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	24MHz
Connectivity	DALI, DMX, I²C, LINbus, SPI, UART/USART, USB
Peripherals	DMA, LCD, LVD, POR, PWM, WDT
Number of I/O	67
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3.75K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 12x12b; D/A 1x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-LQFP
Supplier Device Package	80-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f6482at024xk2246

Table 6. F6482 Series Program Memory Maps (Continued)

Program Memory Address (Hex)	Function
Z8F1682 and Z8F1681 Products (Continued)	
0000–0001	Flash option bits
0002–0003	Reset vector
0004–0005	WDT interrupt vector
0006–0007	Illegal instruction trap
0008–0047	Interrupt vectors*
0048–004B	Oscillator fail traps*
004C–3FFF	Program Flash
F000	NVDS byte read
F3FD	NVDS byte write
Note: *See Table 51 on page 128 for a list of interrupt vectors and traps.	

3.3. Data Memory

F6482 Series MCUs do not use the eZ8 CPU's 64KB Data Memory address space.

3.4. Flash Information Area

Table 7 lists the F6482 Series Flash Information Area. This 1KB space consists of two pages and is accessed by setting bit 7 of the Flash Page Select Register to 1. When access is enabled, the Flash Information Area is mapped into Program Memory and overlays the `FC00h` to `FFFFh` address range. When Information Area access is enabled, all reads from these Program Memory addresses return the Information Area data rather than the Program Memory data. Access to the Flash Information Area is read-only.

Table 7. F6482 Series Flash Memory Information Area Map

Program Memory Address (Hex)	Function
FC00–FC3F	Zilog option bits.
FC40–FC53	Part number: a 20-character ASCII alphanumeric code, left-justified and filled with Fh.
FC54–FC5F	Reserved.
FC60–FC7F	Zilog calibration data.
FC80–FFFF	Reserved.

Table 20. Port Alternate Function Mapping (Z8Fxx82 64-Pin Parts) (Continued)

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Subregisters
Port J[3]	PJ0	SEG13	LCD Segment	N/A
	PJ1	SEG14	LCD Segment	
	PJ2	SEG17	LCD Segment	
	PJ3	SEG18	LCD Segment	

Notes

1. Because there are at most two choices of alternate function for some pins of Ports A, B, D, F, G and H, the Alternate Function Set register AFS2 is not implemented. Additionally, alternate function selection, as described in the Port A–J Alternate Function Subregisters (see page 88), must also be enabled.
2. The alternate function selection for Port C, as described in the Port A–J Alternate Function Subregisters (see page 88), must also be enabled.
3. Because there is only a single alternate function for each Port J pin, the Alternate Function Set registers are not implemented for Port J. Additionally, alternate function selection, as described in the Port A–J Alternate Function Subregisters (see page 88), must also be enabled.

7.10.11. Port A–J Input Data Registers

Reading from the Port A–J Input Data registers, shown in Table 33, returns the sampled values from the corresponding port pins. The Port A–J Input Data registers are read-only. The value returned for any unused ports is 0. Unused ports include those missing on the packages other than the 80-pin package.

Table 33. Port A–J Input Data Registers (P*x*IN)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	PIN7	PIN6	PIN5	PIN4	PIN3	PIN2	PIN1	PIN0
Reset	X	X	X	X	X	X	X	X
R/W	R	R	R	R	R	R	R	R
Address	Port A @ FD2h, Port B @ FD6h, Port C @ FDAh, Port D @ FDEh, Port E @ FE2h, Port F @ FE6h, Port G @ FEAh, Port H @ FEEh, Port J @ FBEh							
Note: *x* = A, B, C, D, E, F, G, H, or J.								

Bit	**Description**
[7:0] P*x*IN	**Port Input Data** Sampled data from the corresponding port pin input. 0: Input data is logical 0 (Low). 1: Input data is logical 1 (High).

