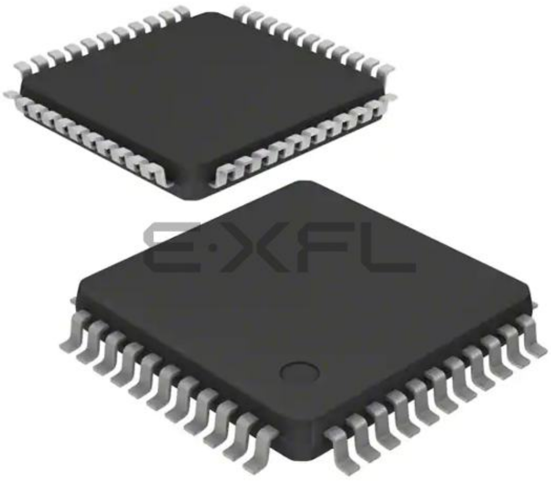
Zilog - Z8F4821AN020EC Datasheet



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	-40°C ~ 105°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/z8f4821an020ec

Warning: DO NOT USE IN LIFE SUPPORT

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer

©2007 by Zilog, Inc. All rights reserved. Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZILOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZILOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering.

Z8, Z8 Encore!, Z8 Encore! XP, Z8 Encore! MC, Crimzon, eZ80, and ZNEO are trademarks or registered trademarks of Zilog, Inc. All other product or service names are the property of their respective owners.

ANAB ACCREDITED

BSI™

ISO 9001:2000
FS 507510

Zilog products are designed and manufactured under an ISO registered 9001:2000 Quality Management System. For more details, please visit www.zilog.com/quality.

Master Interrupt Enable 69
Interrupt Vectors and Priority 70
Interrupt Assertion 70
Software Interrupt Assertion 70
Interrupt Control Register Definitions 71
Interrupt Request 0 Register 71
Interrupt Request 1 Register 72
Interrupt Request 2 Register 73
IRQ0 Enable High and Low Bit Registers 74
IRQ1 Enable High and Low Bit Registers 75
IRQ2 Enable High and Low Bit Registers 76
Interrupt Edge Select Register 78
Interrupt Port Select Register 78
Interrupt Control Register 79

Timers 81

Overview 81
Architecture 81
Operation 82
Timer Operating Modes 82
Reading the Timer Count Values 90
Timer Output Signal Operation 90
Timer Control Register Definitions 90
Timer 0-3 High and Low Byte Registers 90
Timer Reload High and Low Byte Registers 91
Timer 0-3 PWM High and Low Byte Registers 92
Timer 0-3 Control 0 Registers 93
Timer 0-3 Control 1 Registers 94

Watchdog Timer 97

Overview 97
Operation 97
Watchdog Timer Refresh 98
Watchdog Timer Time-Out Response 98
Watchdog Timer Reload Unlock Sequence 99
Watchdog Timer Control Register Definitions 100
Watchdog Timer Control Register 100
Watchdog Timer Reload Upper, High and Low Byte Registers 101

Manual Objectives

This Product Specification provides detailed operating information for the Flash devices within Zilog's Z8 Encore! XP® 64K Series Flash Microcontrollers Microcontroller (MCU) products. Within this document, the Z8F642x, Z8F482x, Z8F322x, Z8F242x, and Z8F162x devices are referred to collectively as the Z8 Encore! XP® 64K Series Flash Microcontrollers unless specifically stated otherwise.

About This Manual

Zilog® recommends that you read and understand everything in this manual before setting up and using the product. However, we recognize that there are different styles of learning. Therefore, we have designed this Product Specification to be used either as a *how to* procedural manual or a reference guide to important data.

Intended Audience

This document is written for Zilog customers who are experienced at working with microcontrollers, integrated circuits, or printed circuit assemblies.

Manual Conventions

The following assumptions and conventions are adopted to provide clarity and ease of use:

Courier Typeface

Commands, code lines and fragments, bits, equations, hexadecimal addresses, and various executable items are distinguished from general text by the use of the `Courier` typeface. Where the use of the font is not indicated, as in the Index, the name of the entity is presented in upper case.

- Example: FLAGS[1] is `smrf`.

Hexadecimal Values

Hexadecimal values are designated by uppercase *H* suffix and appear in the `Courier` typeface.

- Example: R1 is set to `F8H.`

Brackets

The square brackets, [ ], indicate a register or bus.

