Zilog - Z8F0880SJ020EG Datasheet

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LED, LVD, POR, PWM, Temp Sensor, WDT
Number of I/O	23
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f0880sj020eg

Table of Contents

Revision History iii

List of Figures xv

List of Tables xviii

Chapter 1. Overview 1

1.1. Features 1

1.2. Part Selection Guide 2

1.3. Block Diagram 3

1.4. An Overview of the eZ8 CPU and its Peripherals 4

1.4.1. General-Purpose Input/Output 4

1.4.2. Flash Controller 4

1.4.3. Non-Volatile Data Storage 5

1.4.4. Internal Precision Oscillator 5

1.4.5. Crystal Oscillator 5

1.4.6. Secondary Oscillator 5

1.4.7. 10-Bit Analog-to-Digital Converter 5

1.4.8. Low-Power Operational Amplifier 5

1.4.9. Analog Comparator 5

1.4.10. Temperature Sensor 6

1.4.11. Low-Voltage Detector 6

1.4.12. Enhanced SPI 6

1.4.13. UART with LIN 6

1.4.14. Master/Slave I2C 6

1.4.15. Timers 6

1.4.16. Multi-Channel Timer 7

1.4.17. Interrupt Controller 7

1.4.18. Reset Controller 7

1.4.19. On-Chip Debugger 7

1.4.20. Direct LED Drive 7

1.5. Acronyms and Expansions 8

Chapter 2. Pin Description 10

2.1. Available Packages 10

2.2. Pin Configurations 10

2.3. Signal Descriptions 14

2.4. Pin Characteristics 17

Chapter 3. Address Space 19
3.1. Register File 19
3.2. Program Memory 20
3.3. Data Memory 21
3.4. Flash Information Area 21
Chapter 4. Register Map 23
Chapter 5. Reset, Stop Mode Recovery and Low-Voltage Detection 31
5.1. Reset Types 31
5.2. Reset Sources 33
5.2.1. Power-On Reset 34
5.2.2. Voltage Brown-Out Reset 35
5.2.3. Watchdog Timer Reset 36
5.2.4. External Reset Input 37
5.2.5. External Reset Indicator 37
5.2.6. On-Chip Debugger Initiated Reset 37
5.3. Stop Mode Recovery 37
5.3.1. Stop Mode Recovery Using Watchdog Timer Time-Out 38
5.3.2. Stop Mode Recovery Using Timer Interrupt 38
5.3.3. Stop Mode Recovery Using Comparator Interrupt 39
5.3.4. Stop Mode Recovery Using GPIO Port Pin Transition 39
5.3.5. Stop Mode Recovery Using External RESET Pin 39
5.4. Low-Voltage Detection 39
5.5. Reset Register Definitions 40
Chapter 6. Low-Power Modes 42
6.1. STOP Mode 42
6.2. HALT Mode 43
6.3. Peripheral-Level Power Control 43
6.4. Power Control Register Definitions 44
Chapter 7. General-Purpose Input/Output 46
7.1. GPIO Port Availability by Device 46
7.2. Architecture 47
7.3. GPIO Alternate Functions 47
7.4. Direct LED Drive 48
7.5. Shared Reset Pin 48
7.6. Crystal Oscillator Override 48
7.7. 32 kHz Secondary Oscillator Override 48

List of Tables

Table 1. Z8 Encore! XP F1680 Series Part Selection Guide 2
Table 2. F1680 Series MCU Acronyms 8
Table 3. Z8 Encore! XP F1680 Series Package Options 10
Table 4. Signal Descriptions 14
Table 5. Pin Characteristics (20-, 28-, 40- and 44-pin Devices). 17
Table 6. F1680 Series MCU Program Memory Maps 20
Table 7. F1680 Series MCU Flash Memory Information Area Map 22
Table 8. Register File Address Map 23
Table 9. Reset and Stop Mode Recovery Characteristics and Latency 32
Table 10. Reset Sources and Resulting Reset Type 33
Table 11. Stop Mode Recovery Sources and Resulting Action 38
Table 12. Reset Status Register (RSTSTAT) 40
Table 13. Reset Status Per Event 41
Table 14. Power Control Register 0 (PWRCTL0) 44
Table 15. Setup Condition for LVD and VBO Circuits in Different Operation Modes 45
Table 16. Port Availability by Device and Package Type 46
Table 17. Port Alternate Function Mapping, 20-Pin Parts1,2. 49
Table 18. Port Alternate Function Mapping, 28-Pin Parts1,2. 51
Table 19. Port Alternate Function Mapping, 40-/44-Pin Parts1,2 54
Table 20. GPIO Port Registers and Subregisters 58
Table 21. Port A–E GPIO Address Registers (PxADDR) 59
Table 22. Port A–E Control Registers (PxCTL). 60
Table 23. Port A–E Data Direction Subregisters (PxDD) 60
Table 24. Port A–E Alternate Function Subregisters (PxAF). 61
Table 25. Port A–E Output Control Subregisters (PxOC) 62
Table 26. Port A–E High Drive Enable Subregisters (PxHDE) 62
Table 27. Port A–E Stop Mode Recovery Source Enable Subregisters (PxSMRE). . . 63
Table 28. Port A–E Pull-Up Enable Subregisters (PxPUE) 63