Chapter 8. Clock System

The F6482 Series devices use seven possible clock sources, each user-selectable:

- On-chip Internal Precision Oscillator (IPO)
- On-chip Digitally Controlled Oscillator (DCO)
- On-chip High Frequency Crystal Oscillator (HFXO) using off-chip crystal or resonator
- On-chip Low Frequency Crystal Oscillator (LFXO) using off-chip crystal
- Two external clock drives
- On-chip Watchdog Timer Oscillator (WTO)

Two internal clock multiplication systems are available:

- An on-chip Frequency Locked Loop (FLL) which locks the DCO to Peripheral Clock (PCLK, an internal clock)
- An on-chip Phase Locked Loop (PLL) which can clock the USB and/or source System Clock and locks to either the HFXO or external clock drive

Four internal clocks exist and are configured to the clock sources as follows:

- System Clock (SYSCLK) is the clock input to the CPU as well as other system functions. The five possible clock sources for System Clock are:
  - Peripheral Clock (PCLK), a derived clock
  - DCO which can be locked to PCLK using the FLL
  - HFXO or External clock drive (based on PLL clock source select)
  - PLL clock sourced by the HFXO or external clock drive
  - WTO

- Peripheral Clock (PCLK) is the 32.768 kHz clock input to the DCO and can be selected to clock certain peripherals. The three possible clock sources for PCLK are:
  - IPO
  - LFXO
  - External clock 2 drive

- The WTO is the clock input to the Watchdog Timer (WDT) and can be selected to clock certain peripherals

Table 36. Peripheral Clock Sources and Usage

PCLK Sources	Characteristics	Required Setup
Internal Precision Oscillator (IPO)	• 32.768 kHz nominal • ± 2% accuracy with factory trim • No external components required	• Unlock the Clock Control registers • Enable the IPO • Switch System Clock sources as described in the PCLK Source Switching section on page 102 • Select PCLK as the clock source for any desired peripheral(s) in the appropriate peripheral control register(s)
External Clock 2 Drive	• 32.768 kHz when used as PCLK source • Accuracy dependent on external clock source	• Write GPIO registers to configure PA2 pin for external clock function, CLK2IN • Apply external clock signal to GPIO • Unlock the Clock Control registers • Switch System Clock sources as described in the PCLK Source Switching section on page 102 • Select PCLK as the clock source for any desired peripheral(s) in the appropriate peripheral control register(s)

8.2.2.1. PCLK Source Switching

PCLK source switching provides glitch free operation in that changing the clock source or prescale setting will not cause clock glitches. In accomplishing glitch free clock switching operation, there is a delay from the writing of PCKSEL to the actual switching from the currently active clock source to the new, desired clock source. To switch from one PCLK source to another, observe the following procedure:

1. Unlock the clock control registers.
2. Write to the appropriate clock control registers to configure the desired clock source.
3. Enable the desired clock source.
4. Wait for the newly enabled clock source to stabilize.
5. Write CLKCTL1 to select PCLK.
6. After the System Clock source has been switched, disable any unnecessary clock sources, if desired.
7. The clock control registers can be locked by clearing CSTAT.

8.2.3. PLL Clock Selection

The PLL Clock (PLL_{CLK}) is driven by the on-chip PLL. The two possible sources for the PLL are the High Frequency Crystal Oscillator (HFXO) and the External Clock Drive.

Bit	Description
[7:0] PLLNDIV	**Phase Locked Loop (PLL) Feedback Division Ratio** Disable the PLL by clearing PLLEN prior to changing PLLNDIV. 00h: 1. 01h: 2. 02h: 3. 03h: 4. • • • FCh: 253. FDh: 254. FEh: 255. FFh: 256.

Table 73. Triggered One-Shot Mode Initialization Example (Continued)

Register	Value	Comment
T0H	00h	Timer starting value=0001h.
T0L	01h	
T0RH	ABh	Timer reload value=ABCDh.
T0RL	CDh	
PAADDR	02h	Selects Port A Alternate Function Register.
PACTL[1:0]	11b	PACTL[0] enables Timer 0 Input Alternate function if also selected by the Alternate Function Set 1 Register. PACTL[1] enables Timer 0 Output Alternate function if also selected by the Alternate Function Set 1 Register.
PAADDR	07h	Selects Port A Alternate Function Set 1 Register.
PACTL[1:0]	00b	PACTL[0] enables Timer 0 Input Alternate function. PACTL[1] enables Timer 0 Output Alternate function.
ESDADDR	10h	Selects the Timer 0 Input 0 Event System Destination.
ESDCTL	00h	Disconnects the Event System Input0 to Timer 0.
IRQ0ENH[5]	0b	Disables the Timer 0 interrupt.
IRQ0ENL[5]	0b	
T0CTL1	83h	TEN=1 enables the timer. All other bits remain in their appropriate settings.

Note: After receiving an input trigger, Timer 0 will:
1. Count ABCDh timer clocks.
2. Upon Timer 0 reload, generate a single clock cycle active High output pulse on the Timer 0 Output.
3. Wait for the next input trigger event.

