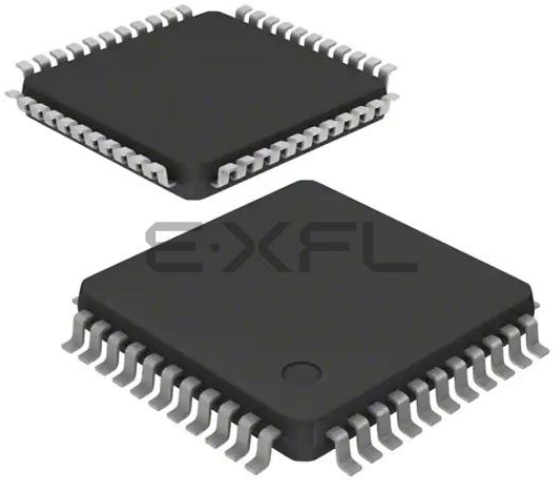

Welcome to E-XFL.COM

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, LVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	60KB (60K 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 10x12b; D/A 1x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	44-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f6081an024xk2246

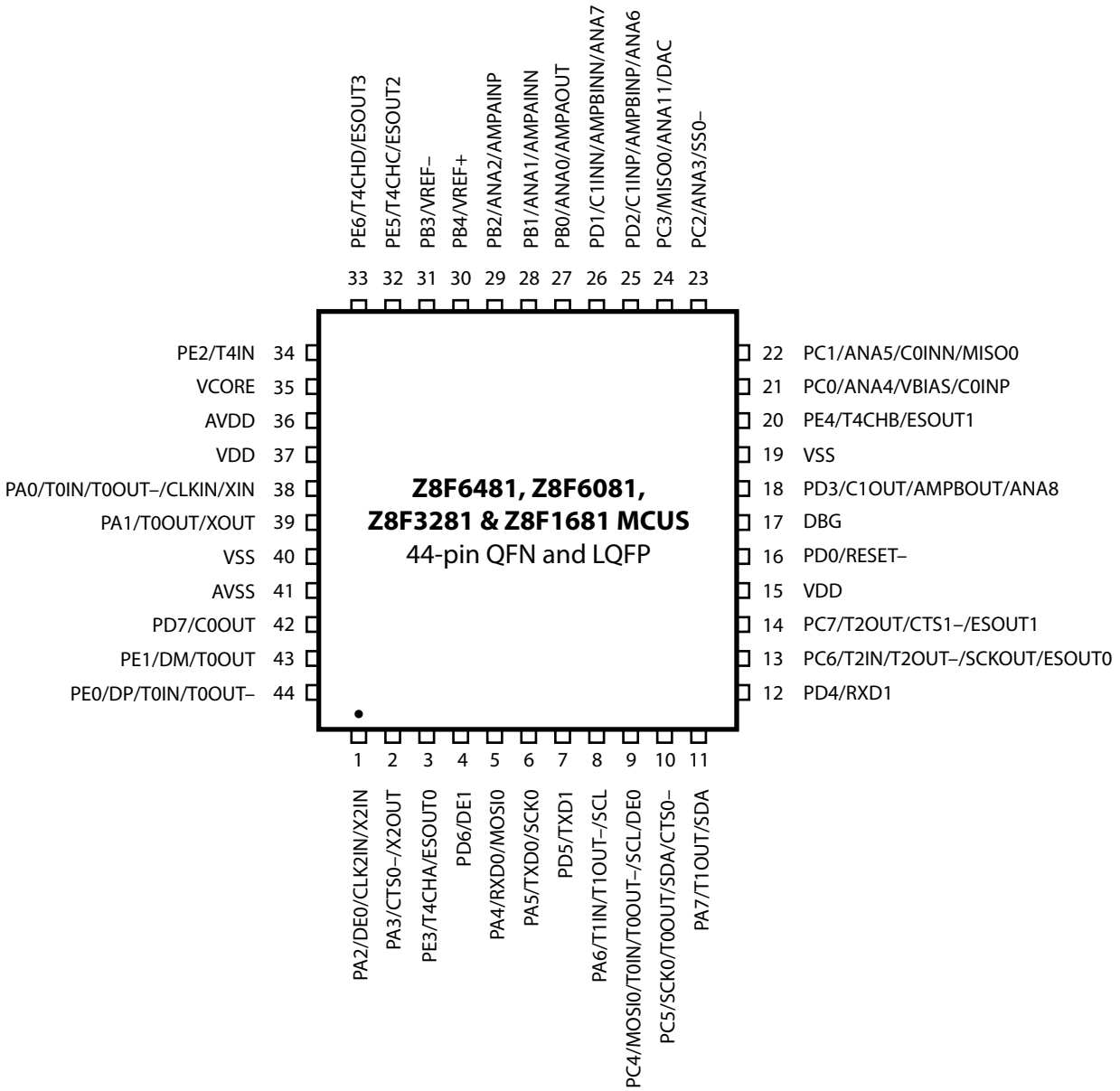

Figure 3. Z8F6481, Z8F6081, Z8F3281 & Z8F1681 MCUs, 44-Pin Quad Flat No Lead (QFN) and Low-Profile Quad Flat Package (LQFP)

Note: It is recommended to connect the QFN bottom pad to V_{SS}.

