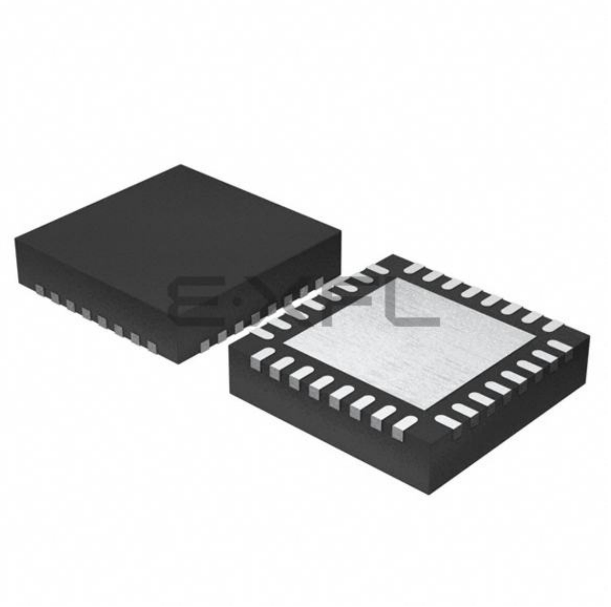
Zilog - Z8F1681QK024XK2247 Datasheet



What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	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	26
Program Memory Size	16KB (16K 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 9x12b; D/A 1x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-VFQFN Exposed Pad
Supplier Device Package	32-QFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f1681qk024xk2247

c. Write FLLNDIVH with the most significant byte of the desired frequency divisor. Writing to FLLNDIVH will trigger the fast locking algorithm.

- If stored values for the DCO control words are available, They can be used to seed FLL convergence using a linear locking algorithm. To initiate the linear locking algorithm, observe the following procedure:
  a. Set SEEDSEL = 1 to indicate that loaded values for the DCO control words should be used during the locking process. Also set FLLEN=1.
  b. Write the value stored from DCOCTLCL to DCOCTLL.
  c. Write the value stored from DCOCTLCH to DCOCTLH.
  d. Write FLLNDIVL with the least significant byte of the desired frequency divisor.
  e. Write FLLNDIVH with the most significant byte of the desired frequency divisor. Writing to FLLNDIVH will trigger the linear locking algorithm.

Three bits, FLLRDY, FLLLL, and FLADONE, are provided in the CLKCTL5 Register to describe the FLL status. When the FLL has achieved lock status, FLLRDY is set. If the FLL loses lock status, FLLLL is set, and FLLRDY is cleared. If the FLL is disabled, both FLLRDY and FLLLL are cleared. If the lock is lost while the FLL is enabled, a system clock fail trap occurs if the FLLIRQE bit in the CLKCTL5 Register is set. The FLADONE bit is set when the fast locking algorithm has completed. The setting of the FLLRDY bit precedes the setting of FLADONE.

Note: The fast locking algorithm should not be interrupted by entering Stop Mode, clearing the FLLEN bit, or by changing the FLL divider value. After the fast locking algorithm has been initiated, always check to determine that it has completed (as evidenced by FLADONE = 1) before entering Stop Mode, clearing the FLLEN bit, or by changing the FLL divider value. The fast locking algorithm is always initiated automatically during System Reset.

While the FLL is enabled, it continues to converge the DCO control codes, as required, to maintain frequency lock. Although the DCO has fine resolution, the resolution is finite and the FLL operation can result in dithering between two values of the DCO control words. If such dither is undesirable, the FLL can be disabled for dither-free operation and enabled periodically to again converge the DCO control words and therefore account for environmental changes. Upon being enabled, the FLL will converge the DCO control words based on the divisor values in FLLNDIVL and FLLNDIVH, achieve lock using the linear algorithm, and set FLLRDY.

Immediately after Stop-Mode Recovery, the DCO is enabled and operates with the DCO control words existing upon entry into Stop Mode. In addition, the FLL is disabled and FLLRDY is cleared. If no frequency change is desired, the FLL can be enabled and will lock using the linear algorithm. When locked, FLLRDY will be set.

14.1.1. Data Format for Standard UART Modes

The UART-LDD always transmits and receives data in an 8-bit data format with the least significant bit first. An even-or-odd parity bit or multiprocessor address/data bit can be optionally added to the data stream. Each character begins with an active Low start bit and ends with either one or two active High stop bits. Figures 28 and 29 show the asynchronous data format employed by the UART-LDD without parity and with parity, respectively.

