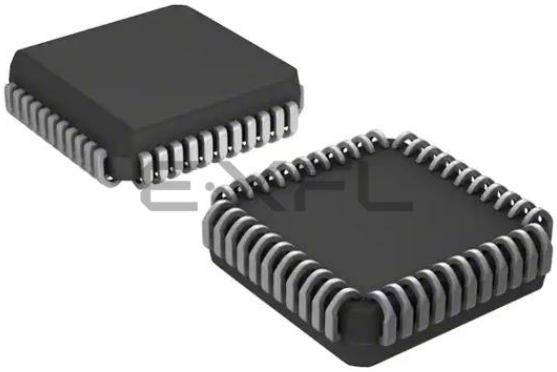

Welcome to E-XFL.COM

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	24KB (24K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K 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-LCC (J-Lead)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f2421vn020eg

Signal and Pin Descriptions

The Z8 Encore! XP F64xx Series product are available in a variety of packages styles and pin configurations. This chapter describes the signals and available pin configurations for each of the package styles. For information about physical package specifications, see the Packaging chapter on page 286.

Available Packages

Table 2 identifies the package styles that are available for each device within the Z8 Encore! XP F64xx Series product line.

Table 2. Z8 Encore! XP F64xx Series Package Options

Part Number	40-Pin PDIP	44-Pin LQFP	44-Pin PLCC	64-Pin LQFP	68-Pin PLCC	80-Pin QFP
Z8F1621	X	X	X			
Z8F1622				X	X	
Z8F2421	X	X	X			
Z8F2422				X	X	
Z8F3221	X	X	X			
Z8F3222				X	X	
Z8F4821	X	X	X			
Z8F4822				X	X	
Z8F4823						X
Z8F6421	X	X	X			
Z8F6422				X	X	
Z8F6423						X

Pin Configurations

Figures 2 through 7 display the pin configurations for all of the packages available in the Z8 Encore! XP F64xx Series. For signal descriptions, see Table 3 on page 14.

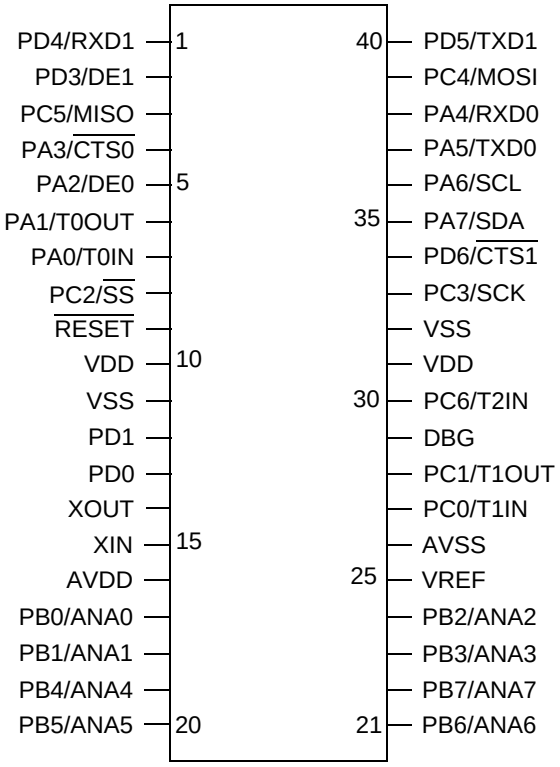

Figure 2. Z8 Encore! XP F64xx Series in 40-Pin Dual Inline Package (PDIP)

Note: Timer 3 and T2OUT are not supported in the 40-pin PDIP package.