- Example: For the register R1[7:0], R1 is an 8-bit register, R1[7] is the most significant bit, and R1[0] is the least significant bit.

Signal Descriptions

Table 3 describes the Z8 Encore! XP signals. To determine the signals available for the specific package styles, see Pin Configurations on page 8.

Table 3. Signal Descriptions

Signal Mnemonic	I/O	Description
General-Purpose I/O Ports A-H		
PA[7:0]	I/O	Port A[7:0]. These pins are used for general-purpose I/O and support 5 V-tolerant inputs.
PB[7:0]	I/O	Port B[7:0]. These pins are used for general-purpose I/O.
PC[7:0]	I/O	Port C[7:0]. These pins are used for general-purpose I/O. These pins are used for general-purpose I/O and support 5 V-tolerant inputs
PD[7:0]	I/O	Port D[7:0]. These pins are used for general-purpose I/O. These pins are used for general-purpose I/O and support 5 V-tolerant inputs
PE[7:0]	I/O	Port E[7:0]. These pins are used for general-purpose I/O. These pins are used for general-purpose I/O and support 5 V-tolerant inputs.
PF[7:0]	I/O	Port F[7:0]. These pins are used for general-purpose I/O. These pins are used for general-purpose I/O and support 5 V-tolerant inputs.
PG[7:0]	I/O	Port G[7:0]. These pins are used for general-purpose I/O. These pins are used for general-purpose I/O and support 5 V-tolerant inputs.
PH[3:0]	I/O	Port H[3:0]. These pins are used for general-purpose I/O.
I^{2}C Controller		
SCL	O	Serial Clock. This is the output clock for the I^{2}C. This pin is multiplexed with a general-purpose I/O pin. When the general-purpose I/O pin is configured for alternate function to enable the SCL function, this pin is open-drain.
SDA	I/O	Serial Data. This open-drain pin transfers data between the I^{2}C and a slave. This pin is multiplexed with a general-purpose I/O pin. When the general-purpose I/O pin is configured for alternate function to enable the SDA function, this pin is open-drain.
SPI Controller		
$\overline{\text{SS}}$	I/O	Slave Select. This signal can be an output or an input. If the Z8 Encore! XP 64K Series Flash Microcontrollers is the SPI master, this pin may be configured as the Slave Select output. If the Z8 Encore! XP 64K Series Flash Microcontrollers is the SPI slave, this pin is the input slave select. It is multiplexed with a general-purpose I/O pin.

Port F Control
PFCTL (FE5H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

Port F Control[7:0]
Provides Access to Port Sub-Registers

Port F Input Data
PFIN (FE6H - Read Only)

D7	D6	D5	D4	D3	D2	D1	D0

Port F Input Data [7:0]

Port F Output Data
PFOUT (FE7H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

Port F Output Data [7:0]

Port G Address
PGADDR (FE8H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

Port G Address[7:0]
Selects Port Sub-Registers:
00H = No function
01H = Data direction
02H = Alternate function
03H = Output control (open-drain)
04H = High drive enable
05H = Stop Mode Recovery enable
06H-FFH = No function

Port G Control
PGCTL (FE9H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

Port G Control[7:0]
Provides Access to Port Sub-Registers

Port G Input Data
PGIN (FEAH - Read Only)

D7	D6	D5	D4	D3	D2	D1	D0

Port G Input Data [7:0]

Port G Output Data
PGOUT (FEBH - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

Port G Output Data [7:0]

Port H Address
PHADDR (FECH - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

Port H Address[7:0]
Selects Port Sub-Registers:
00H = No function
01H = Data direction
02H = Alternate function
03H = Output control (open-drain)
04H = High drive enable
05H = Stop Mode Recovery enable
06H-FFH = No function

Port H Control
PHCTL (FEDH - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

Port H Control [3:0]
Provides Access to Port Sub-Registers

Reserved

Port H Input Data
PHIN (FEEH - Read Only)

D7	D6	D5	D4	D3	D2	D1	D0

Port H Input Data [3:0]

Reserved

Port H Output Data
PHOUT (FEFH - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

Port H Output Data [3:0]

Reserved

On-Chip Debugger Initiated Reset

A Power-On Reset can be initiated using the On-Chip Debugger by setting the RST bit in the OCD Control register. The On-Chip Debugger block is not reset but the rest of the chip goes through a normal system reset. The RST bit automatically clears during the system reset. Following the system reset the POR bit in the WDT Control register is set.