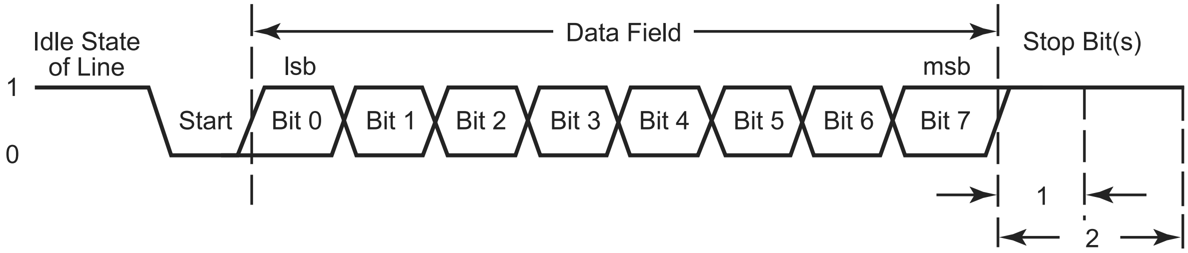


Figure 28. UART-LDD Asynchronous Data Format without Parity

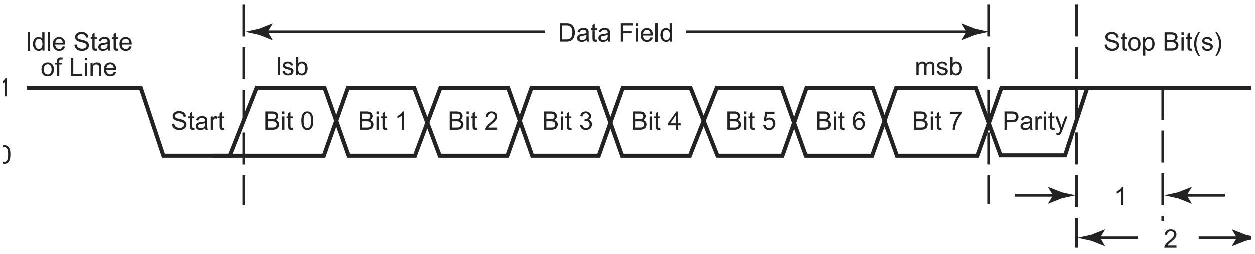


Figure 29. UART-LDD Asynchronous Data Format with Parity

14.1.2. Transmitting Data using the Polled Method

Observe the following steps to transmit data using the polled-operating method:

1. Write to the UART-LDD Baud Rate High and Low Byte registers to set the appropriate baud rate.
2. Enable the UART-LDD pin functions by configuring the associated GPIO port pins for alternate-function operation.
3. If Multiprocessor Mode is appropriate, write to the UART-LDD Control 1 Register to enable Multiprocessor (9-bit) Mode functions.
4. Set the Multiprocessor Mode Select (MPEN) bit to enable Multiprocessor Mode.

Table 150 defines the valid ESPI states.

Table 150. ESPISTATE Values

ESPISTATE Value	Description
00_0000	Idle.
00_0001	Slave Wait for SCK.
00_0010	I^{2}S Slave Mode start delay.
00_0011	I^{2}S Slave Mode start delay.
01_0000	SPI Master Mode start delay.
11_0001	I^{2}S Master Mode start delay.
11_0010	I^{2}S Master Mode start delay.
10_1110	Bit 7 Receive.
10_1111	Bit 7 Transmit.
10_1100	Bit 6 Receive.
10_1101	Bit 6 Transmit.
10_1010	Bit 5 Receive.
10_1011	Bit 5 Transmit.
10_1000	Bit 4 Receive.
10_1001	Bit 4 Transmit.
10_0110	Bit 3 Receive.
10_0111	Bit 3 Transmit.
10_0100	Bit 2 Receive.
10_0101	Bit 2 Transmit.
10_0010	Bit 1 Receive.
10_0011	Bit 1 Transmit.
10_0000	Bit 0 Receive.
10_0001	Bit 0 Transmit.

15.4.7. ESPI 0-1 Baud Rate High and Low Byte Registers

The ESPI 0-1 Baud Rate High and Low Byte registers, shown in Tables 151 and 152, combine to form a 16-bit reload value, BRG[15:0], for the ESPI Baud Rate Generator. The ESPI baud rate is calculated using the following equation:

$$\text{SPI Baud Rate (bits/s)} = \frac{\text{System Clock Frequency (Hz)}}{2 \times \text{BRG[15:0]}}$$

Slave 10-Bit Address Recognition Mode. If IRM=0 during the address phase and the controller is configured for MASTER/SLAVE or SLAVE 10-BIT ADDRESS Mode, the hardware detects a match to the 10-bit slave address defined in the I2CMODE and I2CSLVAD registers and generates the slave address match interrupt (the SAM bit=1 in the I2CISTAT Register). The I^2C controller automatically responds during the Acknowledge phase with the value in the NAK bit of the I2CCTL Register.