Table 9. Reset, Stop-Mode Recovery Characteristics and Latency

	Reset Characteristics and Latency		
Reset Type	**Control Registers**	**eZ8 CPU**	**Reset Latency (Delay)**
System Reset (non-POR/VBO Reset)	Reset (as applicable)	Reset	10 ms Reset Delay
System Reset (POR/VBO Reset)	Reset (as applicable)	Reset	10 ms Reset Delay
Stop-Mode Recovery (standard, FRECOV = 0)	Unaffected, except RSTSTAT, CLKCTL0, CLKCTL5, and IRQCTL registers	Reset	6 System Clock (DCO selected) cycles after Stop-Mode Recovery Delay
Stop-Mode Recovery (fast, FRECOV =1)	Unaffected, except RSTSTAT CLKCTL0, CLKCTL5, and IRQCTL registers	Reset	6 System Clock (DCO selected) cycles

5.2. System Reset

During a System Reset, the IPO, DCO and FLL are enabled. The FLL is configured for the default frequency of approximately 1 MHz with the IPO selected as Peripheral Clock (PCLK) to which the FLL locks the DCO. Upon the conclusion of a System Reset, the System Clock source and clock settings can be configured as desired. To learn more, see the Clock System chapter on page 96.

When System Reset occurs due to a VBO condition, the Reset Delay commences when the supply voltage first exceeds the VBO level (discussed later in this chapter). When System Reset occurs due to a POR condition, the Reset Delay commences from when the supply voltage first exceeds both the the POR and VBO levels. If the external $\overline{\text{RESET}}$ pin remains asserted at the end of the Reset period, the device remains in System Reset until the pin is deasserted.

At the beginning of System Reset, all GPIO pins are configured as inputs with pull-up resistor disabled, except PD0 that is shared with the Reset pin. On Reset, the Port D0 pin is configured as a bidirectional open-drain Reset. The pin is internally driven Low during port reset, after which the user code can reconfigure this pin as a general-purpose output.

During Reset, the eZ8 CPU and on-chip peripherals are idle; however, the Internal Precision Oscillator (IPO) continues to function.

On System 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 not initialized and undefined following System Reset. The eZ8 CPU fetches the Reset vector at Program Memory

To avoid missing interrupts, use the coding style in Example 2 to clear bits in the Interrupt Request 0 Register:

Example 2. Good coding style that avoids lost interrupt requests:

```
ANDX IRQ0, MASK
```

9.3.4. Software Interrupt Assertion

Program code can generate interrupts directly. Writing 1 to the correct bit in the Interrupt Request Register triggers an interrupt (assuming that interrupt is enabled). When the interrupt request is acknowledged by the eZ8 CPU, the bit in the Interrupt Request Register is automatically cleared to 0.

! Caution: Zilog recommends not using a coding style to generate software interrupts by setting bits in the Interrupt Request registers. All incoming interrupts received between execution of the first LDX command and the final LDX command are lost. See Example 3, which follows.

Example 3. Poor coding style that can result in lost interrupt requests:

```
LDX r0, IRQ0
OR r0, MASK
LDX IRQ0, r0
```

To avoid missing interrupts, use the coding style in Example 4 to set bits in the Interrupt Request registers.

Example 4. Good coding style that avoids lost-interrupt requests:

```
ORX IRQ0, MASK
```

9.4. Interrupt Control Register Definitions

For all interrupts other than the Watchdog Timer interrupt, the Primary Oscillator Fail Trap, and the Watchdog Oscillator Fail Trap, the Interrupt Control registers enable individual interrupts, set interrupt priorities, and indicate interrupt requests.

9.4.1. Interrupt Request 0 Register

The Interrupt Request 0 (IRQ0) Register, shown in Table 52, stores the interrupt requests for both vectored and polled interrupts. When a request is presented to the interrupt controller, the corresponding bit in the IRQ0 Register becomes 1. If interrupts are globally enabled (vectored interrupts), the interrupt controller passes an interrupt request to the eZ8

Table 63. IRQ2 Enable High Bit Register (IRQ2ENH)

Bit	7	6	5	4	3	2	1	0
Field	PA7VENH	PA6C0ENH	PA5C1ENH	PAD4ENH	PAD3ENH	PAD2ENH	PAD1ENH	PA0ENH
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	FC7h							

Bit	Description
[7] PA7VENH	**Port A7 or LVD Interrupt Request Enable High Bit**
[6] PA6C0ENH	**Port A6 or Comparator 0 Interrupt Request Enable High Bit**
[5] PA5C1ENH	**Port A5 or Comparator 1 Interrupt Request Enable High Bit**
[4:1] PAD*x*ENH	**Port A*x* or Port D*x* Interrupt Request Enable High Bit** *x* indicates the specific PAD bit (4–1). See Table 69 on page 146 for a selection of either Port A or Port D as the interrupt source.
[0] PA0ENH	**Port A0 Interrupt Request Enable High Bit** See Table 69 on page 146for a selection of either Port A or Port D as the interrupt source.