Stop Mode Recovery

STOP mode is entered by the eZ8 executing a STOP instruction. For detailed STOP mode information, see Low-Power Modes on page 47. During Stop Mode Recovery, the devices are held in reset for 66 cycles of the Watchdog Timer oscillator followed by 16 cycles of the system clock. Stop Mode Recovery only affects the contents of the Watchdog Timer Control register. Stop Mode Recovery does not affect any other values in the Register File, including the Stack Pointer, Register Pointer, Flags, peripheral control registers, and general-purpose RAM.

The eZ8™ CPU fetches the Reset vector at Program Memory addresses 0002H and 0003H and loads that value into the Program Counter. Program execution begins at the Reset vector address. Following Stop Mode Recovery, the STOP bit in the Watchdog Timer Control Register is set to 1. Table 10 lists the Stop Mode Recovery sources and resulting actions.

Table 10. 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

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 the 64K Series devices are configured to respond to interrupts, the eZ8 CPU services the Watchdog Timer interrupt request following the normal Stop Mode Recovery sequence.

Interrupt Controller

Overview

The interrupt controller on the 64K Series products prioritizes the interrupt requests from the on-chip peripherals and the GPIO port pins. The features of the interrupt controller include the following:

- 24 unique interrupt vectors:
  - 12 GPIO port pin interrupt sources
  - 12 on-chip peripheral interrupt sources
- Flexible GPIO interrupts
  - Eight selectable rising and falling edge GPIO interrupts
  - Four dual-edge interrupts
- Three levels of individually programmable interrupt priority
- Watchdog Timer can be configured to generate an interrupt

Interrupt requests (IRQs) allow peripheral devices to suspend CPU operation in an orderly manner and force the CPU to start an interrupt service routine (ISR). Usually this interrupt service routine is involved with the exchange of data, status information, or control information between the CPU and the interrupting peripheral. When the service routine is completed, the CPU returns to the operation from which it was interrupted.

The eZ8 CPU supports both vectored and polled interrupt handling. For polled interrupts, the interrupt control has no effect on operation. For more information on interrupt servicing by the eZ8 CPU, refer to *eZ8™ CPU Core User Manual (UM0128)* available for download at www.zilog.com.

Interrupt Vector Listing

Table 23 lists all of the interrupts available in order of priority. The interrupt vector is stored with the most-significant byte (MSB) at the even Program Memory address and the least-significant byte (LSB) at the following odd Program Memory address.

The timer continues counting up to the 16-bit Reload value stored in the Timer Reload High and Low Byte registers. Upon reaching the Reload value, the timer generates an interrupt and continues counting.

Follow the steps below for configuring a timer for CAPTURE mode and initiating the count:

1. Write to the Timer Control 1 register to:
   - Disable the timer
   - Configure the timer for CAPTURE mode.
   - Set the prescale value.
   - Set the Capture edge (rising or falling) for the Timer Input.
2. Write to the Timer High and Low Byte registers to set the starting count value (typically `0001H`).
3. Write to the Timer Reload High and Low Byte registers to set the Reload value.
4. Clear the Timer PWM High and Low Byte registers to `0000H`. This allows the software to determine if interrupts were generated by either a capture event or a reload. If the PWM High and Low Byte registers still contain `0000H` after the interrupt, then the interrupt was generated by a Reload.
5. If desired, 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 Input alternate function.
7. Write to the Timer Control 1 register to enable the timer and initiate counting.

In CAPTURE mode, the elapsed time from timer start to Capture event can be calculated using the following equation:

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

COMPARE Mode

In COMPARE mode, the timer counts up to the 16-bit maximum Compare value stored in the Timer Reload High and Low Byte registers. The timer input is the system clock. Upon reaching the Compare value, the timer generates an interrupt and counting continues (the timer value is not reset to `0001H`). Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state (from Low to High or from High to Low) upon Compare.

If the Timer reaches `FFFFH`, the timer rolls over to `0000H` and continue counting.

Timer 0-3 Control 1 Registers

The Timer 0-3 Control 1 (TxCTL1) registers enable/disable the timers, set the prescaler value, and determine the timer operating mode.

Table 46. Timer 0-3 Control 1 Register (TxCTL1)

BITS	7	6	5	4	3	2	1	0
FIELD	TEN	TPOL	PRES			TMODE		
RESET	0							
R/W	R/W							
ADDR	F07H, F0FH, F17H, F1FH							

TEN—Timer Enable
0 = Timer is disabled.
1 = Timer enabled to count.

