00	105
F46	UART0 Baud Rate High Byte	U0BRH	FF	105
F47	UART0 Baud Rate Low Byte	U0BRL	FF	105
UART 1				
F48	UART1 Transmit Data	U1TXD	XX	98
	UART1 Receive Data	U1RXD	XX	99
F49	UART1 Status 0	U1STAT0	0000011Xb	100
F4A	UART1 Control 0	U1CTL0	00	102
F4B	UART1 Control 1	U1CTL1	00	102
F4C	UART1 Status 1	U1STAT1	00	100

Note: XX = Undefined.

enabled, the timer output pin changes state (from Low to High or from High to Low) at timer reload.

Observe the following procedure for configuring a timer for COUNTER Mode and initiating the count:

1. Write to the Timer Control 1 Register to:
   - Disable the timer.
   - Configure the timer for COUNTER Mode.
   - Select either the rising edge or falling edge of the timer input signal for the count. This also sets the initial logic level (High or Low) for the timer output alternate function. However, the timer output function does not have to be enabled.
2. Write to the Timer High and Low Byte registers to set the starting count value. This only affects the first pass in COUNTER Mode. After the first timer reload in COUNTER Mode, counting always begins at the reset value of 0001H. Generally, in COUNTER Mode the Timer High and Low Byte registers must be written with the value 0001H.
3. Write to the Timer Reload High and Low Byte registers to set the reload value.
4. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
5. Configure the associated GPIO port pin for the timer input alternate function.
6. If using the timer output function, configure the associated GPIO port pin for the timer output alternate function.
7. Write to the Timer Control 1 Register to enable the timer.

In COUNTER Mode, the number of timer input transitions since the timer start is calculated using the following equation:

COUNTER Mode Timer Input Transitions = Current Count Value – Start Value

PWM Mode

In PWM Mode, the timer outputs a Pulse-Width Modulator (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.

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

8. Execute an EI instruction to enable interrupts.

The UART is now configured for interrupt-driven data transmission. Because the UART Transmit Data Register is empty, an interrupt is generated immediately. When the UART transmit interrupt is detected, the associated interrupt service routine performs the following functions:

1. Write the UART Control 1 Register to select the outgoing address bit:
   - Set the MULTIPROCESSOR Bit Transmitter (MPBT) if sending an address byte; clear it if sending a data byte.
2. 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.
3. Clear the UART transmit interrupt bit in the applicable Interrupt Request Register.
4. Execute the IRET instruction to return from the interrupt service routine and wait for the Transmit Data Register to again become empty.

Receiving Data using the Polled Method

Observe the following procedure to configure the UART for polled data reception:

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. Write to the UART Control 1 Register to enable MULTIPROCESSOR Mode functions, if appropriate.
4. Write to the UART Control 0 Register to:
   - Set the receive enable bit (REN) to enable the UART for data reception
   - Enable parity, if appropriate and if MULTIPROCESSOR Mode is not enabled, and select either even or odd parity
5. Check the RDA bit in the UART Status 0 Register to determine if the Receive Data Register contains a valid data byte (indicated by a 1). If RDA is set to 1 to indicate available data, continue to Step 6. If the Receive Data Register is empty (indicated by a 0), continue to monitor the RDA bit awaiting reception of the valid data.

byte indicating an overrun error, the Receive Data Register must be read again to clear the error bits is the UART Status 0 Register. Updates to the Receive Data Register occur only when the next data word is received.

UART Data and Error Handling Procedure

Figure 18 displays the recommended procedure for use in UART receiver interrupt service routines.