Chapter 1. Overview

Zilog's F1680 Series of MCUs is based on Zilog's advanced 8-bit eZ8 CPU core. This microcontroller, a member of the Z8 Encore! XP® product line, is optimized for low-power applications and supports 1.8V to 3.6V of low-voltage operation with extremely low Active, Halt and Stop Mode currents, plus it offers a wide assortment of speed and low-power options. In addition, the feature-rich analog and digital peripherals of the Z8 Encore! XP F1680 Series of MCUs makes them suitable for a variety of applications including safety and security, utility metering, digital power supervisory, hand-held electronic devices and general motor control applications.

For simplicity, the remainder of this document refers to the entire Z8 Encore! XP F1680 Series of MCUs as the F1680 Series MCU.

1.1. Features

Key features of the F1680 Series MCU include:

- 20MHz eZ8 CPU core
- 8KB, 16KB, or 24KB Flash memory with in-circuit programming capability
- 1KB or 2KB Register RAM
- 1KB Program RAM for program code shadowing and data storage (optional)
- 128B or 256B Non-Volatile Data Storage (NVDS)
- Up to 8-Channel, 10-bit Analog-to-Digital Converter (ADC)
- On-chip Temperature Sensor
- Up to two on-chip analog comparators (20-pin and 28-pin packages contain only one)
- On-chip Low-Power Operational Amplifier (LPO)
- Two full-duplex 9-bit UART ports with the support of Local Interconnect Network (LIN) protocol (20-pin and 28-pin packages contain only one)
- Infrared Data Association (IrDA)-compliant infrared encoders/decoders, integrated with UARTs
- Enhanced Serial Peripheral Interface (SPI) controller (except 20-pin packages)
- I^{2}C controller which supports Master/Slave modes
- Three enhanced 16-bit Timers with Capture, Compare and PWM capability
- Additional two basic 16-bit timers with interrupt (shared as UART Baud Rate Generator)

1.4.3. Non-Volatile Data Storage

Non-Volatile Data Storage (NVDS) is a hybrid hardware/software scheme to implement byte-programmable data memory and is capable of over 100,000 write cycles.

1.4.4. Internal Precision Oscillator

The internal precision oscillator (IPO) is a trimmable clock source which requires no external components. You can select IPO frequency from one of eight frequencies (43.2 kHz to 11.0592 MHz) and is available with factory-trimmed calibration data.

1.4.5. Crystal Oscillator

The crystal oscillator circuit provides highly accurate clock frequencies using an external crystal, ceramic resonator, or RC network.

1.4.6. Secondary Oscillator

The secondary oscillator is a low-power oscillator, which is optimized for use with a 32 kHz watch crystal. It can be used as timer/counter clock source in any mode.

1.4.7. 10-Bit Analog-to-Digital Converter

The Analog-to-Digital Converter (ADC) converts an analog input signal to a 10-bit binary number. The ADC supports up to eight analog input sources multiplexed with GPIO ports.

1.4.8. Low-Power Operational Amplifier

The low-power operational amplifier (LPO) is a general-purpose operational amplifier primarily targeted for current sense applications. The LPO output can be internally routed to the ADC or externally to a pin.

1.4.9. Analog Comparator

The analog comparator compares the signal at an input pin with either an internal programmable voltage reference or a second-input pin. The comparator output is used to either drive an output pin or to generate an interrupt.

Table 2. F1680 Series MCU Acronyms (Continued)

Abbreviations/ Acronyms	Expansions
LSB	Least-significant byte
PWM	Pulse-Width Modulation
CI	Channel Interrupt
TI	Timer Interrupt
Endec	Encoder/Decoder
I^2S	Inter IC Sound
TDM	Time division multiplexing
TTL	Transistor-Transistor Logic
SAR	Successive Approximation Register

reconfigure the oscillator control block such that the correct system clock source is enabled and selected.

