Zilog - Z16FMC64AG20EG Datasheet

Welcome to E-XFL.COM

Embedded - Microcontrollers - Application Specific: Tailored Solutions for Precision and Performance

Embedded - Microcontrollers - Application Specific represents a category of microcontrollers designed with unique features and capabilities tailored to specific application needs. Unlike general-purpose microcontrollers, application-specific microcontrollers are optimized for particular tasks, offering enhanced performance, efficiency, and functionality to meet the demands of specialized applications.

What Are Embedded - Microcontrollers - Application Specific?

Application specific microcontrollers are engineered to

Details	
Product Status	Active
Applications	Motor Control
Core Processor	Zneo™
Program Memory Type	FLASH (64kB)
Controller Series	Z16FMC
RAM Size	4K x 8
Interface	I²C, IrDA, LIN, SPI, UART/USART
Number of I/O	46
Voltage - Supply	2.7V ~ 3.6V
Operating Temperature	-40°C ~ 105°C
Mounting Type	Surface Mount
Package / Case	64-LQFP Exposed Pad
Supplier Device Package	64-LQFP
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z16fmc64ag20eg

Reset and Stop Mode Recovery

The reset controller within the Z16FMC Series controls the RESET and Stop Mode Recovery operations. In a typical operation, the following events cause a Reset to occur:

- Power-On Reset
- Voltage Brownout
- WDT time-out (when configured through the `WDT_RES` option bit to initiate a Reset)
- External $\overline{\text{RESET}}$ pin assertion
- OCD initiated Reset (`OCDCTL[0]` set to 1)
- Fault detect logic

When the Z16FMC is in STOP mode, a Stop Mode Recovery is initiated by either of the following:

- WDT time-out
- GPIO port input pin transition on an enabled Stop Mode Recovery source

Reset Types

The Z16FMC provides two different types of Reset operation (System Reset and Stop Mode Recovery). The type of Reset is a function of both the current operating mode of the Z16FMC device and the source of the Reset. Table 6 lists the types of Reset and their operating characteristics.

Table 6. Reset and Stop Mode Recovery Characteristics and Latency

	Reset Characteristics and Latency		
Reset Type	**Peripheral Control Registers**	**CPU**	**Reset Latency (Delay)**
System Reset	Reset (as applicable)	Reset	A minimum of 66 internal precision oscillator cycles.
Stop Mode Recovery	Unaffected, except RSTSRC and OSCCTL registers	Reset	A minimum of 66 internal precision oscillator cycles.

External Pin Reset

The input-only $\overline{\text{RESET}}$ pin has a schmitt-triggered input, an internal pull-up, an analog filter and a digital filter to reject noise. After the $\overline{\text{RESET}}$ pin is asserted for at least four system clock cycles, the device progresses through the System Reset sequence. While the $\overline{\text{RESET}}$ input pin is asserted Low, the Z16FMC device continues to be held in the Reset state. If the $\overline{\text{RESET}}$ pin is held Low beyond the System Reset time-out, the device exits the Reset state 16 system clock cycles following $\overline{\text{RESET}}$ pin deassertion. If the $\overline{\text{RESET}}$ pin is released before the System Reset time-out, the $\overline{\text{RESET}}$ pin is driven Low by the chip until the completion of the time-out as described in the next section. In STOP mode, the digital filter is bypassed as the system clock is disabled.

Following a System Reset initiated by the external $\overline{\text{RESET}}$ pin, the `EXT` status bit in the the Reset Status and Control Register (see page 35) is set to 1.

External Reset Indicator

During System Reset, the $\overline{\text{RESET}}$ pin functions as an open drain (active Low) RESET mode indicator in addition to the input functionality. This Reset output feature allows a Z16FMC device to Reset other components to which it is connected, even if the Reset is caused by internal sources such as POR, VBO, or WDT events and as an indication of when the reset sequence completes.

After an internal reset event occurs, the internal circuitry begins driving the $\overline{\text{RESET}}$ pin Low. The $\overline{\text{RESET}}$ pin is held Low by the internal circuitry until the appropriate delay listed in Table 6 on page 29 has elapsed.

User Reset

A System Reset is initiated by setting RSTSCR[0]. If the Write was caused by the OCD, the OCD is not Reset.

Fault Detect Logic Reset

Fault detect circuitry exists to detect *Illegal* state changes which is caused by transient power or electrostatic discharge events. When such a fault is detected, a system reset is forced. Following the system reset, the `FLTD` bit in the the Reset Status and Control Register (see page 35) is set.