Table 64. IRQ2 Enable Low Bit Register (IRQ2ENL)

Bit	7	6	5	4	3	2	1	0
Field	PA7VENL	PA6C0ENL	PA5C1ENL	PAD4ENL	PAD3ENL	PAD2ENL	PAD1ENL	PA0ENL
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	FC8h							

Bit	Description
[7] PA7VENL	**Port A7 or LVD Interrupt Request Enable Low Bit**
[6] PA6C0ENL	**Port A6 or Comparator 0 Interrupt Request Enable Low Bit**
[5] PA5C1ENL	**Port A5 or Comparator 1 Interrupt Request Enable Low Bit**

14.1. UART-LDD Architecture

The UART-LDD consists of three primary functional blocks: transmitter, receiver and baud-rate generator. The UART-LDD's transmitter and receiver function independently but use the same baud rate and data format. The basic UART operation is enhanced by the Noise Filter. Figure 27 shows the UART-LDD architecture.

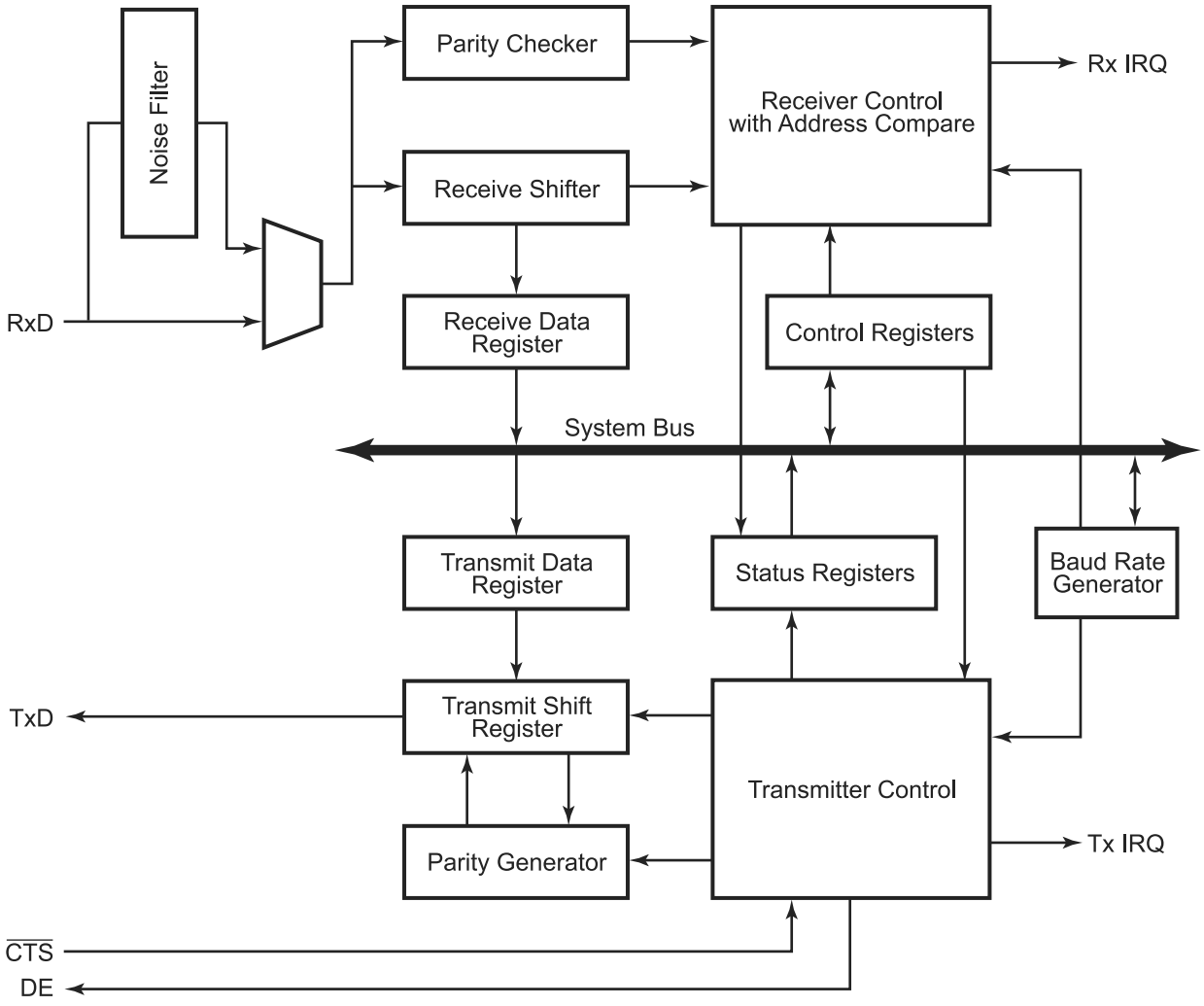

Figure 27. UART-LDD Block Diagram

14.1.14. UART-LDD and DMA Support

The UART-LDD will assert DMA RX request whenever receive data is available (RDA=1) and will deassert DMA RX request whenever the Receive Data Register is read by the DMA or software. When using DMA, it can be desirable to clear RDAIRQ so that interrupts occur on receive errors but not upon receive data.