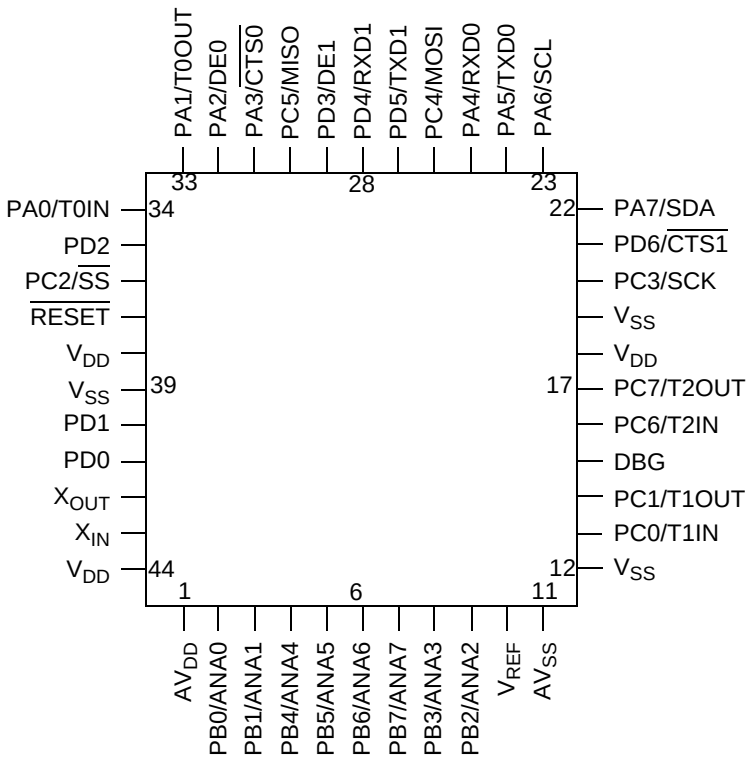

Figure 4. Z8 Encore! XP F64xx Series in 44-Pin Low-Profile Quad Flat Package (LQFP)

Note: Timer 3 is not available in the 44-pin LQFP package.

System Reset

During a system reset, the Z8 Encore! XP F64xx Series devices are held in Reset for 66 cycles of the Watchdog Timer oscillator followed by 16 cycles of the system clock. At the beginning of Reset, all GPIO pins are configured as inputs.

During Reset, the eZ8 CPU and on-chip peripherals are idle; however, the on-chip crystal oscillator and Watchdog Timer oscillator continue to run. The system clock begins operating following the Watchdog Timer oscillator cycle count. The eZ8 CPU and on-chip peripherals remain idle through the 16 cycles of the system clock.

Upon Reset, control registers within the Register File that have a defined Reset value are loaded with their reset values. Other control registers (including the Stack Pointer, Register Pointer, and Flags) and general-purpose RAM are undefined following Reset. 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.

Reset Sources

Table 9 lists the reset sources as a function of the operating mode. The text following provides more detailed information about the individual Reset sources. A Power-On Reset/Voltage Brown-Out event always takes priority over all other possible reset sources to ensure a full system reset occurs.

Table 9. Reset Sources and Resulting Reset Type

Operating Mode	Reset Source	Reset Type
NORMAL or HALT modes	Power-On Reset/Voltage Brown-Out	system reset
	Watchdog Timer time-out when configured for Reset	system reset
	$\overline{\text{RESET}}$ pin assertion	system reset
	On-Chip Debugger initiated Reset (OCDCTL[0] set to 1)	system reset except the On-Chip Debugger is unaffected by the reset
STOP Mode	Power-On Reset/Voltage Brown-Out	system reset
	$\overline{\text{RESET}}$ pin assertion	system reset
	DBG pin driven Low	system reset

If TPOL is set to 0, the ratio of the PWM output High time to the total period is calculated using the following equation:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{Reload Value} - \text{PWM Value}}{\text{Reload Value}} \times 100$$

If TPOL is set to 1, the ratio of the PWM output High time to the total period is calculated using the following equation:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{PWM Value}}{\text{Reload Value}} \times 100$$

CAPTURE Mode

In CAPTURE Mode, the current timer count value is recorded when the appropriate external timer input transition occurs. The Capture count value is written to the Timer PWM High and Low Byte Registers. The timer input is the system clock. The TPOL bit in the Timer Control 1 Register determines if the Capture occurs on a rising edge or a falling edge of the timer input signal. When the capture event occurs, an interrupt is generated and the timer continues counting.

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.