10.1.3.3. Dual Input Triggered One-Shot Mode

In Dual Input Triggered One-Shot Mode (TMODE=1010), the first to arrive of the timer Input 0 or Input 1 signals triggers counting. In addition, the input that was first to arrive is recorded by the timer for later use in interrupt and output generation. Two timer output options and four interrupt options are available. Previous and current timer input triggers can be used in the interrupt control. If a trigger event occurs while counting, INPCAP is set to indicate that the interrupt is due to the trigger event. If a reload occurs, INPCAP is cleared to indicate that the interrupt is not due to a trigger event. The timer operates in the following sequence:

1. The Timer idles until a trigger is received. Due to the assertion of an input, both inputs must initially be deasserted to receive a new trigger. The Timer trigger, in essence the first to arrive of the timer Input 0 and Input 1 signals, is taken from the GPIO port pin timer input alternate function or from the Event System. If required, enable the Noise Filter and set the Noise Filter control by writing to the relevant bits in the Noise Filter Control Register. The TPOL bit in the Timer Control 1 Register selects whether the

plished by pulling the bus Low for at least 250 µs but less than 5ms. Transmitting a 00h character is one way to transmit the wake-up message.

If the CPU is in Stop Mode, the UART-LDD is not active and the wake-up message must be detected by a GPIO edge detect Stop-Mode Recovery. The duration of the Stop-Mode Recovery sequence can preclude making an accurate measurement of the wake-up message duration.

If the CPU is in a halt or operational mode, the UART-LDD (if enabled) times the duration of the wake-up and provides an interrupt following the end of the break sequence if the duration is ≥ 3 bit times. The total duration of the wake-up message in bit times can be obtained by reading the RxBreakLength field in the Mode Select and Status Register. After a wake-up message has been detected, the UART-LDD can be placed (by software) either into LIN Master or LIN Slave Wait for Break states, as appropriate. If the break duration exceeds 15 bit times, the RxBreakLength field contains the value Fh. If the UART-LDD is disabled, wake-up message is detected via a port pin interrupt and timed by software. If the device is in Stop Mode, the High to Low transition on the port pin will bring the device out of Stop Mode.

The *LIN Sleep state* is selected by software setting LinState[1:0]=00. The decision to move from an active state to sleep state is based on the LIN messages as interpreted by software.

14.1.10.5. LIN Slave Operation

LIN Slave Mode is selected by setting LMST=0, LSLV=1, ABEN=1 or 0 and LinState[1:0]=01b (Wait for Break state). The LIN slave detects the start of a new message by the break which appears to the slave as a break of at least 11 bit times in duration. The UART-LDD detects the break and generates an interrupt to the CPU. The duration of the break is observable in the RxBreakLength field of the Mode Select and Status Register. A break of less than 11 bit times in duration does not generate a break interrupt when the UART-LDD is in a Wait for Break state. If the break duration exceeds 15 bit times, the RxBreakLength field contains the value Fh.

Following the break, the UART-LDD hardware automatically transits to the *Autobaud state*, where it autobauds by timing the duration of the first 8 bit times of the Synch character as defined in the LIN standard. The duration of the autobaud period is measured by the BRG Counter which will update every 8th system clock cycle between the start bit and the beginning of bit 7 of the autobaud sequence. At the end of the autobaud period, the duration measured by the BRG counter (auto baud period divided by 8) is automatically transferred to the Baud Reload High and Low registers if the ABEN bit of the LIN Control Register is set. If the BRG Counter overflows before reaching the start of bit 7 in the autobaud sequence the Autobaud Overrun Error interrupt occurs, the OE bit in the Status 0 Register is set and the Baud Reload registers are not updated. To autobaud within 2% of the master's baud rate, the slave system clock must be a minimum of 100 times the baud rate. To avoid an autobaud overrun error, the system clock must not be greater than 2^{19}

15.2. ESPI Signals

The four ESPI signals are:

- Master-In/Slave-Out (MISO)
- Master-Out/Slave-In (MOSI)
- Serial Clock (SCK)
- Slave Select ($\overline{\text{SS}}$)

The following paragraphs discuss these signals as they operate in both Master and Slave modes.

15.2.1. Master-In/Slave-Out

The Master-In/Slave-Out (MISO) pin is configured as an input in a master device and as an output in a slave device. Data is transferred most significant bit first. The MISO pin of a slave device is placed in a high-impedance state if the slave is not selected. When the ESPI is not enabled, this signal is in a high-impedance state. The direction of this pin is controlled by the MMEN bit of the ESPI Control Register.