The UART-LDD will assert DMA TX request whenever the Transmit Data Register is empty (TDRE=1) and will deassert DMA TX request whenever the Transmit Data Register is written by the DMA or software.

14.1.15. UART-LDD Baud Rate Generator

The UART-LDD Baud Rate Generator creates a lower frequency baud rate clock for data transmission. The input to the Baud Rate Generator is the system clock. The UART-LDD Baud Rate High and Low Byte registers combine to create a 16-bit baud rate divisor value (BRG[15:0]) that sets the data-transmission rate (baud rate) of the UART-LDD. The UART-LDD data rate for normal UART operation and DMX operation is calculated using the following equation:

$$\text{UART Data Rate (bits/s)} = \frac{\text{System Clock Frequency (Hz)}}{16 \times \text{UART Baud Rate Divisor Value}}$$

The UART-LDD data rate for LIN Mode UART operation is calculated using the following equation:

$$\text{UART Data Rate (bits/s)} = \frac{\text{System Clock Frequency (Hz)}}{\text{UART Baud Rate Divisor Value}}$$

The UART-LDD data rate for DALI Mode operation is calculated using the following equation:

$$\text{UART Data Rate (bits/s)} = \frac{\text{System Clock Frequency (Hz)}}{32 \times \text{UART Baud Rate Divisor Value}}$$

When the UART-LDD is disabled, the BRG functions as a basic 16-bit timer with interrupt on time-out. To configure the BRG as a timer with interrupt on time-out, follow the procedure below:

1. Disable the UART-LDD receiver by clearing the REN bit in the UART-LDD Control 0 Register to 0 (i.e., the TEN bit can be asserted; transmit activity can occur).
2. Load the appropriate 16-bit count value into the UART-LDD Baud Rate High and Low Byte registers.

Bit	Description (Continued)
[1] TXE	**Transmitter Empty** This bit indicates that the Transmit Shift Register is empty and character transmission is finished. 0: Data is currently transmitting. 1: Transmission is complete.
[0] CTS	**Clear to Send Signal** When this bit is read it returns the level of the $\overline{CTS}$ signal. If LBEN=1, the $\overline{CTS}$ input signal is replaced by the internal Receive Data Available signal to provide flow control in a loopback mode. CTS only affects transmission if the CTSE bit=1.

Table 126. UART-LDD 0–1 Status 0 Registers, DMX Mode (U*x*STAT0)

Bit	7	6	5	4	3	2	1	0
Field	RDA	Reserved	OE	Reserved	BRKD	TDRE	TXE	CTS
Reset	0	0	0	0	0	1	1	X
R/W	R	R	R	R	R	R	R	R
Address	U0STAT0 @ F41h, U1STAT0 @ F49h							

Note: R=read; X=undefined; x = 0,1.

Bit	Description
[7] RDA	**Receive Data Available** This bit indicates that the UART-LDD Receive Data Register has received data. Reading the UART-LDD Receive Data Register clears this bit. 0: The UART-LDD Receive Data Register is empty. 1: There is a byte in the UART-LDD Receive Data Register.
[6]	**Reserved** This bit is reserved and must be programmed to 0.
[5] OE	**Overrun Error** This bit indicates that an overrun error has occurred. An overrun occurs when new data is received and the Receive Data Register is not read. Reading the Receive Data Register clears this bit. 0: No overrun error occurred. 1: An overrun error occurred.
[4]	**Reserved** This bit is reserved and must be programmed to 0.
[3] BRKD	**Break Detect** This bit indicates that a break occurred. If the break condition exists for at least 22 bit times (all bits are zero) then this bit is set to 1. Reading the Receive Data Register clears this bit. 0: No break occurred. 1: A break occurred.

13.3.5. UART-LDD 0–1 Control 0 Registers

The UART-LDD 0–1 Control 0 registers, shown in Table 129, configure the basic properties of UART-LDD's transmit and receive operations. A more detailed discussion of each bit follows the table.

Table 129. UART-LDD 0–1 Control 0 Registers (U*x*CTL0)

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	U0CTL0 @ F42h, U1CTL0 @ F4Ah							

Note: R/W=read/write; *x* = 0,1.