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 Reset Status Register is set to 1. Table 11 lists the Stop Mode Recovery sources and resulting actions. The text following provides more detailed information about 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)
	Interrupt from Timer enabled for STOP Mode operation	Stop Mode Recovery followed by interrupt (if interrupts are enabled)
	Interrupt from Comparator enabled for STOP Mode operation	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

5.3.1. Stop Mode Recovery Using Watchdog Timer Time-Out

If the WDT times out during STOP Mode, the device undergoes a Stop Mode Recovery sequence. In the Reset Status Register, the WDT and stop bits are set to 1. If the WDT is configured to generate an interrupt on time-out and the F1680 Series MCU is configured to respond to interrupts. The eZ8 CPU services the WDT interrupt request following the normal Stop Mode Recovery sequence.

5.3.2. Stop Mode Recovery Using Timer Interrupt

If a Timer with 32K crystal enabled for STOP Mode operation interrupts during STOP Mode, the device undergoes a Stop Mode Recovery sequence. In the Reset Status Register, the stop bit is set to 1. If the F1680 Series MCU is configured to respond to interrupts, the

Table 19. Port Alternate Function Mapping, 40-/44-Pin Parts[1,2] (Continued)

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Register AFS1
Port C	PC0	Reserved		AFS1[0]: 0
		ANA4/C0INP/LED	ADC or Comparator 0 Input (P), or LED drive	AFS1[0]: 1
	PC1	Reserved		AFS1[1]: 0
		ANA5/C0INN/LED	ADC or Comparator 0 Input (N), or LED Drive	AFS1[1]: 1
	PC2	$\overline{\text{SS}}$	SPI Slave Select	AFS1[2]: 0
		ANA6/LED	ADC Analog Input or LED Drive	AFS1[2]: 1
	PC3	MISO	SPI Master In Slave Out	AFS1[3]: 0
		LED	LED drive	AFS1[3]: 1
	PC4	MOSI	SPI Master Out Slave In	AFS1[4]: 0
		LED	LED Drive	AFS1[4]: 1
	PC5	SCK	SPI Serial Clock	AFS1[5]: 0
		LED	LED Drive	AFS1[5]: 1
	PC6	T2IN/$\overline{\text{T2OUT}}$	Timer 2 Input/Timer2 Output Complement	AFS1[6]: 0
		LED	LED Drive	AFS1[6]: 1
	PC7	T2OUT	Timer 2 Output	AFS1[7]: 0
		LED	LED Drive	AFS1[7]: 1

Notes:

1. Because there are at most two choices of alternate functions for some pins in Ports A–C, the Alternate Function Set Register (AFS2) is implemented but not used to select the function. The alternate function selection must also be enabled, as described in the Port A–E Alternate Function Subregisters section on page 61.
2. Because there is only one alternate function for each Port D and Port E pin, the Alternate Function Set registers are not implemented for Ports D and E. Enabling the alternate function selections automatically enables the associated alternate function, as described in the Port A–E Alternate Function Subregisters section on page 61.
3. This timer function is only available in the 44-pin package; its alternate functions are reserved in the 40-pin package.

7.11.5. Port A–E Output Control Subregisters

The Port A–E Output Control Subregister, shown in Table 25, is accessed through the Port A–E Control Register by writing 03H to the Port A–E Address Register. Setting the bits in the Port A–E Output Control subregisters to 1 configures the specified port pins for open-drain operation. These subregisters affect the pins directly and, as a result, alternate functions are also affected.

Table 25. Port A–E Output Control Subregisters (PxOC)

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	POC7	POC6	POC5	POC4	POC3	POC2	POC1	POC0
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
Address	If 03H in Port A–E Address Register, accessible through the Port A–E Control Register							

Bit	Description
[7:0] POC	**Port Output Control** These bits function independently of the alternate function bit and always disable the drains if set to 1. 0 = The drains are enabled for any output mode (unless overridden by the alternate function). 1 = The drain of the associated pin is disabled (open-drain mode).

7.11.6. Port A–E High Drive Enable Subregisters

The Port A–E High Drive Enable Subregister, shown in Table 26, is accessed through the Port A–E Control Register by writing 04H to the Port A–E Address Register. Setting the bits in the Port A–E High Drive Enable subregisters to 1 configures the specified port pins for high-current output drive operation. The Port A–E High Drive Enable Subregister affects the pins directly and, as a result, alternate functions are also affected.