Observe the following procedure 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 appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.

since the previous pulse was detected). This gives the endec a sampling window of minus four baud rate clocks to plus eight baud rate clocks around the expected time of an incoming pulse. If an incoming pulse is detected inside this window this process is repeated. If the incoming data is a logical 1 (no pulse), the endec returns to the initial state and waits for the next falling edge. As each falling edge is detected, the endec clock counter is reset, resynchronizing the endec to the incoming signal. This action allows the endec to tolerate jitter and baud rate errors in the incoming data stream. Resynchronizing the endec does not alter the operation of the UART, which ultimately receives the data. The UART is only synchronized to the incoming data stream when a start bit is received.

Infrared Encoder/Decoder Control Register Definitions

All infrared endec configuration and status information is set by the UART control registers as defined in the UART Control Register Definitions section on page 98.

Caution: To prevent spurious signals during IrDA data transmission, set the IREN bit in the UARTx Control 1 Register to 1 to enable the Infrared Encoder/Decoder before enabling the GPIO Port alternate function for the corresponding pin.

SDA and SCL Signals

I²C sends all addresses, data and acknowledge signals over the SDA line, most significant bit first. SCL is the common clock for the I²C Controller. When the SDA and SCL pin alternate functions are selected for their respective GPIO ports, the pins are automatically configured for open-drain operation.

The master (I²C) is responsible for driving the SCL clock signal, although the clock signal can become skewed by a slow slave device. During the low period of the clock, the slave pulls the SCL signal Low to suspend the transaction. The master releases the clock at the end of the low period and notices that the clock remains low instead of returning to a High level. When the slave releases the clock, the I²C Controller continues the transaction. All data is transferred in bytes and there is no limit to the amount of data transferred in one operation. When transmitting data or acknowledging read data from the slave, the SDA signal changes in the middle of the low period of SCL and is sampled in the middle of the High period of SCL.

I²C Interrupts

The I²C Controller contains four sources of interrupts—Transmit, Receive, Not Acknowledge and baud rate generator. These four interrupt sources are combined into a single interrupt request signal to the Interrupt Controller. The transmit interrupt is enabled by the IEN and TXI bits of the Control Register. The Receive and Not Acknowledge interrupts are enabled by the IEN bit of the Control Register. The baud rate generator interrupt is enabled by the BIRQ and IEN bits of the Control Register.

Not Acknowledge interrupts occur when a Not Acknowledge condition is received from the slave or sent by the I²C Controller and neither the start or stop bit is set. The Not Acknowledge event sets the NCKI bit of the I²C Status Register and can only be cleared by setting the start or stop bit in the I²C Control Register. When this interrupt occurs, the I²C Controller waits until either the stop or start bit is set before performing any action. In an interrupt service routine, the NCKI bit should always be checked prior to servicing transmit or receive interrupt conditions because it indicates the transaction is being terminated.

Receive interrupts occur when a byte of data has been received by the I²C Controller (master reading data from slave). This procedure sets the RDRF bit of the I²C Status Register. The RDRF bit is cleared by reading the I²C Data Register. The RDRF bit is set during the acknowledge phase. The I²C Controller pauses after the acknowledge phase until the receive interrupt is cleared before performing any other action.

Transmit interrupts occur when the TDRE bit of the I²C Status Register sets and the TXI bit in the I²C Control Register is set. transmit interrupts occur under the following conditions when the transmit data register is empty:

- The I²C Controller is enabled

Bit	Description (Continued)
[2] IRQA	**DMA_ADC Interrupt Request Indicator** This bit is automatically reset to 0 each time a read from this register occurs. 0 = DMA_ADC is not the source of the interrupt from the DMA Controller. 1 = DMA_ADC completed transfer of data from the last ADC analog input and generated an interrupt.
[1] IRQ1	**DMA1 Interrupt Request Indicator** This bit is automatically reset to 0 each time a read from this register occurs. 0 = DMA1 is not the source of the interrupt from the DMA Controller. 1 = DMA1 completed transfer of data to/from the End Address and generated an interrupt.
[0] IRQ0	**DMA0 Interrupt Request Indicator** This bit is automatically reset to 0 each time a read from this register occurs. 0 = DMA0 is not the source of the interrupt from the DMA Controller. 1 = DMA0 completed transfer of data to/from the End Address and generated an interrupt.