Bit	Description
[7] TEN	**Transmit Enable** This bit enables or disables the transmitter. The enable is also controlled by the $\overline{CTS}$ signal and the CTSE bit. If the $\overline{CTS}$ 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 $\overline{CTS}$ signal has no effect on the transmitter. 1: The UART-LDD recognizes the $\overline{CTS}$ signal as an enable control for the transmitter.
[4] PEN	**Parity Enable** This bit enables or disables parity and should be cleared for DALI (MSEL=100) and DMX (MSEL=101) modes. 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.

Table 135. UART-LDD 0–1 Address Compare Registers (U*x*ADDR)

Bit	7	6	5	4	3	2	1	0
Field	COMP_ADDR							
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	U0ADDR @ F45h, U1ADDR @ F4Dh							

Note: R/W=read/write; *x* = 0,1.

Bit	Description
[7:0] COMP_ADDR	**Compare Address** This 8-bit value is compared to the incoming address bytes. For DMX slaves, a 9th bit, COMP_ADDR[8] in the DMX Control Register is also used.

13.3.12. UART-LDD 0–1 Baud Rate High and Low Byte Registers

The UART-LDD 0–1 Baud Rate High and Low Byte registers, shown in Tables 136 and 137, combine to create a 16-bit baud rate divisor value (BRG[15:0]) that sets the data transmission rate (baud rate) of the UART-LDD.

Table 136. UART-LDD 0-1 Baud Rate High Byte Registers (U*x*BRH)

Bit	7	6	5	4	3	2	1	0
Field	BRH							
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	U0BRH @ F46h, U1BRH @ F4Eh							

Note: R/W=read/write; *x* = 0,1.

Bit	Description
[7:0] BRH	**Baud Rate High** These bits set the High byte of the baud rate divisor value.

Chapter 16. I²C Master/Slave Controller

The I²C Master/Slave Controller ensures that the F6482 Series devices are bus-compatible with the I²C protocol. The I²C bus consists of the serial data signal (SDA) and a serial clock signal (SCL) bidirectional lines. The features of the I²C controller include:

- Operates in MASTER/SLAVE or SLAVE ONLY modes
- Supports arbitration in a multimaster environment (MASTER/SLAVE Mode)
- Supports data rates up to 400 Kbps
- 7-bit or 10-bit slave address recognition (interrupt-only on address match)
- Optional general call address recognition
- Optional digital filter on receive SDA, SCL lines
- Optional interactive receive mode allows software interpretation of each received address and/or data byte before acknowledging
- Unrestricted number of data bytes per transfer
- Baud Rate Generator can be used as a general-purpose timer with an interrupt if the I²C controller is disabled

16.1. Architecture

Figure 45 shows the architecture of the I²C controller.

ter to enable transmit interrupts. When the master initiates the data transfer, the I^2C controller holds SCL Low until the software has written the first data byte to the I2CDATA Register.

4. SCL is released and the first data byte is shifted out.
5. After the first bit of the first data byte has been transferred, the I^2C controller sets the TDRE bit, which asserts the transmit data interrupt.
6. The software responds to the transmit data interrupt (TDRE=1) by loading the next data byte into the I2CDATA Register, which clears TDRE.
7. After the data byte has been received by the master, the master transmits an Acknowledge instruction (or a Not Acknowledge instruction if this byte is the final data byte).
8. The bus cycles through Step 5 to Step 7 until the final byte has been transferred. If the software has not yet loaded the next data byte when the master brings SCL Low to transfer the most significant data bit, the slave I^2C controller holds SCL Low until the Data Register has been written. When a Not Acknowledge instruction is received by the slave, the I^2C controller sets the NCKI bit in the I2CISTAT Register, causing the Not Acknowledge interrupt to be generated.
9. The software responds to the Not Acknowledge interrupt by clearing the TXI bit in the I2CCTL Register, and by asserting the FLUSH bit of the I2CCTL Register to *empty* the Data Register.
10. When the master has completed the final acknowledge cycle, it asserts a stop or restart condition on the bus.
11. The slave I^2C controller asserts the stop/restart interrupt (i.e., sets the SPRS bit in the I2CISTAT Register).
12. The software responds to the stop/restart interrupt by reading the I2CISTAT Register, which clears the SPRS bit.

16.2.6.8. Slave Transmit Transaction With 10-Bit Address

The data transfer format for a master reading data from a slave with 10-bit addressing is shown in Figure 53. The following procedure describes the I^2C Master/Slave Controller operating as a slave in 10-bit addressing mode, transmitting data to the bus master.