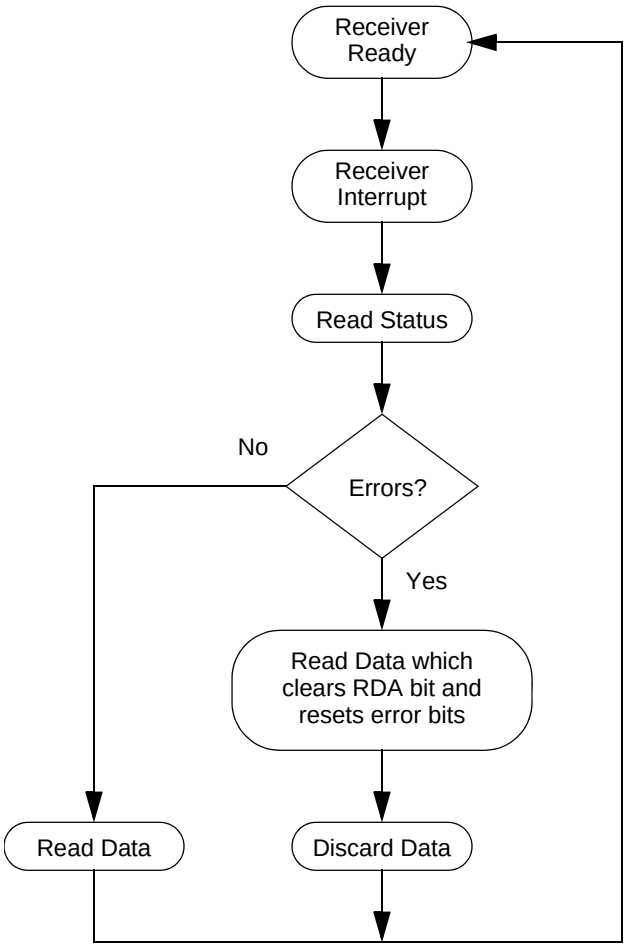

Figure 18. UART Receiver Interrupt Service Routine Flow

Baud Rate Generator Interrupts

If the Baud Rate Generator interrupt enable is set, the UART Receiver interrupt asserts when the UART Baud Rate Generator reloads. This action allows the Baud Rate Generator to function as an additional counter if the UART functionality is not employed.

Slave Operation

The SPI block is configured for SLAVE Mode operation by setting the SPIEN bit to 1 and the MMEN bit to 0 in the SPICTL Register and setting the SSIO bit to 0 in the SPIMODE Register. The IRQE, PHASE, CLKPOL, WOR bits in the SPICTL Register and the NUMBITS field in the SPIMODE Register must be set to be consistent with the other SPI devices. The STR bit in the SPICTL Register may be used if appropriate to force a *start-up* interrupt. The BIRQ bit in the SPICTL Register and the SSV bit in the SPIMODE Register are not used in SLAVE Mode. The SPI baud rate generator is not used in SLAVE Mode so the SPIBRH and SPIBRL registers need not be initialized.

If the slave has data to send to the master, the data must be written to the SPIDAT Register before the transaction starts (first edge of SCK when $\overline{SS}$ is asserted). If the SPIDAT Register is not written prior to the slave transaction, the MISO pin outputs whatever value is currently in the SPIDAT Register.

Due to the delay resulting from synchronization of the SPI input signals to the internal system clock, the maximum SPICLK baud rate that can be supported in SLAVE Mode is the system clock frequency (X_{IN}) divided by 8. This rate is controlled by the SPI master.

Error Detection

The SPI contains error detection logic to support SPI communication protocols and recognize when communication errors have occurred. The SPI Status Register indicates when a data transmission error has been detected.

Overrun (Write Collision)

An overrun error (write collision) indicates that a write to the SPI Data Register was attempted while a data transfer was in progress (in either MASTER or SLAVE modes). An overrun sets the OVR bit in the SPI Status Register to 1. Writing a 1 to OVR clears this error flag. The data register is not altered when a write occurs while data transfer is in progress.

Mode Fault (Multimaster Collision)

A mode fault indicates when more than one Master is trying to communicate at the same time (a multimaster collision). The mode fault is detected when the enabled Master's $\overline{SS}$ pin is asserted. A mode fault sets the COL bit in the SPI Status Register to 1. Writing a 1 to COL clears this error flag.