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 about the DMA and configuring for ADC operations, see the Direct Memory Access Controller chapter on page 150.

ADC Control Register Definitions

This section defines the features of the following ADC Control registers.

ADC Control Register: see page 165

ADC Data High Byte Register: see page 167

ADC Data Low Bits Register: see page 168

ADC Control Register

The ADC Control Register selects the analog input channel and initiates the analog-to-digital conversion.

Table 87. ADC Control Register (ADCCTL)

Bit	7	6	5	4	3	2	1	0
Field	CEN	Reserved	VREF	CONT	ANAIN[3:0]			
RESET	0		1	0				
R/W	R/W							
Address	F70H							

Bit	Description
[7] CEN	**Conversion Enable** 0 = Conversion is complete. Writing a 0 produces no effect. The ADC automatically clears this bit to 0 when a conversion has been completed. 1 = Begin conversion. Writing a 1 to this bit starts a conversion. If a conversion is already in progress, the conversion restarts. This bit remains 1 until the conversion is complete.
[6]	**Reserved** This bit is reserved and must be programmed to 0.
[5] VREF	**Voltage Reference** 0 = Internal voltage reference generator enabled. The V_{REF} pin should be left unconnected (or capacitively coupled to analog ground) if the internal voltage reference is selected as the ADC reference voltage. 1 = Internal voltage reference generator disabled. An external voltage reference must be provided through the V_{REF} pin.

Table 92. Z8 Encore! XP F64xx Series Information Area Map

Flash Memory Address (Hex)	Function
FE00H–FE3FH	Reserved
FE40H–FE53H	Part Number 20-character ASCII alphanumeric code Left-justified and filled with zeros
FE54H–FFFFH	Reserved

Operation

The Flash Controller provides the proper signals and timing for the Byte Programming, Page Erase, and Mass Erase operations within Flash memory. The Flash Controller contains a protection mechanism, via the Flash Control Register (FCTL), to prevent accidental programming or erasure. The following subsections provide details about the Lock, Unlock, Sector Protect, Byte Programming, Page Erase and Mass Erase operations.

Timing Using the Flash Frequency Registers

Before performing a program or erase operation in Flash memory, you must first configure the Flash Frequency High and Low Byte registers. The Flash Frequency registers allow programming and erasure of the Flash with system clock frequencies ranging from 20kHz through 20MHz (the valid range is limited to the device operating frequencies).

The Flash Frequency High and Low Byte registers combine to form a 16-bit value, FFREQ, to control timing for Flash program and erase operations. The 16-bit Flash Frequency value must contain the system clock frequency in kHz. This value is calculated using the following equation:.

$$\text{FFREQ}[15:0] = \frac{\text{System Clock Frequency (Hz)}}{1000}$$

Caution: Flash programming and erasure are not supported for system clock frequencies below 20kHz, above 20MHz, or outside of the devices' operating frequency range. The Flash Frequency High and Low Byte registers must be loaded with the correct value to ensure proper Flash programming and erase operations.

For more information about bypassing the Flash Controller, refer to the Third Party Flash Programming Support for Z8 Encore! MCUs Application Note (AN0117), which is available for download at www.zilog.com.

Flash Controller Behavior in Debug Mode

The following changes in Flash Controller behavior occur when the Flash Controller is accessed using the On-Chip Debugger:

- The Flash Write Protect option bit is ignored
- The Flash Sector Protect Register is ignored for programming and erase operations
- Programming operations are not limited to the page selected in the Page Select Register
- Bits in the Flash Sector Protect Register can be written to one or zero
- The second write of the Page Select Register to unlock the Flash Controller is not necessary
- The Page Select Register can be written when the Flash Controller is unlocked
- The Mass Erase command is enabled through the Flash Control Register

! **Caution:** For security reasons, the Flash Controller allows only a single page to be opened for write/erase operations. When writing multiple Flash pages, the Flash Controller must go through the unlock sequence again to select another page.

Flash Control Register Definitions

This section defines the features of the following Flash Control registers.