S	Slave Address 1st Byte	W=0	A	Slave Address 2nd Byte	A	S	Slave Address 1st Byte	R=1	A	Data	A	Data	$\overline{A}$	P

Figure 53. Data Transfer Format, Slave Transmit Transaction with 10-Bit Address

1. The software configures the controller for operation as a slave in 10-bit addressing mode.

Table 161. I2CSTATE_H

State Encoding	State Name	State Description
0000	Idle	I^2C bus is idle or I^2C controller is disabled.
0001	Slave Start	I^2C controller has received a start condition.
0010	Slave Bystander	Address did not match; ignore remainder of transaction.
0011	Slave Wait	Waiting for stop or restart condition after sending a Not Acknowledge instruction.
0100	Master Stop2	Master completing stop condition (SCL=1, SDA=1).
0101	Master Start/Restart	MASTER Mode sending start condition (SCL=1, SDA=0).
0110	Master Stop1	Master initiating stop condition (SCL=1, SDA=0).
0111	Master Wait	Master received a Not Acknowledge instruction, waiting for software to assert stop or start control bits.
1000	Slave Transmit Data	Nine substates, one for each data bit and one for the Acknowledge.
1001	Slave Receive Data	Nine substates, one for each data bit and one for the Acknowledge.
1010	Slave Receive Addr1	Slave receiving first address byte (7- and 10-bit addressing) Nine substates, one for each address bit and one for the Acknowledge.
1011	Slave Receive Addr2	Slave receiving second address byte (10-bit addressing) nine substates, one for each address bit and one for the Acknowledge.
1100	Master Transmit Data	Nine substates, one for each data bit and one for the Acknowledge.
1101	Master Receive Data	Nine substates, one for each data bit and one for the Acknowledge.
1110	Master Transmit Addr1	Master sending first address byte (7- and 10-bit addressing) nine substates, one for each address bit and one for the Acknowledge.
1111	Master Transmit Addr2	Master sending second address byte (10-bit addressing) nine substates, one for each address bit and one for the Acknowledge.

Table 162. I2CSTATE_L

State I2CSTATE_H	Substate I2CSTATE_L	Substate Name	State Description
0000–0100	0000	–	There are no substates for these I2CSTATE_H values.
0110–0111	0000	–	There are no substates for these I2CSTATE_H values.
0101	0000	Master Start	Initiating a new transaction
	0001	Master Restart	Master is ending one transaction and starting a new one without letting the bus go idle.

19.6. Event System Register Definitions

Four register addresses provide access to the Event System subregisters that control source and destination selection for each Event System channel. Table 219 lists these Event System registers and subregisters.

Table 219. Event System Registers and Subregisters

Event System Source Selection Registers and Subregisters		
Register Mnemonic	**Address**	**Register Name**
ESSSA	F98h	Event System Source Subaddress Register
ESSSD	F99h	Event System Source Subdata Register
Subregister Mnemonic	**Subregister Address***	**Subregister Name**
ESCHxSRC	0–7h	Event System Channel 0–7 Source subregisters

Event System Destination Selection Registers and Subregisters		
Register Mnemonic	**Address**	**Register Name**
ESDSA	F9Ah	Event System Destination Subaddress Register
ESDSD	F9Bh	Event System Destination Subdata Register
Subregister Mnemonic	**Subregister Address***	**Subregister Name**
ESCHDSTx	0–3Fh	Event System Destination 0–3F Channel subregisters

Note: *The ESSSA Register contains the ESCHxSRC Subregister address; the ESDSA Register contains the ESCHDSTx Subregister address.

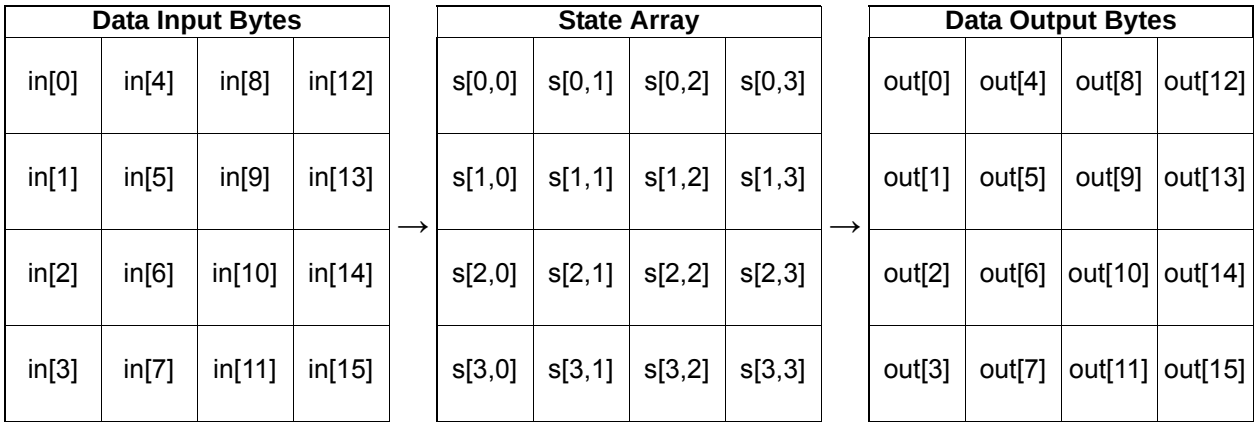

Figure 63. AES State Array Input and Output

Three status flags in the AESSTAT Register, KEYLD, IVLD and DATALD, indicate whether the AES accelerator has been properly setup with the 16-byte loads of the AESKEY, AESIV and AESDATA registers. The IVLD flag is only applicable when using the CBC or OFB confidentiality modes. Disabling the AES accelerator clears the load status flags, IVLD, KEYLD and DATALD. IVLD or KEYLD are also cleared during reloading of the respective registers. DATALD is cleared upon completion of an encryption/decryption operation in conjunction with clearing START/BUSY in the AESCTL Register.