TPOL—Timer Input/Output Polarity
Operation of this bit is a function of the current operating mode of the timer.

ONE-SHOT mode

When the timer is disabled, the Timer Output signal is set to the value of this bit. When the timer is enabled, the Timer Output signal is complemented upon timer Reload.

CONTINUOUS mode

When the timer is disabled, the Timer Output signal is set to the value of this bit. When the timer is enabled, the Timer Output signal is complemented upon timer Reload.

COUNTER mode

When the timer is disabled, the Timer Output signal is set to the value of this bit. When the timer is enabled, the Timer Output signal is complemented upon timer Reload.

0 = Count occurs on the rising edge of the Timer Input signal.
1 = Count occurs on the falling edge of the Timer Input signal.

PWM mode

0 = Timer Output is forced Low (0) when the timer is disabled. When enabled, the Timer Output is forced High (1) upon PWM count match and forced Low (0) upon Reload.

5. Write the Watchdog Timer Reload Low Byte register (WDTL).

All steps of the Watchdog Timer Reload Unlock sequence must be written in the order just listed. There must be no other register writes between each of these operations. If a register write occurs, the lock state machine resets and no further writes can occur, unless the sequence is restarted. The value in the Watchdog Timer Reload registers is loaded into the counter when the Watchdog Timer is first enabled and every time a WDT instruction is executed.

Watchdog Timer Control Register Definitions

Watchdog Timer Control Register

The Watchdog Timer Control (WDTCTL) register (Table 48) is a Read-Only register that indicates the source of the most recent Reset event, indicates a Stop Mode Recovery event, and indicates a Watchdog Timer time-out. Reading this register resets the upper four bits to 0.

Writing the 55H, AAH unlock sequence to the Watchdog Timer Control (WDTCTL) register address unlocks the three 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.

Table 48. Watchdog Timer Control Register (WDTCTL)

BITS	7	6	5	4	3	2	1	0
FIELD	POR	STOP	WDT	EXT	Reserved			SM
RESET	See descriptions below			0				
R/W	R							
ADDR	FF0H							

Reset or Stop Mode Recovery Event	POR	STOP	WDT	EXT
Power-On Reset	1	0	0	0
Reset using $\overline{\text{RESET}}$ pin assertion	0	0	0	1
Reset using Watchdog Timer time-out	0	0	1	0
Reset using the On-Chip Debugger (OCDCTL[1] set to 1)	1	0	0	0
Reset from STOP Mode using DBG Pin driven Low	1	0	0	0
Stop Mode Recovery using GPIO pin transition	0	1	0	0
Stop Mode Recovery using Watchdog Timer time-out	0	1	1	0