16.2.6.2. General Call and Start Byte Address Recognition

If GCE=1 and IRM=0 during the address phase and the controller is configured for master/slave or slave in either 7- or 10-bit address modes, the hardware detects a match to the General Call Address or the start byte and generates the slave address match interrupt. A General Call Address is a 7-bit address of all zeroes, with the $R/\overline{W}$ bit=0. A start byte is a 7-bit address of all zeroes, with the $R/\overline{W}$ bit=1. The SAM and GCA bits are set in the I2CISTAT Register. The RD bit in the I2CISTAT Register distinguishes a General Call Address from a start byte which is cleared to 0 for a General Call Address). For a General Call Address, the I^2C controller automatically responds during the address acknowledge phase with the value in the NAK bit of the I2CCTL Register. If the software is set to process the data bytes associated with the GCA bit, the IRM bit can optionally be set following the SAM interrupt to allow the software to examine each received data byte before deciding to set or clear the NAK bit. A start byte will not be acknowledged – a requirement of the I^2C specification.

16.2.6.3. Software Address Recognition

To disable hardware address recognition, the IRM bit must be set to 1 prior to the reception of the address byte(s). When IRM=1, each received byte generates a receive interrupt (RDRF=1 in the I2CISTAT Register). The software must examine each byte and determine whether to set or clear the NAK bit. The slave holds SCL Low during the acknowledge phase until the software responds by writing to the I2CCTL Register. The value written to the NAK bit is used by the controller to drive the I^2C bus, then releasing the SCL. The SAM and GCA bits are not set when IRM=1 during the address phase, but the RD bit is updated based on the first address byte.

16.2.6.4. Slave Transaction Diagrams

In the following transaction diagrams, the shaded regions indicate data transferred from the master to the slave and the unshaded regions indicate the data transferred from the slave to the master. The transaction field labels are defined as follows:

S	Start
W	Write
A	Acknowledge
A	Not Acknowledge
P	Stop

Bit	Description (Continued)
[6] START	**Send Start Condition** When set, this bit causes the I^{2}C controller (when configured as the master) to send a start condition. After it is asserted, this bit is cleared by the I^{2}C controller after it sends the start condition or by deasserting the IEN bit. If this bit is 1, it cannot be cleared by writing to the bit. After this bit is set, a start condition is sent if there is data in the I2CDATA or I^{2}C Shift Register. If there is no data in one of these registers, the I^{2}C controller waits until data is loaded. If this bit is set while the I^{2}C controller is shifting out data, it generates a restart condition after the byte shifts and the Acknowledge phase completes. If the stop bit is also set, it waits until the stop condition is sent before the start condition. If start is set while a SLAVE Mode transaction is underway to this device, the start bit will be cleared and ARBLST bit in the Interrupt Status Register will be set.
[5] STOP	**Send Stop Condition** When set, this bit causes the I^{2}C controller (when configured as the master) to send the stop condition after the byte in the I^{2}C Shift Register has completed transmission or after a byte is received in a receive operation. When set, this bit is reset by the I^{2}C controller after a stop condition has been sent or by deasserting the IEN bit. If this bit is 1, it cannot be cleared to 0 by writing to the register. If a stop is set while a SLAVE Mode transaction is underway, the stop bit is cleared by hardware.
[4] BIRQS	**Baud Rate Generator Interrupt Request Select** This bit is ignored when the I^{2}C controller is enabled. If this bit is set=1 when the I^{2}C controller is disabled (IEN=0), the baud rate generator is used as an additional timer causing an interrupt to occur every time the baud rate generator counts down to one. The baud rate generator runs continuously in this mode, generating periodic interrupts.
[3] TXI	**Enable TDRE Interrupts** This bit enables interrupts when the I^{2}C Data Register is empty.
[2] NAK	**Send NAK** Setting this bit sends a Not Acknowledge condition after the next byte of data has been received. It is automatically deasserted after the Not Acknowledge is sent or the IEN bit is cleared. If this bit is 1, it cannot be cleared to 0 by writing to the register.
[1] FLUSH	**Flush Data** Setting this bit clears the I^{2}C Data Register and sets the TDRE bit to 1. This bit allows flushing of the I^{2}C Data Register when an NAK condition is received after the next data byte is written to the I^{2}C Data Register. Reading this bit always returns 0.
[0] FILTEN	**I^{2}C Signal Filter Enable** Setting this bit enables low-pass digital filters on the SDA and SCL input signals. This function provides the spike suppression filter required in I^{2}C Fast Mode. These filters reject any input pulse with periods less than a full system clock cycle. The filters introduce a 3-system clock cycle latency on the inputs.