The encryption/decryption operation can be initiated either manually or with auto-start. Setting the START/BUSY bit will manually start the requested encryption/decryption regardless of the load status flag settings. If the AUTODIS bit is cleared, encryption/decryption will auto-start when the 16th byte of data is loaded only if the KEYLD and IVLD (if applicable) are set. Table 226 describes conditions required to auto-start:

Table 226. Register Bit Settings for Auto-Start

AUTODIS	Mode	IVLD	KEYLD	DATALD	Auto-Start
0	01 (OFB) or 10 (CBC)	0	X	X	NO
0		0	0	X	NO
0		1	1	0	NO
0		1	1	1	YES
0	00 (ECB)	X	0	X	NO
0		X	1	0	NO
0		X	1	1	YES
1	X	X	X	X	NO

21.2.8. ADC Interrupts and DMA

The ADC can generate an interrupt request upon each new ADC result for any completed conversion or calibration (START= 01, 10, 11). The ADC can also generate an interrupt if the conversion result is outside the range defined by the window threshold registers (ADCUWINH, ADCUWINL, ADCLINH, ADCLWINL). Use the IRQ bit in the ADC Control 0 Register to select whether interrupts are generated due to only exceeding the window thresholds or due to both exceeding the window thresholds and end of convert. An interrupt request that is pending when the ADC is disabled or idle (ready) is not automatically cleared.

The ADC will assert a DMA request upon each new ADC result, and will deassert a DMA request whenever the ADCD_L Register (DMACTL=0) or ADCD_H Register (DMACTL=1) is read by the DMA or software. When DMACTL=0, the typical goal is to transfer both data bytes, and the DMA is configured to have fixed word address control for the DMA source address; the source address is configured to be the ADCD_H Register address. When DMACTL=1, the typical goal is to transfer only the most significant data byte, and the DMA is configured to have fixed address control for the DMA source address; the source address is configured to be the ADCD_H Register address.

If starting a new ADC conversion and DMA transfer sequence, reading the ADC_L Register (DMACTL=0) or the ADC_H Register (DMACTL=1) prior to enabling DMA and starting conversion ensures that any residual ADC DMA request from prior ADC activity is deasserted. A DMA request is not asserted upon the completion of an offset or gain calibration.

21.2.9. Calibration and Compensation

Both gain and offset calibration can be performed in situ to achieve even higher accuracy than specified in the Electrical Characteristics chapter on page 598. These calibration operations are performed using the current ADC configuration, as defined by the INMODE, PRESCALE, ST, and SST bits. After these parameters are reconfigured, initiating calibration prior to performing conversions can optimize results. Only initiate calibration when continuous conversion is not selected (i.e., CONTCONV = 0).

Offset calibration is performed when the START bits are written to 10. Prior to initiating offset calibration, 14-bit resolution must be selected by setting RESOLUT = 1. The offset result can be used for both 12-bit and 14-bit resolution conversions. The calibration is complete when START is cleared to 00. The offset calibration result is stored in the OFFSET field in the ADCOFF Register as a two’s-complement value, and is automatically applied to compensate subsequent conversions by hardware. OFFSET can be read by software and can be stored and rewritten any time to the ADCOFF Register to allow consistent usage of offset calibrations. For example, when using multiple input modes, each mode can exhibit a unique offset calibration value.

27.2.1. Flash Operation Timing

Before performing either a program or erase operation on Flash memory, the Digitally Controlled Oscillator (DCO) must be running and must be locked using the Frequency Locked Loop (FLL) to a minimum frequency of 1 MHz.

27.2.2. Flash Code Protection Against External Access

The user code contained within Flash memory can be protected against external access with the On-Chip Debugger. Programming the FRP Flash option bit prevents the reading of user code with the On-Chip Debugger. To learn more, see the Flash Option Bit Address Space section on page 544 and the On-Chip Debugger section on page 558.

27.2.3. Flash Code Protection Against Accidental Program and Erasure

The F6482 Series provides several levels of protection against accidental program and erasure of the Flash memory contents. This protection is provided by a combination of the Flash option bits, the register locking mechanism, the page select redundancy, and the block level protection control of the Flash Controller.

27.2.3.1. Flash Code Protection Using the Flash Option Bits

The FWP Flash option bit provides Flash Program Memory protection as listed in Table 277. To learn more, see the Flash Option Bit Address Space section on page 544.