Slave Mode Abort

In the SLAVE Mode of operation, if the $\overline{SS}$ pin deasserts before all bits in a character have been transferred, the transaction is aborted. When this condition occurs, the ABT bit is set in the SPISTAT Register as well as the IRQ bit (indicating the transaction is complete).

Bit	Description (Continued)
[4] PHASE	**Phase Select** Sets the phase relationship of the data to the clock. For more information about operation of the PHASE bit, see the SPI Clock Phase and Polarity Control section on page 116.
[3] CLKPOL	**Clock Polarity** 0 = SCK idles Low (0). 1 = SCK idle High (1).
[2] WOR	**Wire-OR (OPEN-DRAIN) Mode Enabled** 0 = SPI signal pins not configured for open-drain. 1 = All four SPI signal pins (SCK, SS, MISO, MOSI) configured for open-drain function. This setting is typically used for multimaster and/or multislave configurations.
[1] MMEN	**SPI Master Mode Enable** 0 = SPI configured in SLAVE Mode. 1 = SPI configured in MASTER Mode.
[0] SPIEN	**SPI Enable** 0 = SPI disabled. 1 = SPI enabled.

SPI Status Register

The SPI Status Register, shown in Table 66, indicates the current state of the SPI. All bits revert to their reset state if the SPIEN bit in the SPICTL Register = 0.

Table 66. SPI Status Register (SPISTAT)

Bit	7	6	5	4	3	2	1	0
Field	IRQ	OVR	COL	ABT	Reserved		TXST	SLAS
RESET	0							1
R/W	R/W*				R			
Address	F62H							

Note: R/W* = Read access. Write a 1 to clear the bit to 0.

Bit	Description
[7] 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.
[6] OVR	**Overrun** 0 = An overrun error has not occurred. 1 = An overrun error has been detected.

Operation

This section describes the operational aspects of the ADC's power-down and conversion features.

Automatic Power-Down

If the ADC is idle (i.e., no conversions are in progress) for 160 consecutive system clock cycles, portions of the ADC are automatically powered down. From this powered-down state, the ADC requires 40 system clock cycles to power up. The ADC powers up when a conversion is requested using the ADC Control Register.

Single-Shot Conversion

When configured for single-shot conversion, the ADC performs a single analog-to-digital conversion on the selected analog input channel. After completion of the conversion, the ADC shuts down. Observe the following procedure for setting up the ADC and initiating a single-shot conversion:

1. Enable the appropriate analog inputs by configuring the general-purpose I/O pins for alternate function. This configuration disables the digital input and output drivers.
2. Write to the ADC Control Register to configure the ADC and begin the conversion. The bit fields in the ADC Control Register can be written simultaneously:
   - Write to the ANAIN[3:0] field to select one of the 12 analog input sources
   - Clear CONT to 0 to select a single-shot conversion
   - Write to the $\overline{\text{VREF}}$ bit to enable or disable the internal voltage reference generator
   - Set CEN to 1 to start the conversion
3. CEN remains 1 while the conversion is in progress. A single-shot conversion requires 5129 system clock cycles to complete. If a single-shot conversion is requested from an ADC powered-down state, the ADC uses 40 additional clock cycles to power up before beginning the 5129 cycle conversion.
4. When the conversion is complete, the ADC control logic performs the following operations:
   - 10-bit data result written to {ADCD_H[7:0], ADCD_L[7:6]}
   - CEN resets to 0 to indicate the conversion is complete
   - An interrupt request is sent to the Interrupt Controller
5. If the ADC remains idle for 160 consecutive system clock cycles, it is automatically powered down.

4. Write the second unlock command 8CH to the Flash Control Register.
5. Rewrite the page written in Step 2 to the Page Select Register.
6. Write Flash memory using LDC or LDCI instructions to program the Flash.
7. Repeat Step 6 to program additional memory locations on the same page.
8. Write 00H to the Flash Control Register to lock the Flash Controller.