16.3.4. I^{2}C Baud Rate High and Low Byte Registers

The I^{2}C Baud Rate High and Low Byte registers, shown in Tables 157 and 158, combine to form a 16-bit reload value, BRG[15:0], for the I^{2}C Baud Rate Generator.

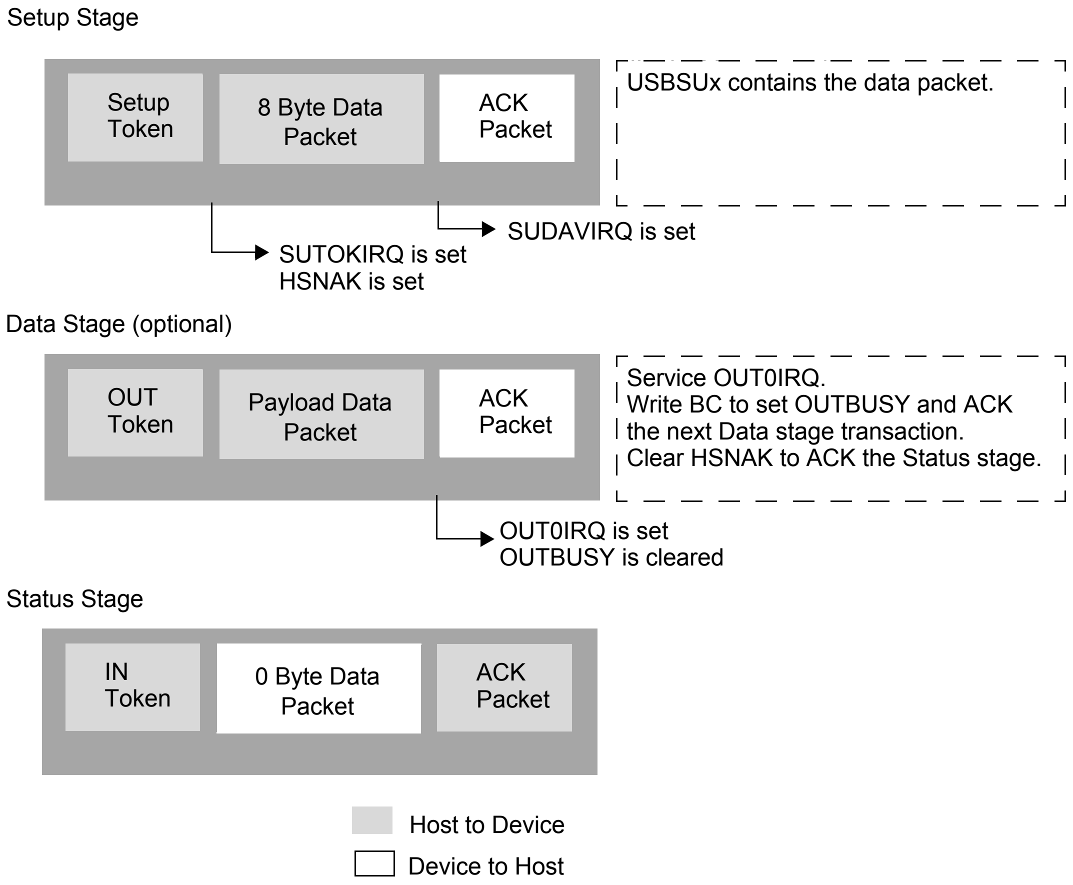


Figure 56. Control Write Transfer Example

17.2.4.2. Control Read

Control read transfer is similar to control write transfer. The difference is in the Data stage. During the Data stage of ca control read transfer, after each acknowledge by the host, the USB Module sets the IN endpoint 0 interrupt request bit, IN0IRQ, in the USB IN Interrupt Request Subregister (USBINIRQ) and generates a USB interrupt if INxIEN is set in the USB IN Interrupt Enable Subregister (USBINIEN). Software should load new data into the IN endpoint 0 buffer memory and then reload the IN0BC Subregister with the number of data bytes loaded. Reloading the IN0BC Subregister causes the INBUSY bit to be set in the USBEP0CS Subregister and arms the endpoint for the next IN transaction. For the first data transaction after the setup transaction, software should arm IN endpoint 0 buffer memory based upon the Setup stage transaction. The Status stage of a control transfer is the final operation in the sequence. Software should clear the HSNAK bit (by writing a 1

17.3.21. USB OUT 0–3 Byte Count Subregisters

The USB OUT 0–3 Byte Count subregisters, shown in Table 190, contain the USB OUT endpoint byte counts.