Stop Mode Recovery

STOP mode is entered by execution of a `STOP` instruction by the CPU. For detailed information about STOP mode, see the Low-Power Modes chapter on page 36. During Stop

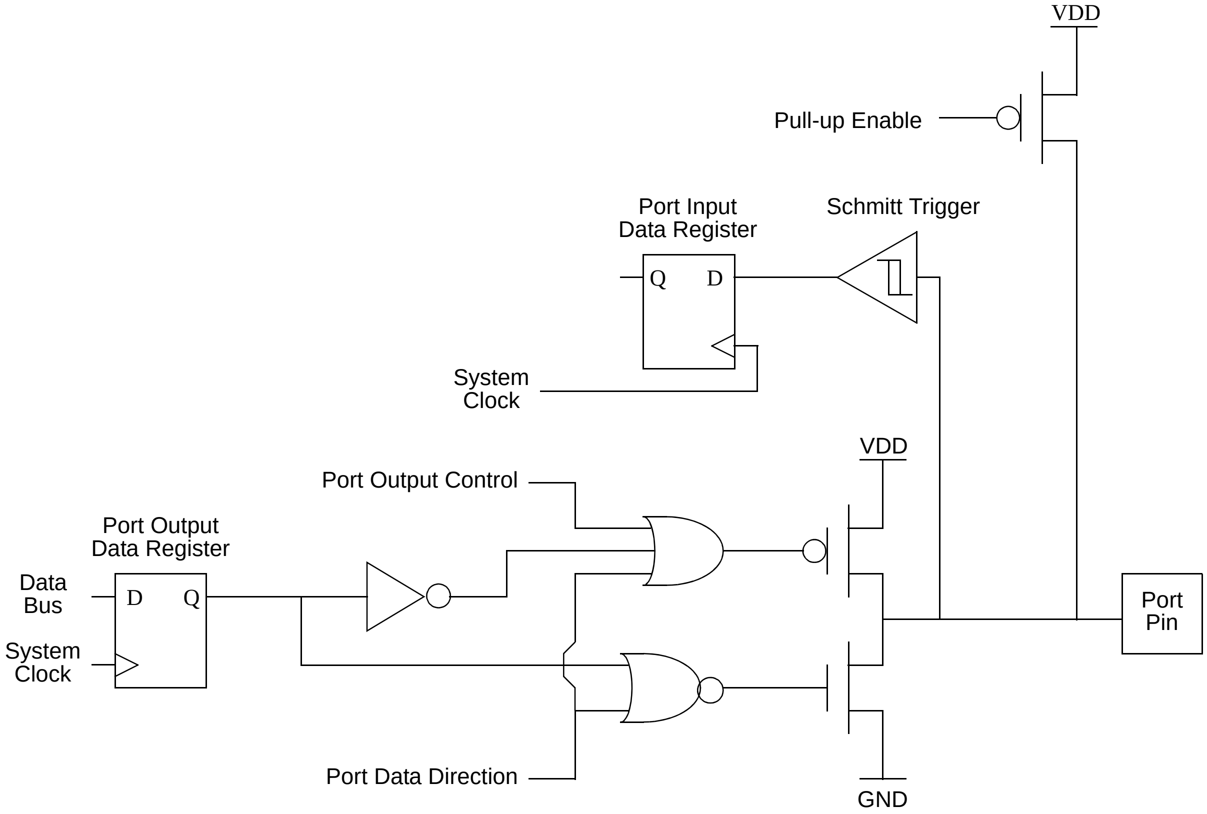

Figure 8. GPIO Port Pin Block Diagram

GPIO Alternate Functions

Many GPIO port pins are used for GPIO and to provide access to the on-chip peripheral functions such as timers and serial communication devices. The Port A–H alternate function registers configure these pins for either GPIO or alternate function operation. When a pin is configured for alternate function, control of the port pin direction (I/O) is passed from the Port A–H data direction registers to the alternate function assigned to this pin. Table 12 lists the alternate functions associated with each port pin.

Table 33. Interrupt Request 1 Register (IRQ1) and Interrupt Request 1 Set Register (IRQ1SET)

Field	PAD7I	PAD6I	PAD5I	PAD4I	PAD3I	PAD2I	PAD1I	PAD0I
RESET	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W
ADDR	FF_E035H							
Note: IRQ1SET at address FF_E035H is write only and used to set the interrupts identified.								