Page Erase

Flash memory can be erased one page (512 bytes) at a time. Page-erasing Flash memory sets all bytes in a page to the value FFH. The Page Select Register identifies the page to be erased. While the Flash Controller executes the Page Erase operation, the eZ8 CPU idles; however, the system clock and on-chip peripherals continue to operate. The eZ8 CPU resumes operation after the Page Erase operation completes. Interrupts that occur when the Page Erase operation is in progress are serviced after the Page Erase operation is complete. When the Page Erase operation is complete, the Flash Controller returns to its locked state. Only pages located in unprotected sectors can be erased.

Observe the following procedure to perform a Page Erase operation:

1. Write 00H to the Flash Control Register to reset the Flash Controller.
2. Write the page to be erased to the Page Select Register.
3. Write the first unlock command 73H to the Flash Control Register.
4. Write the second unlock command 8CH to the Flash Control Register.
5. Rewrite the page written in Step 2 to the Page Select Register.
6. Write the Page Erase command 95H to the Flash Control Register.

Mass Erase

The Flash memory cannot be mass-erased by user code.

Flash Controller Bypass

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

Flash Controller Bypass is recommended for gang programming applications and large volume customers who do not require in-circuit programming of Flash memory.

```
DBG ← 00H
DBG OCDREV[15:8] (Major revision number)
DBG OCDREV[7:0] (Minor revision number)
```

Read OCD Status Register (02H). The Read OCD Status Register command reads the OCDSTAT Register.

```
DBG ← 02H
DBG OCDSTAT[7:0]
```

Write OCD Control Register (04H). The Write OCD Control Register command writes the data that follows to the OCDCTL Register. When the Read Protect option bit is enabled, the DBGMODE bit (OCDCTL[7]) can only be set to 1, it cannot be cleared to 0 and the only method of putting the device back into normal operating mode is to reset the device.

```
DBG ← 04H
DBG ← OCDCTL[7:0]
```

Read OCD Control Register (05H). The Read OCD Control Register command reads the value of the OCDCTL Register.

```
DBG ← 05H
DBG OCDCTL[7:0]
```

Write Program Counter (06H). The Write Program Counter command writes the data that follows to the eZ8 CPU's Program Counter (PC). If the device is not in DEBUG Mode or if the Read Protect option bit is enabled, the Program Counter (PC) values are discarded.

```
DBG ← 06H
DBG ← ProgramCounter[15:8]
DBG ← ProgramCounter[7:0]
```

Read Program Counter (07H). The Read Program Counter command reads the value in the eZ8 CPU's Program Counter (PC). If the device is not in DEBUG Mode or if the Read Protect option bit is enabled, this command returns FFFFH.

```
DBG ← 07H
DBG ProgramCounter[15:8]
DBG ProgramCounter[7:0]
```

Write Register (08H). The Write Register command writes data to the Register File. Data can be written 1-256 bytes at a time (256 bytes can be written by setting size to zero). If the device is not in DEBUG Mode, the address and data values are discarded. If the Read Protect option bit is enabled, then only writes to the Flash Control Registers are allowed and all other register write data values are discarded.

```
DBG ← 08H
DBG ← {4'h0,Register Address[11:8]}
DBG ← Register Address[7:0]
```

On-Chip Oscillator

The products in the Z8 Encore! XP F64xx Series feature an on-chip oscillator for use with external crystals with frequencies from 32kHz to 20MHz. In addition, the oscillator can support external RC networks with oscillation frequencies up to 4MHz or ceramic resonators with oscillation frequencies up to 20MHz. This oscillator generates the primary system clock for the internal eZ8 CPU and the majority of the on-chip peripherals. Alternatively, the X_{IN} input pin can also accept a CMOS-level clock input signal (32kHz–20MHz). If an external clock generator is used, the X_{OUT} pin must be left unconnected.

When configured for use with crystal oscillators or external clock drivers, the frequency of the signal on the X_{IN} input pin determines the frequency of the system clock (that is, no internal clock divider). In RC operation, the system clock is driven by a clock divider (divide by 2) to ensure 50% duty cycle.