Table 26. Port A–E High Drive Enable Subregisters (PxHDE)

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	PHDE7	PHDE6	PHDE5	PHDE4	PHDE3	PHDE2	PHDE1	PHDE0
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
Address	If 04H in Port A–E Address Register, accessible through the Port A–E Control Register							

Bit	Description
[7:0] PHDE	**Port High Drive Enabled** 0 = The Port pin is configured for standard output current drive. 1 = The Port pin is configured for high output current drive.

9.2.3.10. 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 counts timer clocks up to a 16-bit reload value. 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) on Compare.

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

Observe the following steps to configure a timer for COMPARE Mode and initiate the count:

1. Write to the Timer Control 1 Register to:
   - Disable the timer
   - Configure the timer for COMPARE Mode
   - Set the prescale valu.
   - Set the initial logic level (High or Low) for the Timer Output alternate function, if required
2. Write to the Timer Control 2 Register to choose the timer clock source.
3. Write to the Timer Control 0 Register to set the timer interrupt configuration field TICONFIG.
4. Write to the Timer High and Low Byte registers to set the starting count value.
5. Write to the Timer Reload High and Low Byte registers to set the Compare value.
6. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
7. When using the Timer Output function, configure the associated GPIO port pin for the Timer Output alternate function.
8. Write to the Timer Control 1 Register to enable the timer and initiate counting.

In COMPARE Mode, the timer clock always provides the timer input. The Compare time is calculated using the following equation:

9.2.3.11. GATED Mode

In GATED Mode, the timer counts only when the Timer Input signal is in its active state (asserted) as determined by the TPOL bit in the Timer Control 1 Register. When the Timer Input signal is asserted, counting begins. A Timer Interrupt is generated when the Timer Input signal is deasserted or a timer reload occurs. To determine if a Timer Input signal

10.7.8. Multi-Channel Timer Channel-y Control Registers

Each channel has a control register to enable the channel, select the input/output polarity, enable channel interrupts and select the channel mode of operation.

Table 78. Multi-Channel Timer Channel Control Register (MCTCHyCTL)[1]

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	CHEN	CHPOL	CHIEN	CHUE	Reserved	CHOP		
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
Address	See note 2.							
Notes: 1. y = A, B, C, D. 2. If 02H, 03H, 04H and 05H are in the Subaddress Register, they are accessible through Subregister 2.								

Bit	Description
[7] CHEN	**Channel Enable** 0 = Channel is disabled. 1 = Channel is enabled.
[6] CHPOL	**Channel Input/Output Polarity** Operation of this bit is a function of the current operating method of the channel. **ONE-SHOT Operation** When the channel is disabled, the Channel Output signal is set to the value of this bit. When the channel is enabled, the Channel Output signal toggles for one system clock on reaching the Channel Capture/Compare Register value. **CONTINUOUS COMPARE Operation** When the channel is disabled, the Channel Output signal is set to the value of this bit. When the channel is enabled, the Channel Output signal toggles (from Low to High or High to Low) on reaching the Channel Capture/Compare Register value. **PWM OUTPUT Operation** 0 = Channel Output is forced Low when the channel is disabled. When enabled, the Channel Output is forced High on Channel Capture/Compare Register value match and forced Low on reaching the Timer Reload Register value (modulo mode) or counting down through the channel Capture/Compare register value (count up/down mode). 1 = Channel Output is forced Low when the channel is disabled. When enabled, the Channel Output is forced High on Channel Capture/Compare Register value match and forced Low on reaching the Timer Reload Register value (modulo mode) or counting down through the channel Capture/Compare register value (count up/down mode). **CAPTURE Operation** 0 = Count is captured on the rising edge of the Channel Input signal. 1 = Count is captured on the falling edge of the Channel Input signal.