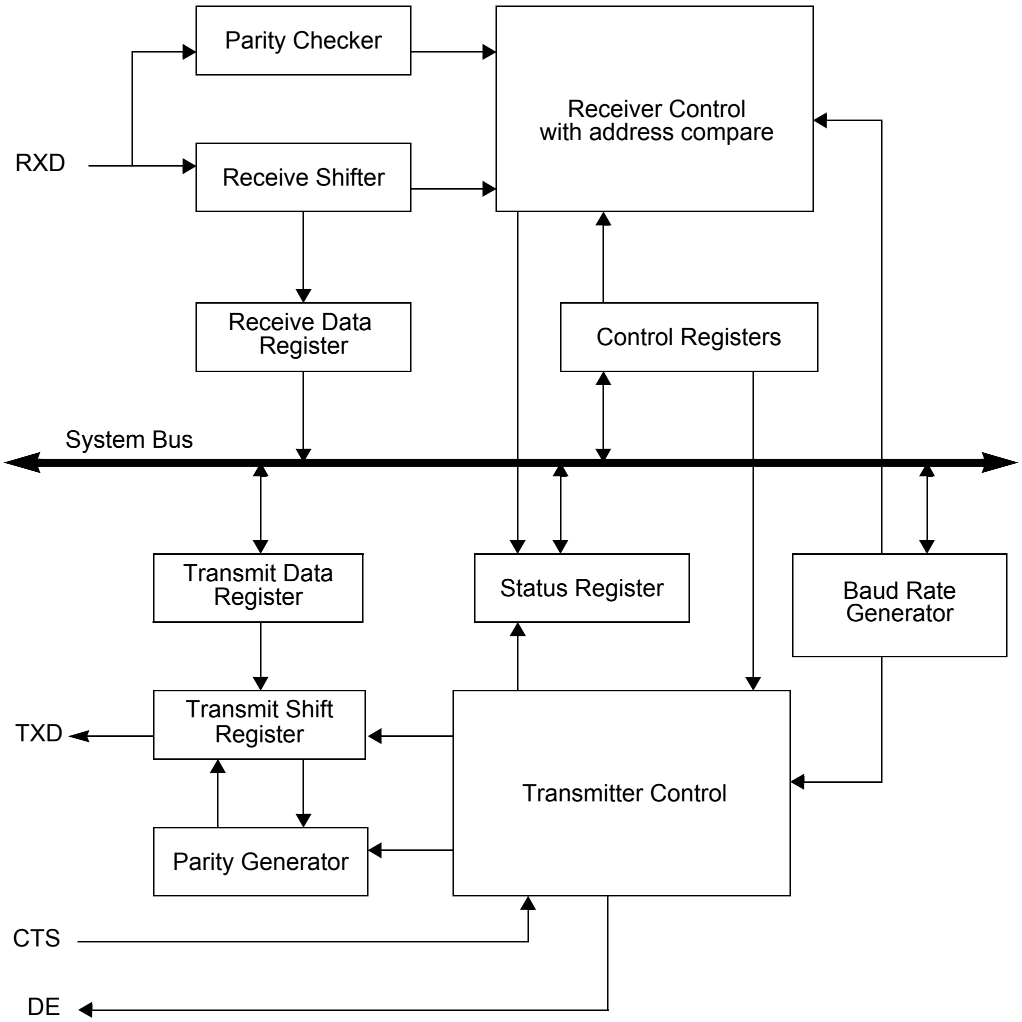


Figure 13. UART Block Diagram

Operation

Data Format

The UART always transmits and receives data in an 8-bit data format, least-significant bit first. An even or odd parity bit can be optionally added to the data stream. Each character begins with an active Low Start bit and ends with either 1 or 2 active High Stop bits. Figure 14 and Figure 15 on page 105 displays the asynchronous data format employed by the UART without parity and with parity, respectively.

Table 61. UART Baud Rates (Continued)

9.60	23	9.73	1.32	9.60	12	9.60	0.00
4.80	47	4.76	-0.83	4.80	24	4.80	0.00
2.40	93	2.41	0.23	2.40	48	2.40	0.00
1.20	186	1.20	0.23	1.20	96	1.20	0.00
0.60	373	0.60	-0.04	0.60	192	0.60	0.00
0.30	746	0.30	-0.04	0.30	384	0.30	0.00

Architecture

Figure 27 displays the architecture of the I^{2}C Controller.