Operating Modes

The Z8 Encore! XP F64xx Series products support four different oscillator modes:

- On-chip oscillator configured for use with external RC networks (<4MHz)
- Minimum power for use with very low frequency crystals (32kHz to 1.0MHz)
- Medium power for use with medium frequency crystals or ceramic resonators (0.5MHz to 10.0MHz)
- Maximum power for use with high frequency crystals or ceramic resonators (8.0MHz to 20.0MHz)

The oscillator mode is selected through user-programmable option bits. For more information, see the Option Bits chapter on page 180.

Crystal Oscillator Operation

Figure 40 displays a recommended configuration for connection with an external fundamental-mode, parallel-resonant crystal operating at 20MHz. Recommended 20MHz crystal specifications are provided in Table 105. Resistor R1 is optional and limits total power dissipation by the crystal. The printed circuit board layout must add no more than 4 pF of stray capacitance to either the X_{IN} or X_{OUT} pins. If oscillation does not occur, reduce the values of capacitors C1 and C2 to decrease loading.

DC Characteristics

Table 107 lists the DC characteristics of the Z8 Encore! XP F64xx Series products. All voltages are referenced to V_{SS}, the primary system ground.

Table 107. DC Characteristics

		T_A = –40°C to 125°C				
Symbol	**Parameter**	**Minimum**	**Typical**	**Maximum**	**Units**	**Conditions**
V_{DD}	Supply Voltage	3.0	–	3.6	V	
V_{IL1}	Low Level Input Voltage	–0.3	–	0.3*V_{DD}	V	For all input pins except $\overline{\text{RESET}}$, DBG, X_{IN}
V_{IL2}	Low Level Input Voltage	–0.3	–	0.2*V_{DD}	V	For $\overline{\text{RESET}}$, DBG, and X_{IN}.
V_{IH1}	High Level Input Voltage	0.7*V_{DD}	–	5.5	V	Port A, C, D, E, F, and G pins.
V_{IH2}	High Level Input Voltage	0.7*V_{DD}	–	V_{DD}+0.3	V	Port B and H pins.
V_{IH3}	High Level Input Voltage	0.8*V_{DD}	–	V_{DD}+0.3	V	$\overline{\text{RESET}}$, DBG, and X_{IN} pins
V_{OL1}	Low Level Output Voltage Standard Drive	–	–	0.4	V	I_{OL} = 2 mA; V_{DD} = 3.0V High Output Drive disabled.
V_{OH1}	High Level Output Voltage Standard Drive	2.4	–	–	V	I_{OH} = –2 mA; V_{DD} = 3.0V High Output Drive disabled.
V_{OL2}	Low Level Output Voltage High Drive	–	–	0.6	V	I_{OL} = 20 mA; V_{DD} = 3.3V High Output Drive enabled T_A = –40°C to +70°C
V_{OH2}	High Level Output Voltage High Drive	2.4	–	–	V	I_{OH} = –20 mA; V_{DD} = 3.3V High Output Drive enabled; T_A = –40°C to +70°C
V_{OL3}	Low Level Output Voltage High Drive	–	–	0.6	V	I_{OL} = 15 mA; V_{DD} = 3.3V High Output Drive enabled; T_A = +70°C to +105°C

Notes:
1. This condition excludes all pins that have on-chip pull-ups, when driven Low.
2. These values are provided for design guidance only and are not tested in production.

Table 111 list the Flash memory electrical characteristics and timing.

Table 111. Flash Memory Electrical Characteristics and Timing

	V_{DD} = 3.0–3.6V T_A = –40°C to 125°C				
Parameter	**Minimum**	**Typical**	**Maximum**	**Units**	**Notes**
Flash Byte Read Time	50	–	–	ns	
Flash Byte Program Time	20	–	40	µs	
Flash Page Erase Time	10	–	–	ms	
Flash Mass Erase Time	200	–	–	ms	
Writes to Single Address Before Next Erase	–	–	2		
Flash Row Program Time	–	–	8	ms	Cumulative program time for single row cannot exceed limit before next erase. This parameter is only an issue when bypassing the Flash Controller.
Data Retention	100	–	–	years	25°C
Endurance, –40°C to 105°C	10,000	–	–	cycles	Program/erase cycles
Endurance, 106°C to 125°C	1,000	–	–	cycles	Program/erase cycles

Table 112 lists the Watchdog Timer electrical characteristics and timing.