Bits	Description
7:0	**PAD*x*I – Port A/D Pin *x* Interrupt Request** 0 = No interrupt request is pending for GPIO port A/D pin *x*. 1 = An interrupt request from GPIO port A/D pin *x* is awaiting service. Writing 1 to these bits resets them to 0. Here *x* indicates the specific GPIO port pin number (0 through 7). PAD7I and PAD0I have interrupt sources other than Port A and Port D as selected by the Port A Irq Mux registers. PAD7I is configured to provide the comparator interrupt. PAD0I is configured to provide the OCD interrupt.

Note: These bits are set any time the selected port is toggled. The setting of these bits are not affected by the associated interrupt enable bits.

Interrupt Request 2 Register

The interrupt request 2 (IRQ2) Register (see Table 34) stores interrupt requests for both vectored and polled interrupts. When a request is presented to the interrupt controller, the corresponding bit in the IRQ2 Register becomes 1. If interrupts are globally enabled (vectored interrupts), the interrupt controller passes an interrupt request to the CPU. If interrupts are globally disabled (polled interrupts), the CPU reads the Interrupt Request 1 Register to determine, if any interrupt requests are pending. Writing 1 to the bits in this register clears the interrupt. The bits of this register are set by writing 1 to the interrupt request 2 set regsiter (IRQ2SET) at address `FF_E039H`.

$$\text{One-Shot Mode Time-Out Period(s)} = \frac{(\text{Reload Value} - \text{Start Value} + 1) \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

TRIGGERED ONE-SHOT Mode

In TRIGGERED ONE-SHOT mode, the timer operates as follows:

1. The timer is non-active until a trigger is received. The timer trigger is taken from the timer input pin. The TPOL bit in the Timer Control 1 Register selects whether the trigger occurs on the rising edge or the falling edge of the timer input signal.
2. Following the trigger event, the timer counts system clocks up to the 16-bit Reload value stored in the timer reload high and low byte registers.
3. After reaching the Reload value, the timer outputs a pulse on the timer output pin, generates an interrupt and resets the count value in the timer high and low byte registers to `0001H`. The duration of the output pulse is a single system clock. The TPOL bit also sets the polarity of the output pulse.
4. The timer now idles until the next trigger event. Trigger events, which occur while the timer is responding to a previous trigger is ignored.

Observe the following steps to configure timer 0 in TRIGGERED ONE-SHOT mode and initiate operation:

1. Write to the timer control registers to:
   - Disable the timer
   - Configure the timer for TRIGGERED ONE-SHOT mode
   - Set the prescale value
   - Set the initial output level (High or Low) via the TPOL bit for the timer output alternate function
   - Set the INTERRUPT mode
2. Write to the timer high and low byte registers to set the starting count value.
3. Write to the timer reload high and low byte registers to set the reload value.
4. Enable the timer interrupt, if required and set the timer interrupt priority by writing to the relevant interrupt registers.
5. When using the timer output function, configure the associated GPIO port pin for the timer output alternate function.
6. Write to the Timer Control 1 Register to enable the timer. Counting does not start until the appropriate input transition occurs.

The timer period is calculated by the following equation (Start Value = 1):

Table 46. Timer 0–2 Reload High Byte Register (TxRH)

Bits	7	6	5	4	3	2	1	0
Field	TRH							
RESET	FFH							
R/W	R/W							
ADDR	FF_E302H, FF_E312H, FF_E322H							

Bits	Description
7:0	**TRH – Timer Reload Register High** TRH is one of two bytes which form the 16-bit Reload value, {TRH[7:0], TRL[7:0]}. This value sets the maximum count value which initiates a timer reload to `0001H`.

Table 47. Timer 0–2 Reload Low Byte Register (TxRL)

Bits	7	6	5	4	3	2	1	0
Field	TRL							
RESET	FF							
R/W	R/W							
ADDR	FF_E303H, FF_E313H, FF_E323H							

Bits	Description
7:0	**TRL – Timer Reload Register Low** TRL is one of two bytes which form the 16-bit Reload value, {TRH[7:0], TRL[7:0]}. This value sets the maximum count value which initiates a timer reload to `0001H`.

Timer 0–2 PWM High and Low Byte Registers

The timer 0–2 PWM high and low byte (TxPWMH and TxPWML) registers (Tables 48 and 49) define PWM operations. These registers also store the timer counter values for the CAPTURE modes.