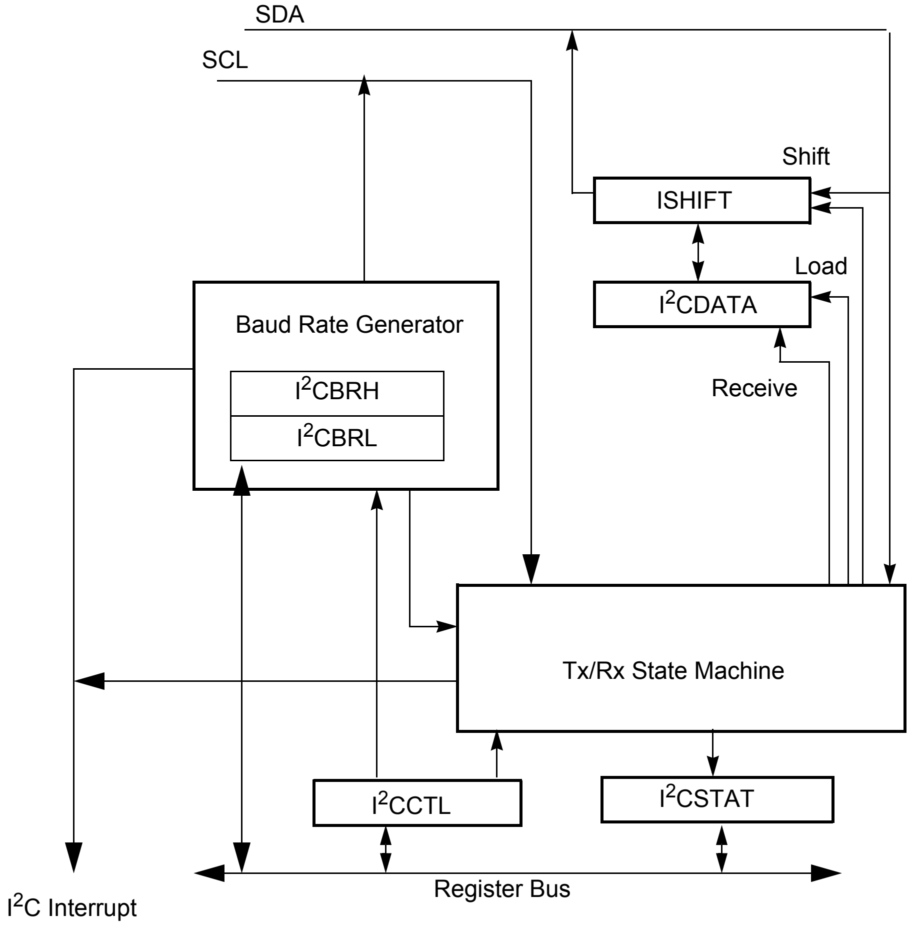


Figure 27. I^{2}C Controller Block Diagram

Operation

The I^{2}C Controller operates in MASTER mode to transmit and receive data. Only a single master is supported. Arbitration between two masters must be accomplished in software. I^{2}C supports the following operations:

- Master transmits to a 7-bit slave
- Master transmits to a 10-bit slave

reading the I^{2}C Data register. Once the I^{2}C data register has been read, the I^{2}C reads the next data byte.

Address Only Transaction with a 7-bit Address

In the situation where software determines if a slave with a 7-bit address is responding without sending or receiving data, a transaction can be done which only consists of an address phase. Figure 28 displays this 'address only' transaction to determine if a slave with a 7-bit address will acknowledge. As an example, this transaction can be used after a 'write' has been done to a EEPROM to determine when the EEPROM completes its internal write operation and is once again responding to I^{2}C transactions. If the slave does not Acknowledge, the transaction can be repeated until the slave does Acknowledge.

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

Figure 28. 7-Bit Address Only Transaction Format

Follow the steps below for an address only transaction to a 7-bit addressed slave:

1. Software asserts the `IEN` bit in the I^{2}C Control register.
2. Software asserts the `TXI` bit of the I^{2}C Control register to enable Transmit interrupts.
3. The I^{2}C interrupt asserts, because the I^{2}C Data register is empty (TDRE = 1)
4. Software responds to the `TDRE` bit by writing a 7-bit slave address plus write bit (=0) to the I^{2}C Data register. As an alternative this could be a read operation instead of a write operation.
5. Software sets the START and STOP bits of the I^{2}C Control register and clears the TXI bit.
6. The I^{2}C Controller sends the START condition to the I^{2}C slave.
7. The I^{2}C Controller loads the I^{2}C Shift register with the contents of the I^{2}C Data register.
8. Software polls the STOP bit of the I^{2}C Control register. Hardware deasserts the STOP bit when the address only transaction is completed.
9. Software checks the ACK bit of the I^{2}C Status register. If the slave acknowledged, the ACK bit is = 1. If the slave does not acknowledge, the ACK bit is = 0. The NCKI interrupt does not occur in the not acknowledge case because the STOP bit was set.

If the current ADC Analog Input is not the highest numbered input to be converted, DMA_ADC initiates data conversion in the next higher numbered ADC Analog Input.

Configuring DMA_ADC for Data Transfer

Follow the steps below to configure and enable DMA_ADC:

1. Write the DMA_ADC Address register with the 7 most-significant bits of the Register File address for data transfers.
2. Write to the DMA_ADC Control register to complete the following:
   - Enable the DMA_ADC interrupt request, if desired
   - Select the number of ADC Analog Inputs to convert
   - Enable the DMA_ADC channel

Caution: *When using the DMA_ADC to perform conversions on multiple ADC inputs, the Analog-to-Digital Converter must be configured for SINGLE-SHOT mode. If the `ADC_IN` field in the DMA_ADC Control Register is greater than 000b, the ADC must be in SINGLE-SHOT mode.*

CONTINUOUS mode operation of the ADC can only be used in conjunction with DMA_ADC if the `ADC_IN` field in the DMA_ADC Control Register is reset to 000b to enable conversion on ADC Analog Input 0 only.