Table 190. USB OUT 0–3 Byte Count Subregisters (USBOxBC)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved	BC						
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Address	If USBSA = 45h, 47h, 49h, 4Bh in the USB Subaddress Register, accessible through the USB Subdata Register							

Bit	Description
[7]	**Reserved** This bit is reserved and must be programmed to 0.
[6:0] BC	**OUT Byte Count** 00–40: BC contains the number of bytes sent during the last OUT transfer from the host to the OUT endpoint x. 41-7F: Reserved.

Table 210. DMA Destination Address Low Subregister (DMAxDSTL)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	DSTL							
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 DMASA = 3h in the DMAxSA Register, accessible through the DMA 0–3 Subregister							

Bit	**Description**
[7:0] DSTL	**Destination Address Low** 00–FF: Lower 8 bits of the DMA destination address.

19.6.6. Event System Destination 0–3F Channel Subregisters

The Event System Destination 0–3F Channel (ESDSTxCH) subregisters, shown in Table 225, determine whether each destination is connected to the Event System and select the Event System channel to connect to the destination.

Table 225. Event System Destination 0–3F Channel Subregisters (ESDSTxCH)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved				DSTCON	DSTCHSEL		
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 00h -3Fh in Event System Destination Subaddress Register, accessible through the Event System Destination Subdata Register							

Bit	Description
[7:4]	**Reserved** This bit is reserved and must be programmed to 0.
[3] DSTCON	**Event System Destination Connection** 0: The selected Event System channel is not connected to the addressed destination. The destination is connected to logic 0. 1: The selected Event System channel is connected to the addressed destination.
[2:0] DSTCHSEL	**Event System Destination Channel Selection** 000: Channel 0 is connected to the destination if DSTCON=1. 001: Channel 1 is connected to the destination if DSTCON=1. 010: Channel 2 is connected to the destination if DSTCON=1. 011: Channel 3 is connected to the destination if DSTCON=1. 100: Channel 4 is connected to the destination if DSTCON=1. 101: Channel 5 is connected to the destination if DSTCON=1. 110: Channel 6 is connected to the destination if DSTCON=1. 111: Channel 7 is connected to the destination if DSTCON=1.

21.3.7. ADC Data High Register

The ADC Data High Register, shown in Table 241, contains the MSBs of the ADC result. Access to the ADC Data High Register is read-only. Reading the ADC Data High Register latches data in the ADC Low Register.

Table 241. ADC Data High Register (ADCD_H)

Bits	7	6	5	4	3	2	1	0
Field	ADCDH							
Reset	00h							
R/W	R							
Address	F76h							

Bit	Description
ADCDH	**ADC Data High**
[7:6]	Reserved: these bits must be programmed to 00.
[5:0]	00–3F: The 6 MSBs of the last conversion result are held in the 6 LSBs of this data register until the next ADC conversion has completed.

21.3.8. ADC Data Low Register

The ADC Data Low Register, shown in Table 242, contains the LSBs of the ADC result. Access to the ADC Data Low Register is read-only. Reading the ADC Data High Register latches data in the ADC Low Register.

Table 242. ADC Data Low Register (ADCD_L)

Bits	7	6	5	4	3	2	1	0
Field	ADCDL							
Reset	00h							
R/W	R							
Address	F77h							

Bit	Description
ADCDL	**ADC Data Low** ADC Data Low is a function of RESOLUT.
RESOLUT = 0 (12-bit)	
[7:2]	00–3F: The 6 LSBs of the last conversion result are latched into this data register whenever the ADC Data High Byte register is read.
[1:0]	0–1: Reserved.

The negative Op Amp A inputs are selected with the INNSEL bit in the AMPACTL1 Register, and include:

- GPIO pin used as Op Amp A negative input, AMPAINN
- Op Amp A output through internal feedback network using an internal connection with gain, defined by the GAIN bit
- Op Amp A output, a unity gain configuration using an internal connection

The Op Amp A output, AMPAOUT, can be selected as an internal input to OPAMP B, Comparator 0, Comparator 1, and the ADC. Additionally, it can be connected to the GPIO used as AMPAOUT by setting the OUTCTL bit in the AMPACTL0 Register, and configuring the appropriate alternate function, as described in the General-Purpose Input/Output chapter on page 55. This GPIO can also be selected as an input to the ADC.