Table 277. Flash Code Protection Using the Flash Option Bit

FWP	Flash Code Protection Description
0	Programming and erasing disabled for all of Flash Program Memory. In user code programming, Page Erase, and Mass Erase are all disabled. Mass Erase is available through the On-Chip Debugger.
1	Programming and Page Erase are enabled for all of Flash Program Memory. Mass Erase is available through the On-Chip Debugger.

27.2.3.2. Flash Code Protection Using the Flash Controller

At Reset, the Flash Controller locks to prevent accidental program or erasure of Flash memory. Observe the following procedure to unlock the Flash Controller from user code:

1. Write the Page Select Register with the target page.
2. Write the first unlock command, 73h, to the Flash Control Register.
3. Write the second unlock command, 8Ch, to the Flash Control Register.
4. Rewrite the Page Select Register with the target page previously stored in this register in Step 1.

Bit	Description (Continued)
[6] WDT_AO	**Watchdog Timer Always ON** 0: Watchdog Timer is automatically enabled upon application of system power. Watchdog Timer cannot be disabled. 1: Watchdog Timer is enabled upon execution of the WDT instruction. After it is enabled, the Watchdog Timer can only be disabled by a Reset or Stop-Mode Recovery. This setting is the default for unprogrammed (erased) Flash.
[5:4]	**Reserved** These bits are reserved
[3:2] VBOCTL	**Voltage Brown-Out Protection Control** 00: Reserved (defaults to disabled). 01: Voltage Brown-Out Protection is disabled. 10: Voltage Brown-Out Protection is enabled in Normal and Halt modes, but is disabled in Stop Mode. 11: Voltage Brown-Out Protection is always enabled. This setting is the default for unprogrammed (erased) Flash.
[1] FRP	**Flash Read Protect** 0: User program code is inaccessible. Limited control features are available through the On-Chip Debugger. 1: User program code is accessible. All On-Chip Debugger commands are enabled. This setting is default for unprogrammed (erased) Flash.
[0] FWP	**Flash Write Protect** This option bit provides Flash Program Memory protection: 0: Programming is disabled. Page Erase through user software is disabled. Mass Erase is available using the On-Chip Debugger. 1: Programming, Page Erase and Mass Erase are enabled for all of Flash Program Memory.

Table 287. Flash Option Bits at Program Memory Address 0001h

Bit	7	6	5	4	3	2	1	0
Field	Reserved							
Reset	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Program Memory 0001h							

Note: X=undefined; R/W=read/write.

Bit	Description
[7:0]	**Reserved** These bits are reserved.

```
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 downcounter, the returned value is inverted when this register is read; therefore, the returned result appears to be an upcounter. If the device is not in Debug Mode, this command returns FFFFh.

```
DBG ← 03h
DBG → ~OCDCNTR[15:8]
DBG → ~OCDCNTR[7:0]
```

Write OCD Control Register (04h). The Write OCD Control Register command writes the data that follows to the OCDCTL Register.

```
DBG ← 04h
DBG ← OCDCTL[7:0]
```

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

```
DBG ← 05h
DBG → OCDCTL[7:0]
```

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

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

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

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

Write Register (08h). The Write Register command writes data to the Register File. Data can be written 1–256 bytes at a time (256 bytes can be written by setting size to 0). If the device is not in Debug Mode, the address and data values are discarded. If the Read Protect option bit is enabled, then only writes to the on-chip peripheral registers are allowed and all other register write data values are discarded.

```
DBG ← 08h
DBG ← {0h,Register Address[11:8]}
DBG ← Register Address[7:0]
```

33.4.6. Non-Volatile Data Storage

Table 336 presents electrical and timing data for the F6482 Series' Non-Volatile Data Storage function.

Table 336. Non-Volatile Data Storage Electrical Characteristics and Timing

	V_{DD}=1.8V to 3.6V T_A=–40°C to +85°C				
Parameter	**Min**	**Typ**	**Max**	**Units**	**Conditions**
NVDS Byte Read Time	34		519	µs	With system clock at 20MHz
NVDS Byte Program Time	0.171		39.7	ms	With system clock at 20MHz
Data Retention	100		–	years	25°C
Endurance	100,000			cycles	Cumulative write cycles for entire memory

33.4.7. Analog-to-Digital Converter

Table 337 presents electrical and timing data for the F6482 Series' Analog-to-Digital Converter function.

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

		V_{DD}=1.8V to 3.6V T_A=–40°C to +85°C				
Symbol	**Parameter**	**Min**	**Typ***	**Max**	**Units**	**Conditions**
N	Resolution			12	Bit	RESOLUT=0 (12-bit)
				14	Bit	RESOLUT=1 (2-pass 14-bit)
INL	Integral Nonlinearity			±2	LSB	RESOLUT=0
				±5	LSB	RESOLUT=1, INMODE=01
DNL	Differential Nonlinearity			≤±1	LSB	No missing codes RESOLUT=0
				≤±2	LSB	RESOLUT=1, INMODE=01

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.