DMA Control Register Definitions

DMA*x* Control Register

The DMA*x* Control register (see Table 77 on page 167) enables and selects the mode of operation for DMA*x*.

Table 77. DMA*x* Control Register (DMA*x*CTL)

BITS	7	6	5	4	3	2	1	0
FIELD	DEN	DLE	DDIR	IRQEN	WSEL	RSS		
RESET	0							
R/W	R/W							
ADDR	FB0H, FB8H							

DEN—DMA*x* Enable
0 = DMA*x* is disabled and data transfer requests are disregarded.

Follow the steps below for setting up the ADC and initiating continuous conversion:

1. Enable the desired analog input by configuring the general-purpose I/O pins for alternate function. This disables the digital input and output driver.
2. Write to the ADC Control register to configure the ADC for continuous conversion. The bit fields in the ADC Control register may be written simultaneously:
   - Write to the ANAIN[3:0] field to select one of the 12 analog input sources.
   - Set CONT to 1 to select continuous conversion.
   - Write to the $\overline{\text{VREF}}$ bit to enable or disable the internal voltage reference generator.
   - Set CEN to 1 to start the conversions.
3. When the first conversion in continuous operation is complete (after 5129 system clock cycles, plus the 40 cycles for power-up, if necessary), the ADC control logic performs the following operations:
   - CEN resets to 0 to indicate the first conversion is complete. CEN remains 0 for all subsequent conversions in continuous operation.
   - An interrupt request is sent to the Interrupt Controller to indicate the conversion is complete.
4. Thereafter, the ADC writes a new 10-bit data result to {ADCD_H[7:0], ADCD_L[7:6]} every 256 system clock cycles. An interrupt request is sent to the Interrupt Controller when each conversion is complete.
5. To disable continuous conversion, clear the CONT bit in the ADC Control register to 0.

DMA Control of the ADC

The Direct Memory Access (DMA) Controller can control operation of the ADC including analog input selection and conversion enable. For more information on the DMA and configuring for ADC operations, see Direct Memory Access Controller on page 165.

Figure 43 displays the typical active mode current consumption while operating at 25 ºC versus the system clock frequency. All GPIO pins are configured as outputs and driven High.