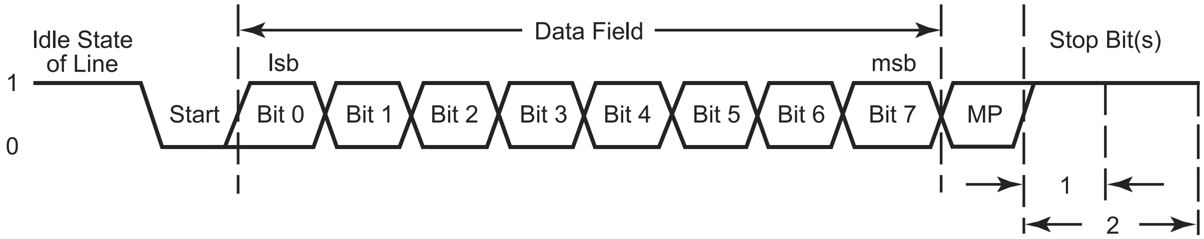

Figure 23. LIN-UART Asynchronous MULTIPROCESSOR Mode Data Format

In MULTIPROCESSOR (9-bit) Mode, the Parity bit location (9th bit) becomes the Multiprocessor control bit. The LIN-UART Control 1 and Status 1 registers provide MULTIPROCESSOR (9-bit) Mode control and status information. If an automatic address matching scheme is enabled, the LIN-UART Address Compare register holds the network address of the device.

12.1.9.1. MULTIPROCESSOR Mode Receive Interrupts

When MULTIPROCESSOR (9-bit) Mode is enabled, the LIN-UART processes only frames addressed to it. To determine whether a frame of data is addressed to the LIN-UART can be made in hardware, software or a combination of the two, depending on the multiprocessor configuration bits. In general, the address compare feature reduces the load on the CPU, because it is not required to access the LIN-UART when it receives data directed to other devices on the multinode network. The following three MULTIPROCESSOR Modes are available in hardware:

- Interrupt on all address bytes
- Interrupt on matched address bytes and correctly framed data bytes
- Interrupt only on correctly framed data bytes

These modes are selected with MPMD[1:0] in the LIN-UART Control 1 Register. For all MULTIPROCESSOR Modes, bit MPEN of the LIN-UART Control 1 Register must be set to 1.

The first scheme is enabled by writing `01b` to MPMD[1:0]. In this mode, all incoming address bytes cause an interrupt, while data bytes never cause an interrupt. The interrupt service routine checks the address byte which triggered the interrupt. If it matches the LIN-UART address, the software clears MPMD[0]. At this point, each new incoming byte interrupts the CPU. The software determines the end of the frame and checks for it by reading the MPRX bit of the LIN-UART Status 1 Register for each incoming byte. If MPRX=1, a new frame begins. If the address of this new frame is different from the LIN-UART's address, then MPMD[0] must be set to 1 by software, causing the LIN-UART

12.3.5. LIN-UART Control 0 Register

The LIN-UART Control 0 Register, shown in Table 89, configures the basic properties of LIN-UART's transmit and receive operations. A more detailed discussion of each bit follows the table.

Table 89. LIN-UART Control 0 Register (U0CTL0 = F42H)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	TEN	REN	CTSE	PEN	PSEL	SBRK	STOP	LBEN
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
Address	F42H, F4AH							

Note: R/W = Read/Write.

Bit	Description
[7] TEN	**Transmit Enable** This bit enables or disables the transmitter. The enable is also controlled by the $\overline{\text{CTS}}$ signal and the CTSE bit. If the $\overline{\text{CTS}}$ signal is Low and the CTSE bit is 1, the transmitter is enabled. 0 = Transmitter disabled. 1 = Transmitter enabled.
[6] REN	**Receive Enable** This bit enables or disables the receiver. 0 = Receiver disabled. 1 = Receiver enabled.
[5] CTSE	**Clear To Send Enable** 0 = The $\overline{\text{CTS}}$ signal has no effect on the transmitter. 1 = The LIN-UART recognizes the $\overline{\text{CTS}}$ signal as an enable control for the transmitter.
[4] PEN	**Parity Enable** This bit enables or disables parity. Even or odd is determined by the PSEL bit. 0 = Parity is disabled. This bit is overridden by the MPEN bit. 1 = The transmitter sends data with an additional parity bit and the receiver receives an additional parity bit.
[3] PSEL	**Parity Select** 0 = Even parity is sent as an additional parity bit for the transmitter/receiver. 1 = Odd parity is sent as an additional parity bit for the transmitter/receiver.