Flash Control Register: see page 175

Flash Status Register: see page 177

Page Select Register: see page 177

Flash Sector Protect Register: see page 178

Flash Frequency High and Low Byte Registers: see page 179

Flash Control Register

The Flash Control Register, shown in Table 93, unlocks the Flash Controller for programming and erase operations, or to select the Flash Sector Protect Register.

```
DBG ← Size[7:0]
DBG ← 1-256 data bytes
```

Read Register (09H). The Read Register command reads data from the Register File. Data can be read 1-256 bytes at a time (256 bytes can be read by setting size to zero). If the device is not in DEBUG Mode or if the Read Protect option bit is enabled, this command returns FFH for all the data values.

```
DBG ← 09H
DBG ← {4'h0,Register Address[11:8]
DBG ← Register Address[7:0]
DBG ← Size[7:0]
DBG 1-256 data bytes
```

Write Program Memory (0AH). The Write Program Memory command writes data to program memory. This command is equivalent to the LDC and LDCI instructions. Data can be written 1-65536 bytes at a time (65536 bytes can be written by setting size to zero). The on-chip Flash Controller must be written to and unlocked for the programming operation to occur. If the Flash Controller is not unlocked, the data is discarded. If the device is not in DEBUG Mode or if the Read Protect option bit is enabled, the data is discarded.

```
DBG ← 0AH
DBG ← Program Memory Address[15:8]
DBG ← Program Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG ← 1-65536 data bytes
```

Read Program Memory (0BH). The Read Program Memory command reads data from program memory. This command is equivalent to the LDC and LDCI instructions. Data can be read 1-65536 bytes at a time (65536 bytes can be read by setting size to zero). If the device is not in DEBUG Mode or if the Read Protect option bit is enabled, this command returns FFH for the data.

```
DBG ← 0BH
DBG ← Program Memory Address[15:8]
DBG ← Program Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG 1-65536 data bytes
```

Write Data Memory (0CH). The Write Data Memory command writes data to Data Memory. This command is equivalent to the LDE and LDEI instructions. Data can be written 1-65536 bytes at a time (65536 bytes can be written by setting size to zero). If the device is not in DEBUG Mode or if the Read Protect option bit is enabled, the data is discarded.

```
DBG ← 0CH
DBG ← Data Memory Address[15:8]
DBG ← Data Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG ← 1-65536 data bytes
```

Bit	Description (Continued)
[5] RPEN	**Read Protect Option Bit Enabled** 0 = The Read Protect option bit is disabled (1). 1 = The Read Protect option bit is enabled (0), disabling many OCD commands.
[4:0]	**Reserved** These bits are reserved and must be programmed to 00000.

Figure 46 displays the maximum HALT Mode current consumption across the full operating temperature range of the device and plotted opposite the system clock frequency. All GPIO pins are configured as outputs and driven High.