23.2.2. Op Amp B

As shown in Figure 81 on page 474, four positive and three negative inputs are available. The positive Op Amp B inputs are selected with the INPSEL bit in the AMPBCTL0 Register, and include:

- 1.0V from the Reference System; see the Comparators and Reference System chapter on page 484 to learn more
- A GPIO pin used as the Op Amp B positive input, AMPBINP
- Op Amp A output. This selection provides an internal connection that does not involve the GPIO used as the Op Amp A output, AMPAOUT
- Internal Programmable Reference 1, with level selected by the PREFLVL bit and source selected by the PREFSRC bit in the CMP1CTL1 Register; see Table 261 on page 496 to learn more

The negative Op Amp B inputs are selected with the INNSEL and MODE bits in the AMPBCTL1 Register, and include:

- A GPIO pin used as the Op Amp B negative input, AMPBINN
- An internal connection from the current drive network
- Op Amp B output, a unity gain configuration using an internal connection

The Op Amp B output, AMPBOUT, can be selected as an internal input to Op Amp A and the ADC. Additionally, it can be connected to the GPIO used as AMPBOUT by selecting OUTCTL=11 in the AMPBCTL0 Register and configuring the appropriate alternate function, as described in the General-Purpose Input/Output chapter on page 55. This GPIO can also be selected as an input to Comparator 0, Comparator 1, and the ADC.

Figure 84 shows a simplified block diagram of the Reference System, which includes the Reference Generator and two programmable references.