15.2.2. Master-Out/Slave-In

The Master-Out/Slave-In (MOSI) pin is configured as an output in a master device and as an input in a slave device. Data is transferred most significant bit first. When the ESPI is not enabled, this signal is in a high-impedance state. The direction of this pin is controlled by the MMEN bit of the ESPI Control Register.

15.2.3. Serial Clock

The Serial Clock (SCK) synchronizes data movement both in and out of the Shift Register via the MOSI and MISO pins. In Master Mode (MMEN=1), the ESPI's Baud Rate Generator creates the serial clock and drives it out on its SCK pin to the slave devices. In Slave Mode, the SCK pin is an input. Slave devices ignore the SCK signal unless their $\overline{\text{SS}}$ pin is asserted.

The master and slave are each capable of exchanging a character of data during a sequence of NUMBITS clock cycles; see the ESPI 0-1 Mode Registers (ESPIxMODE) on page 299 for details. In both master and slave ESPI devices, data is shifted on one edge of the SCK, and is sampled on the opposite edge where data is stable. SCK phase and polarity is determined by the PHASE and CLKPOL bits in the ESPI Control Register.

15.2.4. Slave Select

The Slave Select signal is a bidirectional framing signal with several modes of operation to support SPI and other synchronous serial interface protocols. Slave Select Mode is

16. The I^{2}C controller sends a repeated start condition.
17. The I^{2}C controller loads the I^{2}C Shift Register with the contents of the I^{2}C Data Register (the third address transfer).
18. The I^{2}C controller sends `11110b`, followed by the two most-significant bits of the slave read address and a 1 (read).
19. The I^{2}C slave sends an Acknowledge by pulling the SDA signal Low during the next High period of SCL.
20. The I^{2}C controller shifts in a byte of data from the slave.
21. The I^{2}C controller asserts the Receive interrupt.
22. The software responds by reading the I^{2}C Data Register. If the next data byte is to be the final byte, the software must set the NAK bit of the I^{2}C Control Register.
23. The I^{2}C controller sends an Acknowledge or Not Acknowledge to the I^{2}C slave, based on the value of the NAK bit.
24. If there are more bytes to transfer, the I^{2}C controller returns to Step 20.
25. The I^{2}C controller generates a NAK interrupt (the NCKI bit in the I2CISTAT Register).
26. The software responds by setting the stop bit of the I^{2}C Control Register.
27. A stop condition is sent to the I^{2}C slave.

16.2.6. Slave Transactions

The following subsections describe read and write transactions to the I^{2}C controller configured for 7-bit and 10-bit slave modes.

16.2.6.1. Slave Address Recognition

The following two slave address recognition options are supported; a description of each follows.

- Slave 7-Bit Address Recognition Mode
- Slave 10-Bit Address Recognition Mode

Slave 7-Bit Address Recognition Mode. If IRM=0 during the address phase and the controller is configured for MASTER/SLAVE or SLAVE 7-BIT ADDRESS Mode, the hardware detects a match to the 7-bit slave address defined in the I2CSLVAD Register and generates the slave address match interrupt (the SAM bit=1 in the I2CISTAT Register). The I^{2}C controller automatically responds during the Acknowledge phase with the value in the NAK bit of the I2CCTL Register.

17.2.4. USB Control Transfers Using Endpoint 0

A control transfer consists of two or three stages:

- Setup stage
- Data stage (optional)
- Status stage

The following describes control write and control read transfers, in addition to the associated status and interrupt bit.

17.2.4.1. Control Write

In the Setup stage of a control transfer, after receiving a Setup token, the USB Module sets the HSNAK bit in the USB Endpoint 0 Control and Status Subregister (USBEP0CS) and the SUTOKIRQ bit in the USB Protocol Interrupt Request Subregister (USBIRQ). A USB interrupt is generated if the SUTOKIEN bit is set in the USB Protocol Interrupt Enable Subregister (USBIEN). Subsequently, if an 8-byte data packet is received correctly, the USB Module sets the SUDAVIRQ bit in the USBIRQ Subregister and a USB interrupt is generated if the SUDAVIEN bit is set in the USBIEN Subregister. The 8-byte data packet can be accessed from the USB Setup Buffer Byte 0–7 subregisters (USBSUx), as described in the Setup Buffer section on page 347.