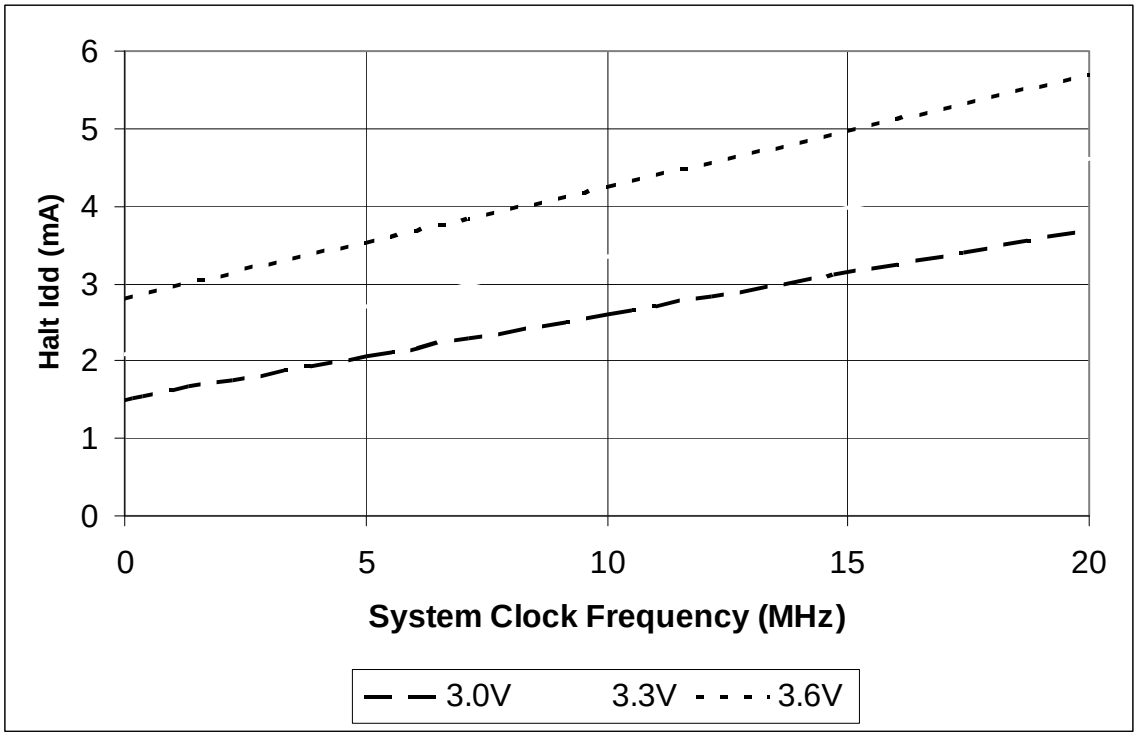

Figure 46. Maximum HALT Mode I_{CC} vs. System Clock Frequency

I²C Timing

Figure 55 and Table 120 provide timing information for I²C pins.

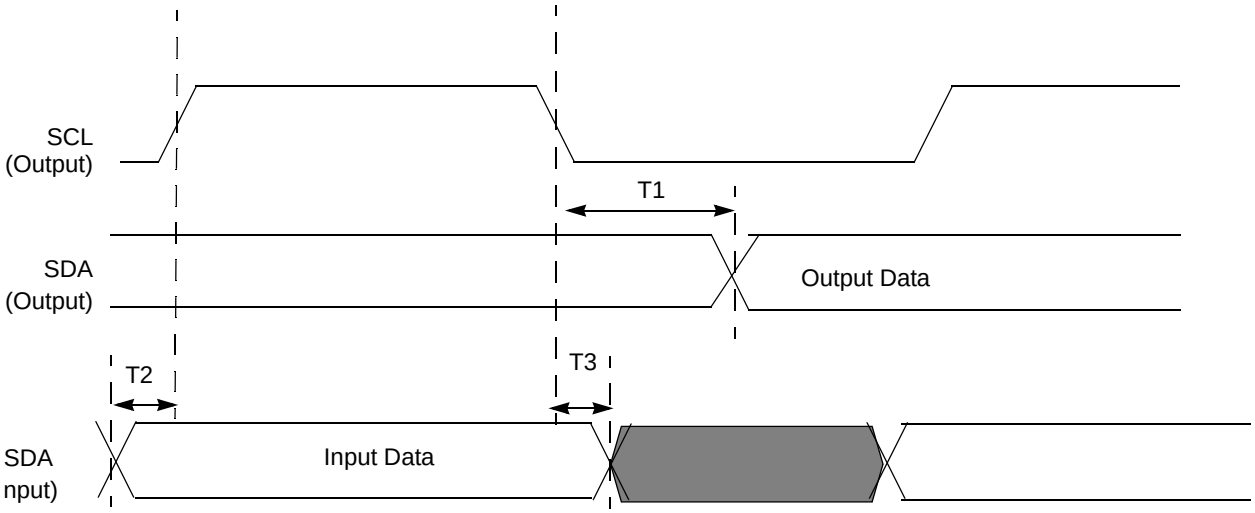

Figure 55. I²C Timing

Table 120. I²C Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
I²C			
T_1	SCL Fall to SDA output delay	SCL period/4	
T_2	SDA Input to SCL rising edge Setup Time	0	
T_3	SDA Input to SCL falling edge Hold Time	0	

Table 128. Arithmetic Instructions (Continued)