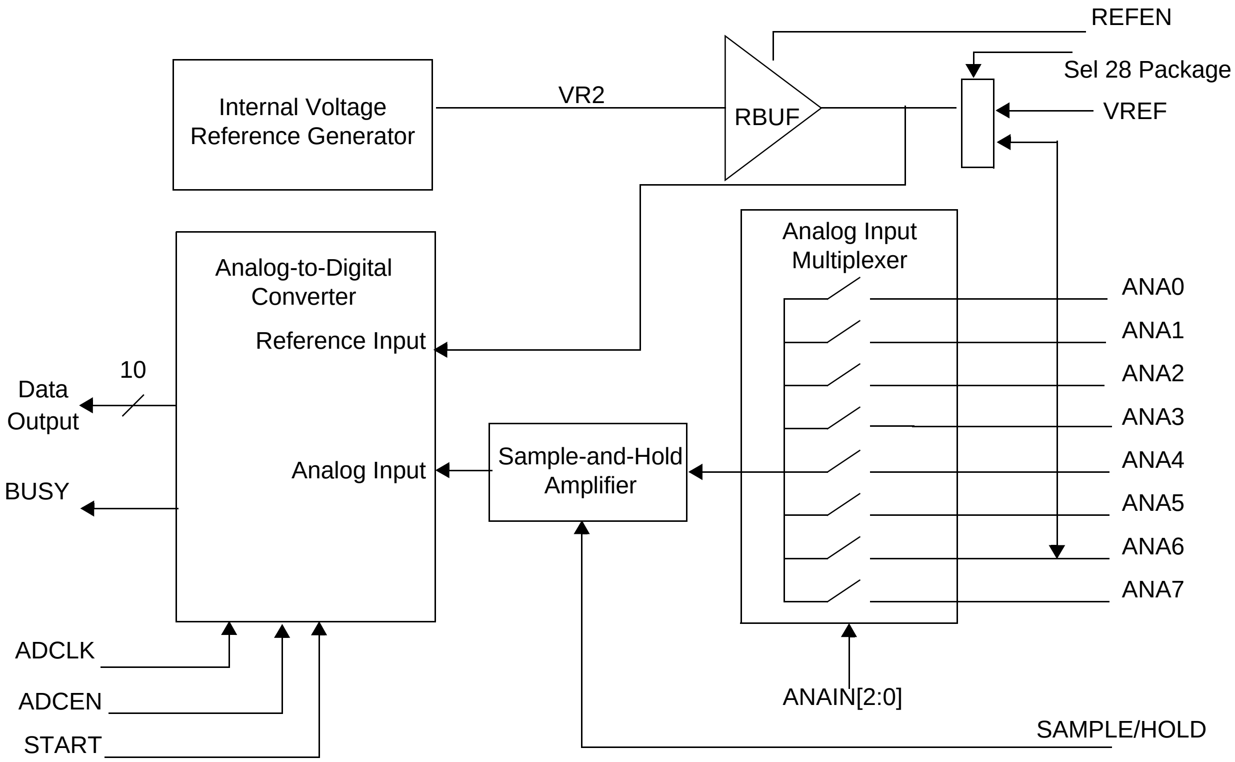

Figure 30. Analog-to-Digital Converter Block Diagram

14.2.1. ADC Timing

Each ADC measurement consists of 3 phases:

1. Input sampling (programmable, minimum of 1.8 µs).
2. Sample-and-hold amplifier settling (programmable, minimum of 0.5 µs).
3. Conversion is 13 ADCLK cycles.

Figure 31 displays the timing of an ADC conversion.

Chapter 16. Enhanced Serial Peripheral Interface

The Enhanced Serial Peripheral Interface (ESPI) supports the Serial Peripheral Interface (SPI) and other synchronous serial interface modes, such as Inter-IC Sound (I^2S) and time division multiplexing (TDM). ESPI includes the following features:

- Full-duplex, synchronous, character-oriented communication
- Four-wire interface ($\overline{SS}$, SCK, MOSI and MISO)
- Data Shift Register is buffered to enable high throughput
- MASTER Mode transfer rates up to a maximum of one-half the system clock frequency
- SLAVE Mode transfer rates up to a maximum of one-eighth the system clock frequency
- Error detection
- Dedicated Programmable Baud Rate Generator
- Data transfer control via polling, interrupt

16.1. Architecture

The ESPI is a full-duplex, synchronous, character-oriented channel that supports a four-wire interface (serial clock, transmit data, receive data and slave select). The ESPI block consists of a shift register, data buffer register, a Baud Rate (clock) Generator, control/status registers and a control state machine. Transmit and receive transfers are in synch as there is a single shift register for both transmitting and receiving data. Figure 33 displays a diagram of the ESPI block.

16.4.5. ESPI Status Register

The ESPI Status Register, shown in Table 113, indicates the current state of the ESPI. All bits revert to their Reset state if the ESPI is disabled.