Table 112. Watchdog Timer Electrical Characteristics and Timing

		V_{DD} = 3.0–3.6V T_A = –40°C to 125°C				
Symbol	**Parameter**	**Minimum**	**Typical**	**Maximum**	**Units**	**Conditions**
F_{WDT}	WDT Oscillator Frequency	5	10	20	kHz	
I_{WDT}	WDT Oscillator Current including internal RC Oscillator	–	<1	5	µA	

Figure 57 and Table 122 provide timing information for UART pins for the case where the Clear To Send input signal ($\overline{CTS}$) is not 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 $\overline{DE}$. $\overline{DE}$ asserts after the UART Transmit Data Register has been written. $\overline{DE}$ remains asserted for multiple characters as long as the Transmit Data Register is written with the next character before the current character has completed.

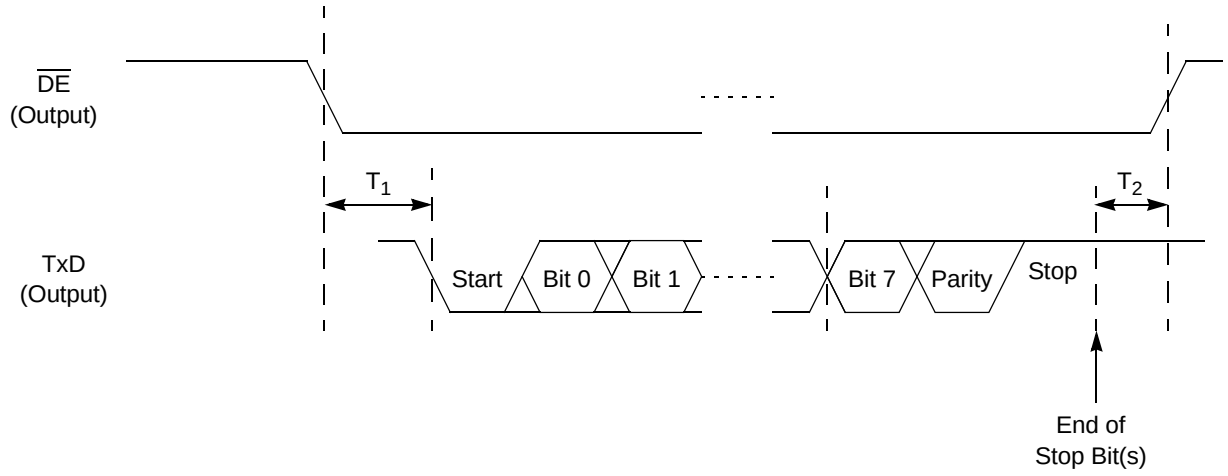

Figure 57. UART Timing without $\overline{CTS}$

Table 122. UART Timing without $\overline{CTS}$

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
T_1	$\overline{DE}$ Assertion to TxD Falling Edge (Start) Delay	1 bit period	1 bit period + 1 * X_{IN} period
T_2	End of stop bit(s) to $\overline{DE}$ Deassertion Delay	1 * X_{IN} period	2 * X_{IN} period

Hex Address: F19

Table 163. Timer 0–3 Low Byte Register (TxL)

Bit	7	6	5	4	3	2	1	0
Field	TL							
RESET	0							1
R/W	R/W							
Address	F01H, F09H, F11H, F19H							

Hex Address: F1A

Table 164. Timer 0–3 Reload High Byte Register (TxRH)

Bit	7	6	5	4	3	2	1	0
Field	TRH							
RESET	1							
R/W	R/W							
Address	F02H, F0AH, F12H, F1AH							

Hex Address: F1B

Table 165. Timer 0–3 Reload Low Byte Register (TxRL)

Bit	7	6	5	4	3	2	1	0
Field	TRL							
RESET	1							
R/W	R/W							
Address	F03H, F0BH, F13H, F1BH							