Mnemonic	Operands	Instruction
MULT	dst	Multiply
SBC	dst, src	Subtract with Carry
SBCX	dst, src	Subtract with Carry using Extended Addressing
SUB	dst, src	Subtract
SUBX	dst, src	Subtract using Extended Addressing

Table 129. Bit Manipulation Instructions

Mnemonic	Operands	Instruction
BCLR	bit, dst	Bit Clear
BIT	p, bit, dst	Bit Set or Clear
BSET	bit, dst	Bit Set
BSWAP	dst	Bit Swap
CCF	—	Complement Carry Flag
RCF	—	Reset Carry Flag
SCF	—	Set Carry Flag
TCM	dst, src	Test Complement Under Mask
TCMX	dst, src	Test Complement Under Mask using Extended Addressing
TM	dst, src	Test Under Mask
TMX	dst, src	Test Under Mask using Extended Addressing

Table 130. Block Transfer Instructions

Mnemonic	Operands	Instruction
LDCI	dst, src	Load Constant to/from program memory and Auto-Increment addresses
LDEI	dst, src	Load External Data to/from Data Memory and Auto-Increment addresses

Hex Address: F61

Table 196. SPI Control Register (SPICTL)

Bit	7	6	5	4	3	2	1	0
Field	IRQE	STR	BIRQ	PHASE	CLKPOL	WOR	MMEN	SPIEN
RESET	0							
R/W	R/W							
Address	F61H							

Hex Address: F62

Table 197. 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.								

Hex Address: F63

Table 198. SPI Mode Register (SPIMODE)

Bit	7	6	5	4	3	2	1	0
Field	Reserved		DIAG	NUMBITS[2:0]			SSIO	SSV
RESET	0							
R/W	R		R/W					
Address	F63H							

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							

SPIBRL register 128, 266
SPICTL register 123, 265
SPIDATA register 123, 264
SPIMODE register 126, 265
SPISTAT register 124, 265
SRA 235
src 229
SRL 235
SRP 233
stack pointer 229
status register, I2C 143
STOP 233
STOP mode 35, 233
Stop Mode Recovery
    sources 33
    using a GPIO port pin transition 34
    using watchdog timer time-out 34
SUB 232
subtract 232
subtract - extended addressing 232
subtract with carry 232
subtract with carry - extended addressing 232
SUBX 232
SWAP 235
swap nibbles 235
symbols, additional 229
system and core resets 30

T

TCM 232
TCMX 232
test complement under mask 232
test complement under mask - extended addressing 232
test under mask 232
test under mask - extended addressing 232
timer signals 16
timers 6, 63
    architecture 63
    block diagram 64
    capture mode 69, 79
    capture/compare mode 71, 79
    compare mode 70, 79
    continuous mode 65, 79
    counter mode 66
    counter modes 79
    gated mode 71, 79
    one-shot mode 64, 79
    operating mode 64
    PWM mode 67, 79
    reading the timer count values 72
    reload high and low byte registers 75
    timer control register definitions 73
    timer output signal operation 73
timers 0-3
    control 0 registers 77
    control 1 registers 78
    high and low byte registers 73, 76
TM 232
TMX 232
transmit
    IrDA data 111
transmit interrupt 131
transmitting UART data-interrupt-driven method 91
transmitting UART data-polled method 90
TRAP 234

U

UART 5, 88
    architecture 88
    asynchronous data format without/with parity 90
    baud rate generator 99
    baud rates table 108
    control register definitions 99
    controller signals 16
    interrupts 97
    multiprocessor mode 94
    receiving data using interrupt-driven method 93
    receiving data using the polled method 92
    transmitting data using the interrupt-driven method 91
    transmitting data using the polled method 90
    x baud rate high and low registers 106
    x control 0 and control 1 registers 103

Customer Support

To share comments, get your technical questions answered or report issues you may be experiencing with our products, please visit Zilog's Technical Support page at http://support.zilog.com.

To learn more about this product, find additional documentation or to discover other facets about Zilog product offerings, please visit the Zilog Knowledge Base at http://zilog.com/kb or consider participating in the Zilog Forum at http://zilog.com/forum.

This publication is subject to replacement by a later edition. To determine whether a later edition exists, please visit the Zilog website at http://www.zilog.com.