Table 113. ESPI Status Register (ESPISTAT)

Bits	7	6	5	4	3	2	1	0
Field	TDRE	TUND	COL	ABT	ROVR	RDRNE	TFST	SLAS
Reset	1	0	0	0	0	0	0	1
R/W	R	R/W*	R/W*	R/W*	R/W*	R	R	R
Address	F64H							
Note: R/W* = Read access. Write a 1 to clear the bit to 0.								

Bit	Description
[7] TDRE	**Transmit Data Register Empty** 0 = Transmit Data Register is full or ESPI is disabled. 1 = Transmit Data Register is empty. A write to the ESPI (Transmit) Data Register clears this bit.
[6] TUND	**Transmit Underrun** 0 = A Transmit Underrun error has not occurred. 1 = A Transmit Underrun error has occurred.
[5] COL	**Collision** 0 = A multi-Master collision (mode fault) has not occurred. 1 = A multi-Master collision (mode fault) has occurred.
[4] ABT	**SLAVE Mode Transaction Abort** This bit is set if the ESPI is configured in SLAVE Mode, a transaction is occurring and $\overline{SS}$ deasserts before all bits of a character have been transferred as defined by the NUMBITS field of the ESPIMODE register. This bit can also be set in SLAVE Mode by an SCK monitor timeout (MMEN = 0, BRGCTL = 1). 0 = A SLAVE Mode transaction abort has not occurred. 1 = A SLAVE Mode transaction abort has occurred.
[3] ROVR	**Receive Overrun** 0 = A Receive Overrun error has not occurred. 1 = A Receive Overrun error has occurred.
[2] RDRNE	**Receive Data Register Not Empty** 0 = Receive Data Register is empty. 1 = Receive Data Register is not empty.

17.2.5.1. Master Arbitration

If a Master loses arbitration during the address byte it releases the SDA line, switches to SLAVE Mode and monitors the address to determine if it is selected as a Slave. If a Master loses arbitration during the transmission of a data byte, it releases the SDA line and waits for the next stop or start condition.

The Master detects a loss of arbitration when a 1 is transmitted but a 0 is received from the bus in the same bit-time. This loss occurs if more than one Master is simultaneously accessing the bus. Loss of arbitration occurs during the address phase (two or more Masters accessing different slaves) or during the data phase, when the masters are attempting to Write different data to the same Slave.

When a Master loses arbitration, the software is informed by means of the Arbitration Lost interrupt. The software can repeat the same transaction at a later time.

A special case can occur when a Slave transaction starts just before the software attempts to start a new master transaction by setting the start bit. In this case, the state machine enters its Slave states before the start bit is set and as a result the I^{2}C controller will not arbitrate. If a Slave address match occurs and the I^{2}C controller receives/transmits data, the start bit is cleared and an Arbitration Lost interrupt is asserted. The software can minimize the chance of this instance occurring by checking the busy bit in the I2CSTATE Register before initiating a Master transaction. If a slave address match does not occur, the Arbitration Lost interrupt will not occur and the start bit will not be cleared. The I^{2}C controller will initiate the master transaction after the I^{2}C bus is no longer busy.

17.2.5.2. Master Address-Only Transactions

It is sometimes preferable to perform an address-only transaction to determine if a particular slave device is able to respond. This transaction can be performed by monitoring the ACKV bit in the I2CSTATE Register after the address has been written to the I2CDATA Register and the start bit has been set. After the ACKV bit is set, the ACK bit in the I2CSTATE Register determines if the slave is able to communicate. The stop bit must be set in the I2CCTL Register to terminate the transaction without transferring data. For a 10-bit slave address, if the first address byte is acknowledged, the second address byte should also be sent to determine if the preferred Slave is responding.

Another approach is to set both the stop and start bits (for sending a 7-bit address). After both bits have been cleared (7-bit address has been sent and transaction is complete), the ACK bit can be read to determine if the Slave has acknowledged. For a 10-bit Slave, set the stop bit after the second TDRE interrupt (which indicates that the second address byte is being sent).