The Data stage of a control transfer consists of one or more OUT bulk-like transactions. After each correct OUT packet is received during the Data stage, the USB Module sets the OUT0IRQ bit in the USB OUT Interrupt Request Subregister (USBOUTIRQ), and a USB interrupt is generated if the OUT0IEN is set in the USB OUT Interrupt Enable Subregister (USBOUTIEN). The OUT0BC Subregister contains the number of data bytes received in the last OUT transaction. Software should service the interrupt request, then prepare the endpoint for the next transaction by reloading the OUT0BC Subregister with any value, which results in hardware setting the OUTBUSY bit in the USBEP0CS Subregister. Until this preparation task is performed, the USB Controller will NAK subsequent data packets.

The Status stage of a control transfer is the final operation in the sequence. Software should clear the HSNAK bit (by writing a 1 to it) to instruct the USB Module to ACK the Status stage. The USB Module sends the STALL handshake when both HSNAK and STALL bits are set. Prior to the Status stage, after the last successful transaction in the Data stage when all expected bytes of the transfer have been received or sent by the USB Module and a STALL handshake is to be sent for any additional data stage tokens, DSTALL is typically set by software. When DSTALL is set, the USB Module will send a STALL handshake if additional data stage tokens are sent and, if this transmission occurs, the STALL bit will be automatically set so that the USB Module will send a STALL handshake in the Status stage. If DSTALL is set, a token that indicates a transition to the status stage (e.g., an OUT token for an IN endpoint) will not cause a STALL handshake.

A control write transfer example is shown in Figure 56.

17.3.6. USB Interrupt Control Register

The USBIRQCTL Register, shown in Table 175, is used to perform a device-initiated Resume and to manage Resume interrupts.

Table 175. USB Interrupt Control Register (USBIRQCTL)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved						RIRQE	WAKEUP
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W
Address	F5Fh							

Bit	**Description**
[7:2]	**Reserved** These bits are reserved and must be programmed to 000000.
[1] RIRQE	**Resume Interrupt Request Enable** 0: Interrupt only for host-initiated resume. PLLCLK does not need to be running for a host-initiated resume interrupt to occur. 1: Interrupt for both host-initiated resume and device-initiated resume that is using the USB to time Idle state duration. For the latter, the USB resume interrupt is generated once the USB has completed timing the Idle state. See Suspend/Resume on page 352 for details.
[0] WAKEUP	**Wake-Up (Device-Initiated Resume)** 0: Do not perform a device-initiated resume. 1: Start a device-initiated resume. When using the USB Controller to time the USB Idle state, RIRQE should also be set if WAKEUP is set. If using the USB to time the USB Idle state, WAKEUP is cleared by the USB Controller when it is waking up and the USB clock is running. See device-initiated resume (Remote Wake-up) on page 362 for details.

17.3.12. USB IN Interrupt Request Subregister

The USB IN Interrupt Request Subregister, shown in Table 181, indicates the IN endpoint interrupt requests. The USB Module sets INxIRQ when it transmits an IN endpoint x data packet and receives an ACK from the host. The interrupt is cleared by writing a 1 to the corresponding register position.

Table 181. USB IN Interrupt Request Subregister (USBINIRQ)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved				IN3IRQ	IN2IRQ	IN1IRQ	IN0IRQ
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W1*	R/W1*	R/W1*	R/W1*
Address	If USBSA = 29h in the USB Subaddress Register, accessible through the USB Subdata Register							
Note: *R/W1 = Writing a 1 clears this bit.								

Bit	Description
[7:4]	**Reserved** These bits are reserved and must be programmed to 0000.
[3] IN3IRQ	**IN Endpoint 3 Interrupt Request** 0: No interrupt request from IN endpoint 3. 1: Interrupt request from IN endpoint 3.
[2] IN2IRQ	**IN Endpoint 2 Interrupt Request** 0: No interrupt request from IN endpoint 2. 1: Interrupt request from IN endpoint 2.
[1] IN1IRQ	**IN Endpoint 1 Interrupt Request** 0: No interrupt request from IN endpoint 1. 1: Interrupt request from IN endpoint 1.
[0] IN0IRQ	**IN Endpoint 0 Interrupt Request** 0: No interrupt request from IN endpoint 0. 1: Interrupt request from IN endpoint 0.

18.3.9. DMA 0–3 Linked List Descriptor Address High and Low Subregisters

The DMA Linked List Descriptor Address High and Low (DMAxLAH and DMAxLAL) subregisters, shown in Tables 215 and 216, contain the 12-bit linked list descriptor address. These registers have an effect only during linked list operation. Writing DMAx-LAL automatically starts the Linked List DMAx even if DMAxLAH is not written.