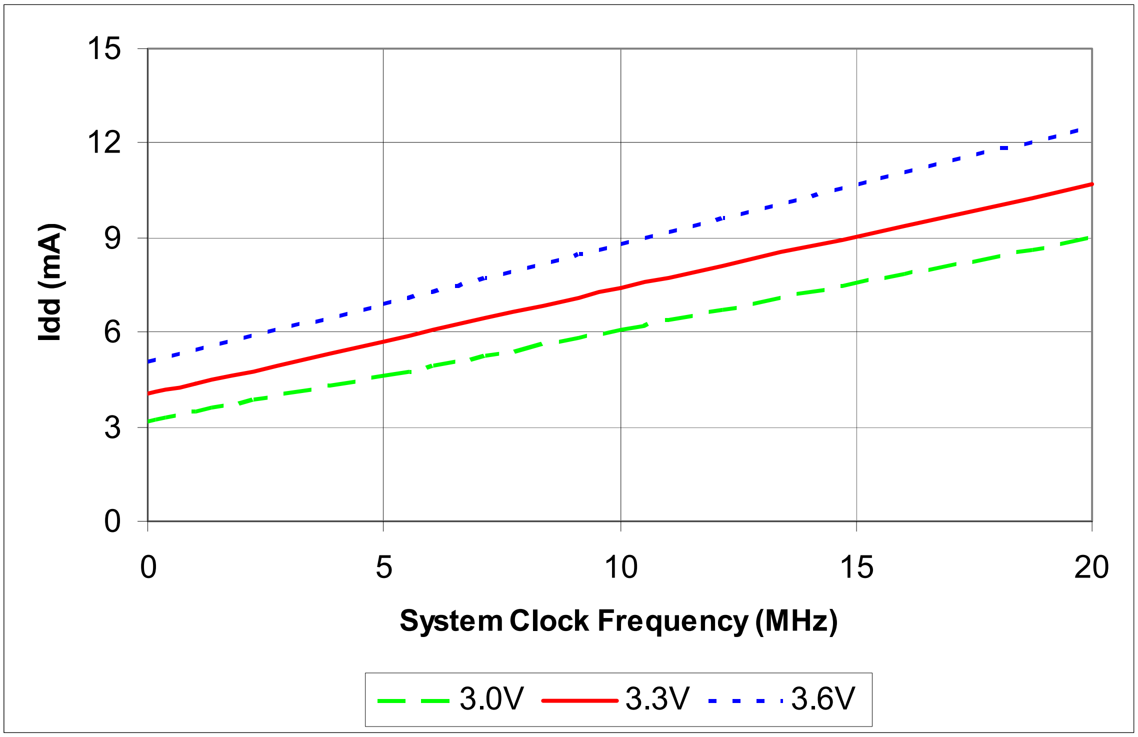


Figure 43. Typical Active Mode Idd Versus System Clock Frequency

AC Characteristics

The section provides information on the AC characteristics and timing. All AC timing information assumes a standard load of 50 pF on all outputs. Table 113 lists the 64K Series AC characteristics and timing.

Table 113. AC Characteristics

		V_{DD} = 3.0–3.6V T_A = –40 °C to 125 °C			
Symbol	**Parameter**	**Minimum**	**Maximum**	**Units**	**Conditions**
F_{sysclk}	System Clock Frequency	–	20.0	MHz	Read-only from Flash memory.
		0.032768	20.0	MHz	Program or erasure of the Flash memory.
F_{XTAL}	Crystal Oscillator Frequency	0.032768	20.0	MHz	System clock frequencies below the crystal oscillator minimum require an external clock driver.
T_{XIN}	Crystal Oscillator Clock Period	50	–	ns	T_{CLK} = 1/F_{sysclk}
T_{XINH}	System Clock High Time	20		ns	
T_{XINL}	System Clock Low Time	20		ns	
T_{XINR}	System Clock Rise Time	–	3	ns	T_{CLK} = 50 ns. Slower rise times can be tolerated with longer clock periods.
T_{XINF}	System Clock Fall Time	–	3	ns	T_{CLK} = 50 ns. Slower fall times can be tolerated with longer clock periods.

Figure 57 and Table 121 provide timing information for UART pins for the case where the Clear To Send input signal ($\overline{\text{CTS}}$) 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 $\overline{\text{DE}}$. $\overline{\text{DE}}$ asserts after the UART Transmit Data Register has been written. $\overline{\text{DE}}$ remains asserted for multiple characters as long as the Transmit Data register is written with the next character before the current character has completed.

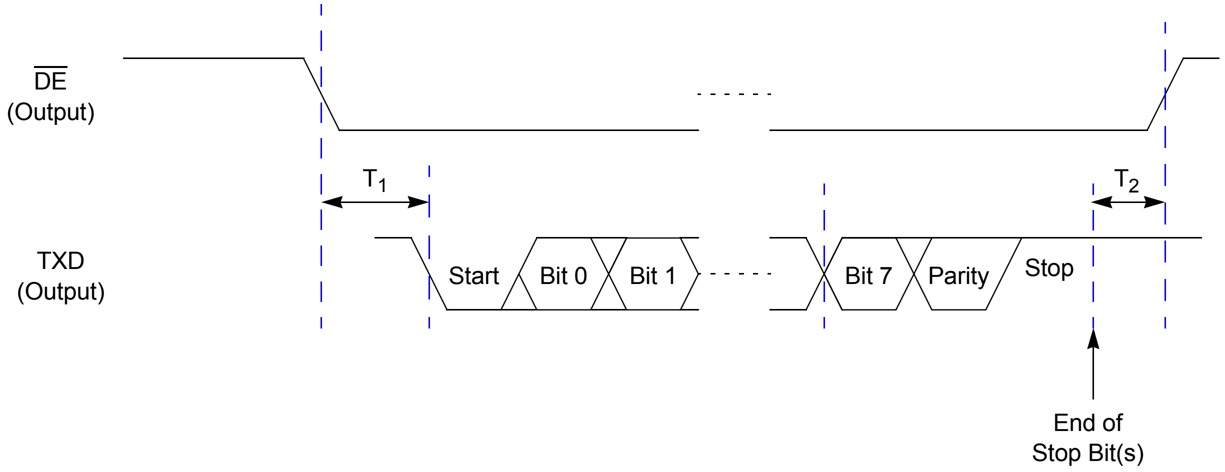


Figure 57. UART Timing without $\overline{\text{CTS}}$

Table 121. UART Timing without $\overline{\text{CTS}}$

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