17.2.5.3. Master Transaction Diagrams

In the following transaction diagrams, the shaded regions indicate the data that is transferred from the Master to the Slave and the unshaded regions indicate the data that is

Chapter 18. Comparator

The Z8 Encore! XP F1680 Series devices feature two same general purpose comparators that compares two analog input signals. For each comparator, a GPIO (C0INP/C1INP) pin provides the positive comparator input, the negative input (C0INN/C1INN) can be taken from either an external GPIO pin or an internal reference. The output of each comparator is available as an interrupt source or can be routed to an external pin using the GPIO multiplex. Features for each comparator include:

- Two inputs which are connected using the GPIO multiplex (MUX)
- One input can be connected to a programmable internal reference
- One input can be connected to the on-chip temperature sensor
- Output can trigger timer counting
- Output can be either an interrupt source or an output to an external pin
- Operation in STOP Mode

18.1. Operation

One of the comparator inputs can be connected to an internal reference which is a user-selectable reference that is user-programmable with 200mV resolution.

The comparator can be powered down to save supply current or to continue to operate in STOP Mode. For details, see the Power Control Register 0 section on page 44. In STOP Mode, the comparator interrupt (if enabled) automatically initiates a Stop Mode Recovery and generates an interrupt request. In the Reset Status Register (see page 40), the stop bit is set to 1. Also, the Comparator request bit in the Interrupt Request 1 Register (see page 74) is set. Following completion of the Stop Mode Recovery, and if interrupts are enabled, the CPU responds to the interrupt request by fetching the comparator interrupt vector.

! **Caution:** Because of the propagation delay of the comparator, spurious interrupts can result after enabling the comparator. Zilog recommends not enabling the comparator without first disabling interrupts, then waiting for the comparator output to settle.

The following code example shows how to safely enable the comparator:

```
di
ldx CMP0,r0
nop
```

Chapter 21. Flash Option Bits

Programmable Flash option bits allow user configuration of certain aspects of Z8 Encore! XP F1680 Series MCU operation. The feature configuration data is stored in Flash program memory and are read during Reset. The features available for control through the Flash option bits include:

- Watchdog Timer time-out response selection–interrupt or System Reset
- Watchdog Timer enabled at Reset
- The ability to prevent unwanted read access to user code in Program Memory
- The ability to prevent accidental programming and erasure of all or a portion of the user code in Program Memory
- VBO configuration-always enabled or disabled during STOP Mode to reduce STOP Mode power consumption
- LVD voltage threshold selection
- Oscillator mode selection for high, medium and low-power crystal oscillators or an external RC oscillator
- Factory trimming information for the IPO and Temperature Sensor

21.1. Operation

This section describes the types of option bits and their configuration in the Option Configuration registers.

21.1.1. Option Bit Configuration by Reset

Each time the Flash option bits are programmed or erased, the device must be Reset for the change to take effect. During any Reset operation (System Reset or Stop Mode Recovery), the Flash option bits are automatically read from Flash Program Memory and written to the Option Configuration registers. These Option Configuration registers control operation of the devices within the Z8 Encore! XP F1680 Series MCU. Option bit control is established before the device exits Reset and the eZ8 CPU begins code execution. The Option Configuration registers are not part of the Register File and are not accessible for read or write access.

Table 163. On-Chip Debugger Commands (Continued)

Debug Command	Command Byte	Enabled when not in DEBUG mode?	Disabled by Read Protect Option Bit
Write Test Mode Register	F0H	–	Flash Test mode is not be enabled if the Information Area Write Protect option bit is enabled.
Read Test Mode Register	F1H	–	–
Write Option Bit registers	F2H	–	Cannot write the Read Protect or Information Area Write Protect option bits.
Read Option Bit registers	F3H	–	–
Note: Unlisted command byte values are reserved.			

In the following bulleted list of OCD Commands, data and commands sent from the host to the On-Chip Debugger are identified by `DBG` ← `Command/Data`. Data returned from the On-Chip Debugger to the host is identified by `DBG` → `Data`.

Read Revision (00H). The Read OCD Revision command determines the version of the On-Chip Debugger. If OCD commands are added, removed or changed, this revision number changes.

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

Write OCD Counter Register (01H). The Write OCD Counter Register command writes the data that follows to the OCDCNTR Register. If the device is not in DEBUG mode, the data is discarded.

```
DBG ← 01H
DBG ← OCDCNTR[15:8]
DBG ← OCDCNTR[7:0]
```

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

```
DBG ← 02H
DBG → OCDSTAT[7:0]
```

Read OCD Counter Register (03H). The OCD Counter Register can be used to count system clock cycles in between breakpoints, generate a BRK when it counts down to 0, or generate a BRK when its value matches the Program Counter. Because this register is really a down counter, the returned value is inverted when this register is read so the returned result appears to be an up counter. If the device is not in DEBUG mode, this command returns `FFFFH`.