Table 215. DMA 0–3 Linked List Descriptor Address High Subregister (DMAxLAH)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved				LAH			
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Address	If DMASA = 8h in the DMAxSA Register, accessible through the DMA 0–3 Subregister							

Bit	Description
[7:4]	**Reserved** These bits are reserved and must be programmed to 0000.
[3:0] LAH	**Linked List Descriptor Address High** LAH and LAL together form the 12-bit address that points to the current linked list descriptor. 0–F: The upper 4 bits of the linked list descriptor address.

Table 216. DMA 0–3 Linked List Descriptor Address Low Subregister (DMAxLAL)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	LAL							
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R	R
Address	If DMASA = 9h in the DMAxSA Register, accessible through the DMA 0–3 Subregister							

Bit	Description
[7:0] LAL	**Linked List Descriptor Address Low** LAH and LAL together form the a 12-bit address that points to the current linked list descriptor. As descriptors are aligned on 8-byte address boundaries, the lower three bits of LAL cannot be written, and are always zero. Writing to this DMAxLAL Register automatically starts the linked list DMA operation. A write to DMAxLAH is not required to start linked list operation. 00–F8: The lower 8 bits of the linked list descriptor address.

Chapter 19. Event System

The F6482 Series devices provide an eight-channel Event System that can route up to eight signals independent of any CPU or DMA activity. Any Event System source can be selected to drive a signal on an Event System channel.

These Event System sources are:

- Software
- Timers
- Multi-Channel Timer
- Real Time Clock (RTC)
- Comparators
- GPIO

Event System Destinations:

- Timers
- Multi-Channel Timer
- Real Time Clock (RTC)
- ADC
- DAC
- GPIO

The Event System is active in all operating modes, including Stop Mode.

19.1. Architecture

This chapter discusses the Event System, including Event System sources, channels, and destinations. A diagram of the Event System is shown in Figure 61.

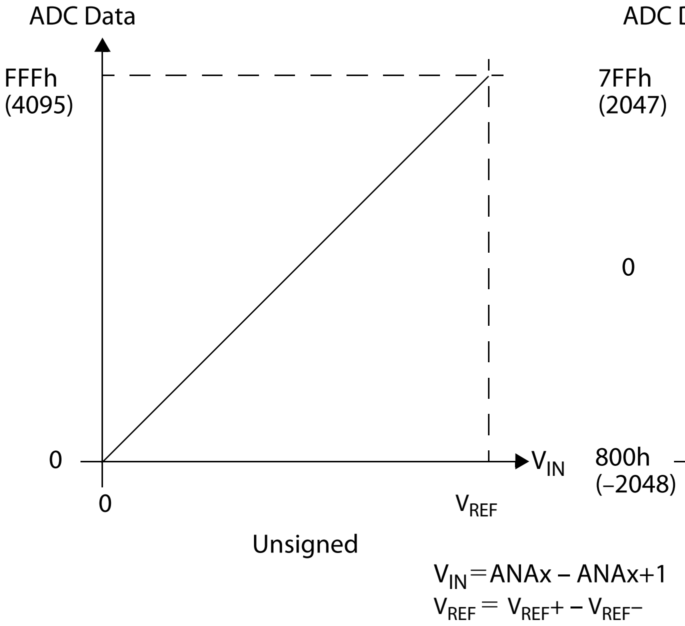

Figure 73. ADC Data (12-Bit) vs. Input Voltage for Unbalanced Differential Input Mode

21.2.3. Conversion Options

Five ADC conversion options are available, and are configured in the ADC Control 0 Register. These conversion options are independent from each other and can be selected in any combination. Furthermore, these conversion options can be enabled for any input mode. Each of these five options is described in the following subsections.

21.2.3.1. Single-Shot or Continuous Conversion

The ADC can be configured for single-shot or continuous conversion. When the CONTCONV bit is cleared, starting the ADC will produce a single result. When CONTCONV is set, starting the ADC will produce a continuous stream of results until CONTCONV is cleared. An interrupt can be generated for each conversion result. The data for each result can be moved by software or DMA.

21.2.3.2. Channel Scanning

The ADC can be configured to automatically scan multiple channels. If SCAN is cleared, channel scanning is not performed, and only the input (or input pair) defined in ANAINL is sampled for conversion.