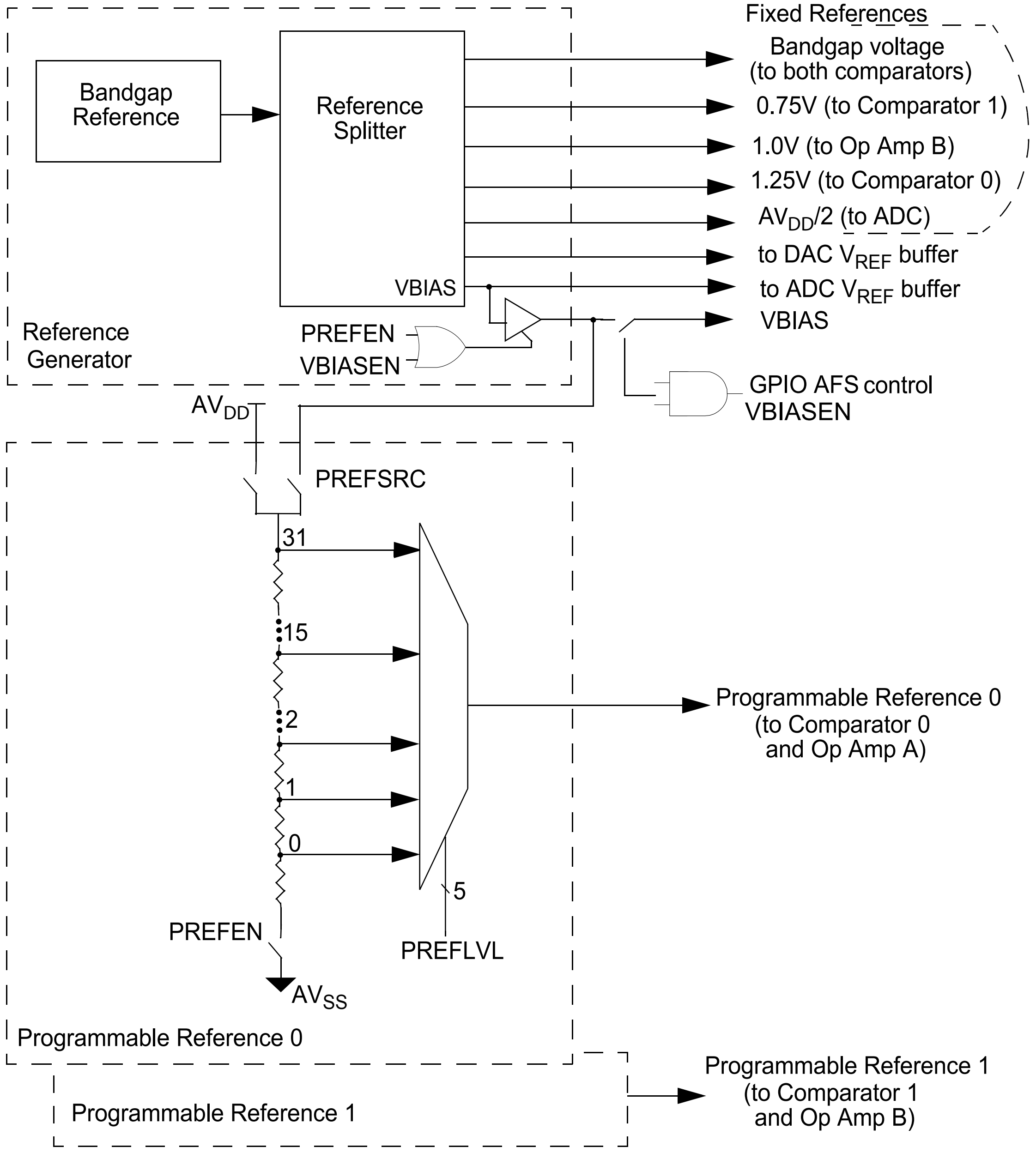


Figure 84. Reference System Block Diagram

24.2. Comparator Operation

Two identical general-purpose CMOS analog comparators each provide rail-to-rail operation with four speed-vs.-power settings and three hysteresis options. These comparators are enabled by setting the COMP0 and COMP1 bits in the PWRCTL0 Register, which is described in the Low-Power Modes chapter on page 50. The power setting is determined by the CPOWER bit, which selects current consumption ranging from 27 µA, with a propagation delay of 150 ns, to 0.2 µA, with a propagation delay of 10 µs. The low power settings can allow for continuous comparator usage in low-power systems. Hysteresis is selected by the HYST bit; selections range from no hysteresis to 40 mV.

A 4-to-1 input multiplexer exists on each comparator positive input and each comparator negative input. The positive input is selected using the INPSEL bit to be either the temperature sensor, GPIO or one of the op amp outputs, AMPAOUT or AMPBOUT. The negative input is selected using the INNSEL and PREFEN bits to be either a GPIO, a fixed reference (0.75V/1.25 V), the bandgap voltage, a programmable internal reference or the DAC output. Multiplexing can be configured such that a GPIO (C0INP/C1INP) pin provides the positive comparator input and/or a GPIO (C0INN/C1INN) provides the negative input. When connecting to GPIO, use the appropriate GPIO alternate function selection, as described in the General-Purpose Input/Output chapter on page 55.

The comparator output polarity is determined by the POLSEL bit. When POLSEL = 0, the comparator output is noninverted such that the comparator output is High when the positive comparator input voltage is greater than the negative comparator input voltage. When POLSEL = 1, the comparator output is inverted such that the comparator output is Low when the positive comparator input voltage is greater than the negative comparator input voltage.

The output of each comparator can be routed to a GPIO pin, C0OUT or C1OUT, as well as to the Event System. When connecting to GPIO, use the appropriate GPIO alternate function selection, as described in the General-Purpose Input/Output chapter on page 55. Additionally, the comparator output state can be read directly from the CSTATUS bit in the COMPCTL Register. An additional output, C01, is the logical OR of each comparator output, and is an Event System source; it is useful for window detection signaling.

A window compare feature provides coordinated detection reporting for the two comparators. WINEN = 1 selects Window Mode. The impact of WINEN and POLSEL on the comparator outputs is summarized in Table 256 on page 488.

24.4. Comparator and Reference System Register Definitions

This section defines the features of the following Comparator and Reference System registers:

Comparator Control Register (CMPCTL) at address `F8Fh`

Comparator 0 Control 0 Register (CMP0CTL0) at address `F90h`

Comparator 0 Control 1 Register (CMP0CTL1) at address `F91h`

Comparator 1 Control 0 Register (CMP1CTL0) at address `F92h`

Comparator 1 Control 1 Register (CMP1CTL1) at address `F93h`

27.3.5. Flash Programming Configuration

The Flash Programming Configuration Register, shown in Table 282, contains a PMODE bit that configures the number of bytes that are programmed simultaneously.

Table 282. Flash Programming Configuration Register (FPCONFIG)

Bits	7	6	5	4	3	2	1	0
Field	Reserved							PMODE
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W
Address	FFAh							

Bit	Description (Continued)
[7:1]	**Reserved** These bits are reserved and must be programmed to 0000000.
[0] PMODE	**Programming Mode** 0: Byte Programming Mode. 1: Word Programming Mode.

The read routine uses 16 bytes of stack space in addition to the 1 byte of address pushed by the user. Sufficient memory must be available for this stack usage. Because of the Flash memory architecture, NVDS reads exhibit a nonuniform execution time. A read operation takes between 71 µs and 258 µs (assuming a 20 MHz system clock). Slower system clock speeds result in proportionally higher execution times.

NVDS byte reads from illegal addresses (those exceeding the NVDS array size) return a value of `0xFF`. Illegal read operations have a 6 µs execution time. The status byte returned by the NVDS read routine is zero upon a successful read. A nonzero status byte indicates that there is a corrupted value in the NVDS array at the location being read. In this case, the value returned in R0 is the byte most recently written to the array that does not have an error.