Hex Address: F1C

Table 166. Timer 0–3 PWM High Byte Register (TxPWMH)

Bit	7	6	5	4	3	2	1	0
Field	PWMH							
RESET	0							
R/W	R/W							
Address	F04H, F0CH, F14H, F1CH							

Hex Address: F43

Table 174. UART Control 1 Register (U*x*CTL1)

Bit	7	6	5	4	3	2	1	0
Field	MPMD[1]	MPEN	MPMD[0]	MPBT	DEPOL	BRGCTL	RDAIRQ	IREN
RESET	0							
R/W	R/W							
Address	F43H and F4BH							

Hex Address: F44

Table 175. UART Status 1 Register (U*x*STAT1)

Bit	7	6	5	4	3	2	1	0
Field	Reserved						NEWFRM	MPRX
RESET	0							
R/W	R				R/W		R	
Address	F44H and F4CH							

Hex Address: F45

Table 176. UART Address Compare Register (U*x*ADDR)

Bit	7	6	5	4	3	2	1	0
Field	COMP_ADDR							
RESET	0							
R/W	R/W							
Address	F45H and F4DH							

Hex Address: F46

Table 177. UART Baud Rate High Byte Register (U*x*BRH)

Bit	7	6	5	4	3	2	1	0
Field	BRH							
RESET	1							
R/W	R/W							
Address	F46H and F4EH							

General-Purpose Input/Output (GPIO)

For more information about these GPIO Control registers, see the GPIO Control Register Definitions section on page 39.

Hex Address: FD0

Table 229. Port A–H GPIO Address Registers (P*x*ADDR)

Bit	7	6	5	4	3	2	1	0
Field	PADDR[7:0]							
RESET	00H							
R/W	R/W							
Address	FD0H, FD4H, FD8H, FDCH, FE0H, FE4H, FE8H, FECH							

Hex Address: FD1

Table 230. Port A–H Control Registers (P*x*CTL)

Bit	7	6	5	4	3	2	1	0
Field	PCTL							
RESET	00H							
R/W	R/W							
Address	FD1H, FD5H, FD9H, FDDH, FE1H, FE5H, FE9H, FEDH							

Hex Address: FD2

Table 231. Port A–H Input Data Registers (PxIN)

Bit	7	6	5	4	3	2	1	0
Field	PIN7	PIN6	PIN5	PIN4	PIN3	PIN2	PIN1	PIN0
RESET	X							
R/W	R							
Address	FD2H, FD6H, FDAH, FDEH, FE2H, FE6H, FEAH, FEEH							

Hex Address: FD3

Table 232. Port A–H Output Data Register (P*x*OUT)

Bit	7	6	5	4	3	2	1	0
Field	POUT7	POUT6	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0
RESET	0							
R/W	R/W							
Address	FD3H, FD7H, FDBH, FDFH, FE3H, FE7H, FEBH, FEFH							

Hex Address: FD4

Table 233. Port A–H GPIO Address Registers (P*x*ADDR)

Bit	7	6	5	4	3	2	1	0
Field	PADDR[7:0]							
RESET	00H							
R/W	R/W							
Address	FD0H, FD4H, FD8H, FDCH, FE0H, FE4H, FE8H, FECH							

Hex Address: FD5

Table 234. Port A–H Control Registers (P*x*CTL)

Bit	7	6	5	4	3	2	1	0
Field	PCTL							
RESET	00H							
R/W	R/W							
Address	FD1H, FD5H, FD9H, FDDH, FE1H, FE5H, FE9H, FEDH							

Hex Address: FD6

Table 235. Port A–H Input Data Registers (PxIN)

Bit	7	6	5	4	3	2	1	0
Field	PIN7	PIN6	PIN5	PIN4	PIN3	PIN2	PIN1	PIN0
RESET	X							
R/W	R							
Address	FD2H, FD6H, FDAH, FDEH, FE2H, FE6H, FEAH, FEEH							

D

E

F