If the SCAN bit is set, channel scanning is enabled, and the configuration of ANAINL and ANAINH determines which channels are scanned. If the bit corresponding to a particular channel is set, the channel will be included in the scan. Scanning commences with the LSB of ANAINL and completes with the MSB of ANAINH. ADC conversions are per-

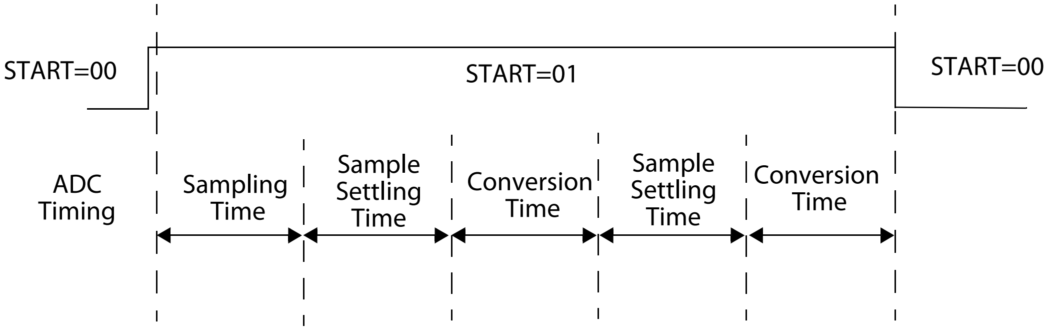

Figure 75. ADC Timing Diagram for 2-Pass 14-Bit Resolution with INMODE=10, 11

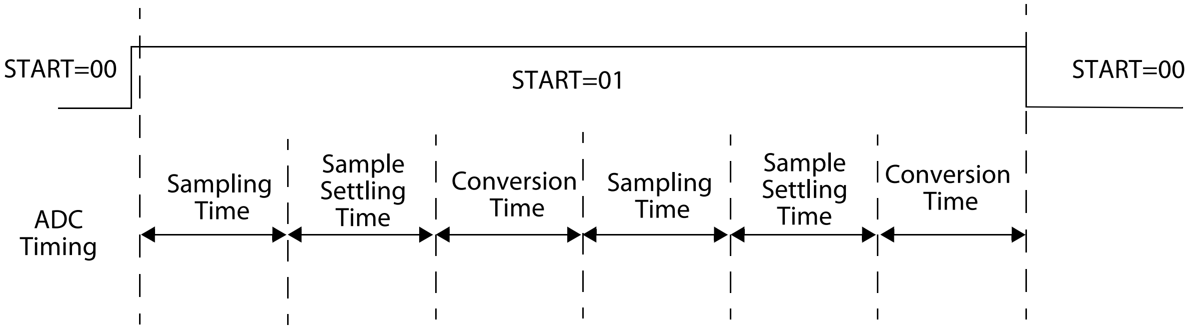

Figure 76. ADC Timing Diagram for 2-Pass 14-Bit Resolution with INMODE=01

21.2.6.3. ADC Wake-up, Sampling, and Settling

As the ADC is designed for low-power applications, the ADC core can be configured to auto-disable when no further conversions are scheduled by clearing ADC to 00 in the PWRCTL1 Register. When triggered to start a new conversion while the ADC is idle and ADC=00, the ADC wake-up period, T_{WAKE_ADC}, is automatically inserted prior to sampling (see the Electrical Characteristics chapter on page 598). If selected as VREF+, the internal voltage reference buffer will also automatically wake-up prior to sampling. When the conversion is completed and ADC = 00, the ADC's auto-disable feature will automatically disable the ADC when no further conversions are scheduled.

If performing multiple sequential conversions due to averaging, scanning or continuous conversion, it is recommended to turn off the ADC auto-disable function by setting ADC to 11 in the PWRCTL1 Register. When ADC is set to 11, the ADC is continuously enabled and T_{WAKE_ADC} is not incurred when the ADC is triggered to perform a conversion. After setting ADC=11, wait T_{WAKE_ADC} prior to triggering an ADC conversion.

available free for download from the Zilog website. While the Flash Controller programs Flash memory, the eZ8 CPU remains idle, but the system clock and on-chip peripherals continue to operate.

After an address is written, the page remains unlocked, allowing for subsequent writes to other addresses on the same page. To exit programming mode and lock Flash memory, write any value to the Flash Control Register except for the Mass Erase or Page Erase commands.

27.2.5. Byte Programming Mode

If the PMODE field is set to Byte Programming Mode, byte writes to program memory from user code program a byte into Flash.