Table 304. Read Status Byte

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved				IGADDR	RDERROR	Reserved	
Default Value	0	0	0	0	0	0	0	0

Bit	**Description**
[7:4]	**Reserved** All bits are reserved and must be programmed to 0000.
[3] IGADDR	**Illegal Address** 0: The write attempted was to a legal address (one that is within the NVDS array size). 1: The write attempted was to an illegal address (one that exceeds the NVDS array size).
[2] RDERROR	**NVDS Read Error** 0: No NVDS read error occurred. 1–7: An NVDS read error occurred. This bit is set when there is a possible corruption of the data at the address read.
[1:0]	**Reserved** All bits are reserved and must be programmed to 00.

29.2.3. Power Failure Protection

The NVDS routines employ error checking mechanisms to ensure that a power failure will endanger only the most recently written byte. Bytes previously written to an array are not perturbed.

A System Reset (such as a pin reset or a watchdog timer reset) that occurs during a write operation also perturbs the byte currently being written. All other bytes in the array remain unperturbed.

Table 325. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode		Op Code(s) (Hex)	Flags						Fetch Cycles	Instr. Cycles
		dst	src		C	Z	S	V	D	H		
AND dst, src	dst ⟵ dst AND src	r	r	52	–	*	*	0	–	–	2	3
		r	Ir	53							2	4
		R	R	54							3	3
		R	IR	55							3	4
		R	IM	56							3	3
		IR	IM	57							3	4
ANDX dst, src	dst ⟵ dst AND src	ER	ER	58	–	*	*	0	–	–	4	3
		ER	IM	59							4	3
ATM	Block all interrupt and DMA requests during execution of the next 3 instructions			2F	–	–	–	–	–	–	1	2
BCLR bit, dst	dst[bit] ⟵ 0	r		E2	–	*	*	0	–	–	2	2
BIT p, bit, dst	dst[bit] ⟵ p	r		E2	–	*	*	0	–	–	2	2
BRK	Debugger Break			00	–	–	–	–	–	–	1	1
BSET bit, dst	dst[bit] ⟵ 1	r		E2	–	*	*	0	–	–	2	2
BSWAP dst	dst[7:0] ⟵ dst[0:7]	R		D5	X	*	*	0	–	–	2	2
BTJ p, bit, src, dst	if src[bit]=p PC ⟵ PC + X		r	F6	–	–	–	–	–	–	3	3
			Ir	F7							3	4
BTJNZ bit, src, dst	if src[bit]=1 PC ⟵ PC + X		r	F6	–	–	–	–	–	–	3	3
			Ir	F7							3	4
BTJZ bit, src, dst	if src[bit]=0 PC ⟵ PC + X		r	F6	–	–	–	–	–	–	3	3
			Ir	F7							3	4
CALL dst	SP ⟵ SP -2 @SP ⟵ PC PC ⟵ dst	IRR		D4	–	–	–	–	–	–	2	6
		DA		D6							3	3
CCF	C ⟵ ~C			EF	*	–	–	–	–	—	1	2

Note: Flags notation:
*=Value is a function of the result of the operation.
–=Unaffected.
X=Undefined.
0=Reset to 0.
1=Set to 1.

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**
T_{SS}	Sample Settling Time[3]	0.4			µs	INMODE=0x
		0.8			µs	INMODE=1x, POWER=00 12-bit (RESOLUT=0)
		3.0			µs	INMODE=1x, POWER=10
		1.0			µs	INMODE=1x,POWER=00 14-bit (RESOLUT=1)
T_{CONV}	Conversion Time		13		ADC clock cycles	12-bit (RESOLUT=0)
			30			14-bit (RESOLUT=1)
T_{WAKE_AR}	Time for Wake up, Internal ADC Reference Buffer			T_{WAKE_ADC}	ADC clock cycles	REFSEL=10
			0.5	1.1	ms	REFSEL=11 C_{VREFP}=1µF
T_{WAKE_ADC}	Time for Wake up, ADC		30		ADC clock cycles	ADCREF = 0
f_{ADC_CLK}	Frequency of ADC clock			2	MHz	12-bit (RESOLUT=0); INMODE=0x; POWER=00
				5	MHz	12-bit (RESOLUT=0); INMODE=1x; POWER=00
				2	MHz	14-bit (RESOLUT=1); POWER=00
				1	MHz	12-bit (RESOLUT=0); POWER=10
				0.8	MHz	14-bit (RESOLUT=1); 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.