27.2.6. Word Programming Mode

If the PMODE field is set to Word Programming Mode, Flash memory must be programmed one word (16 bits) at a time. Two byte writes to program memory from user code are required in the following sequence:

1. Byte write to the even address. This byte is registered until either Word Programming is completed or the register is overwritten by a new byte write to an even address.
2. Byte write to the odd address. Writing an odd address triggers Word Programming of the stored even address byte and the current byte to the current word address (LSB ignored).

If only the odd address is written, Byte Programming of the odd address byte is performed and the even address byte in program memory is unchanged.

Each Flash memory row of 128 bytes is subject to a maximum cumulative program time and, as specified in the Electrical Characteristics chapter on page 598, rows start at address multiples of `0080h`. Therefore, the first three rows start at addresses `0000h`, `0080h`, and `0100h`, respectively.

! **Caution:** The byte at each address of Flash memory cannot be programmed (any bits written to 0) more than twice before an erase cycle occurs.

27.2.7. Page Erase

Flash memory can be erased one page (512 bytes) at a time. Page erasing Flash memory sets all bytes in the active page to the value `FFh`. The Flash Page Select Register identifies the page to be erased. Only a page residing outside the protected block can be erased. With the Flash Controller unlocked, writing the value `95h` to the Flash Control Register initiates the Page Erase operation on the active page. While the Flash Controller executes the Page

Chapter 33. Electrical Characteristics

The data in this chapter has been tabulated prior to qualification (i.e., prequalification) and precharacterization of the F6482 Series product, and is subject to change. Additional electrical characteristics can be found in the individual chapters of this document.

33.1. Absolute Maximum Ratings

Stresses greater than those listed in Table 327 can cause permanent damage to the device. These ratings are stress ratings only. Operation of the device at any condition outside those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods can affect device reliability. For improved reliability, tie unused inputs to one of the supply voltages (V_{DD} or V_{SS}).

Table 327. Absolute Maximum Ratings*

Parameter	Min	Max	Units
Ambient temperature under bias	–40	+125	°C
Storage temperature	–65	+150	°C
Voltage on any pin with respect to V_{SS}	–0.3	+4.0	V
Maximum current on input and/or inactive output pin	–5	+5	µA
Maximum output current from active output pin	–25	+25	mA
28-pin Packages Maximum Ratings at –40°C to 85°C			
Total power dissipation		–	mW
Maximum current into V_{DD} or out of V_{SS}		–	mA
32-Pin QFN Maximum Ratings at –40°C to 85°C			
Total power dissipation		–	mW
Maximum current into V_{DD} or out of V_{SS}		–	mA
44-Pin LQFP Maximum Ratings at –40°C to 85°C			
Total power dissipation		–	mW
Maximum current into V_{DD} or out of V_{SS}		–	mA
64-pin LQFP Maximum Ratings at –40°C to 85°C			
Total power dissipation		–	mW
Maximum current into V_{DD} or out of V_{SS}		–	mA
Note: *Operating temperature is specified in the DC Characteristics section.			

Table 337. Analog-to-Digital Converter Electrical Characteristics and Timing (Continued)

		V_{DD}=1.8V to 3.6V T_A=–40°C to +85°C				
Symbol	**Parameter**	**Min**	**Typ***	**Max**	**Units**	**Conditions**
	Gain Error			±4	LSB	RESOLUT=0, No gain calibration run
				–	LSB	RESOLUT=0, Gain calibration run
				±16	LSB	RESOLUT=1, No gain calibration run
				–	LSB	RESOLUT=1, Gain calibration run
	Offset Error			±3	LSB	RESOLUT=0, No offset calibration run
				±1	LSB	RESOLUT=0, Offset calibration run
				±12	LSB	RESOLUT=1, No offset calibration run
				–	LSB	RESOLUT=1, Offset calibration run
I_{DD}ADCI	ADC Active Current with Internal Reference (REFSEL=1x)		180		µA	INMODE=0x, POWER=00
			230		µA	INMODE=1x, POWER=00
			95		µA	INMODE=0x, POWER=10
			115		µA	INMODE=1x, POWER=10
I_{DD}ADCE	ADC Active Current with External Reference or V_{DD} Reference (REFSEL=0x) including I_{DD}VEXT		155		µA	INMODE=0x, POWER=00
			205		µA	INMODE=1x, POWER=00
			85		µA	INMODE=0x, POWER=10
			105		µA	INMODE=1x, POWER=10

Notes:

1. Data in the Typical column is from characterization at 3.0V and 25°C. These values are provided for design guidance only and are not tested in production.
2. T_S is applied twice if INMODE=10 and RESOLUT=1.
3. T_{SS} is applied twice if RESOLUT=1.