Table 48. Timer 0–2 PWM High Byte Register (TxPWMH)

Bits	7	6	5	4	3	2	1	0
Field	PWMH							
RESET	00H							
R/W	R/W							
ADDR	FF_E304H, FF_E314H, FF_E324H							

Bit Position	Value (H)	Description
7		**Timer Mode High Bit – TMODE[3]** This bit, along with the TMODE[2:0] field in T0CTL1 Register, determines the operating mode of the timer; it is the most significant bit of the timer mode selection value. For more details, see the T0CTL1 Register description.
6:5		**Timer Interrupt Configuration – TICONFIG** This field configures timer interrupt definitions. These bits affect all modes. The effect per mode is explained below: ONE SHOT, CONTINUOUS, COUNTER, PWM, COMPARE, DUAL PWM, TRIGGERED ONE-SHOT, COMPARATOR COUNTER: 0x Timer interrupt occurs on reload. 10 Timer interrupts are disabled. 11 Timer Interrupt occurs on reload. **GATED**: 0x Timer interrupt occurs on reload. 10 Timer interrupt occurs on inactive gate edge. 11 Timer interrupt occurs on reload. **CAPTURE, CAPTURE/COMPARE, CAPTURE RESTART**: 0x Timer interrupt occurs on reload and capture. 10 Timer interrupt occurs on capture only. 11 Timer interrupt occurs on reload only.
4		**Timer Cascade – CASCADE** This field allows the timers to be cascaded for larger counts. Only Counter Mode must be used with this feature.
	0	The timer is not cascaded.
	1	Timer is cascaded. If timer 0 CASCADE bit is set, ANALOG COMPARATOR output is used as input. If timer 1 CASCADE bit is set, the Timer 0 output is used as the input. If timer 2 CASCADE bit is set, the timer 1 output is used as input.

Bit Position	Value (H)	Description (Continued)
6		**Timer Input/Output Polarity – TPOL** This bit is a function of the current operating mode of the timer. It determines the polarity of the input and/or output signal. When the timer is disabled, the timer output signal is set to the value of this bit. **ONE-SHOT mode** – If the timer is enabled, the timer output signal pulses (changes state) for one system clock cycle after timer Reload. **CONTINUOUS mode** – If the timer is enabled, the timer output signal is complemented after timer Reload. **COUNTER mode** – If the timer is enabled, the timer output signal is complemented after timer reload. 0 = Count occurs on the rising edge of the timer input signal. 1 = Count occurs on the falling edge of the timer input signal. **PWM SINGLE OUTPUT mode** – When enabled, the timer output is forced to $\overline{\mathrm{TPOL}}$ after PWM count match and forced back to TPOL after Reload. **CAPTURE mode** – If the timer is enabled, the timer output signal is complemented after timer Reload. 0 = Count is captured on the rising edge of the timer input signal. 1 = Count is captured on the falling edge of the timer input signal. **COMPARE mode** – The timer output signal is complemented after timer Reload. **GATED mode** – The timer output signal is complemented after timer Reload. 0 = Timer counts when the timer input signal is High and interrupts are generated on the falling edge of the timer input. 1 = Timer counts when the timer input signal is Low and interrupts are generated on the rising edge of the timer input. **CAPTURE/COMPARE mode** – If the timer is enabled, the timer output signal is complemented after timer Reload. 0 = Counting starts on the first rising edge of the timer Input signal. The current count is captured on subsequent rising edges of the timer input signal. 1 = Counting starts on the first falling edge of the timer input signal. The current count is captured on subsequent falling edges of the timer input signal.

2. Load the appropriate 16-bit count value into the LIN-UART baud rate high and low byte registers.
3. Enable the BRG timer function and associated interrupt by setting the BRGCTL bit in the LIN-UART Control1 register to 1. Enable the UART receive interrupt in the interrupt controller.

When configured as a general purpose timer, the BRG interrupt interval is calculated using the following equation:

UART BRG Interrupt Interval (s) = System Clock Period (s) × BRG[15:0]

Noise Filter

A noise filter circuit is included to filter noise on a digital input signal, such as UART receive data, before the data is sampled by the block. This noise filter circuit is a requirement for protocols operating in a noisy environment.

The noise filter includes following features:

- Synchronizes the receive input data to the system clock
- Noise filter enable (`NFEN`)input selects whether the noise filter is bypassed (`NFEN =` 0) or included (`NFEN =` 1) in the receive data path
- Noise filter control (`NFCTL[2:0]`)input selects the width of the up/down saturating counter digital filter. The available widths range is from 4 to11 bits
- The digital filter output has hysteresis
- Provides an active low saturated state output (`FiltSatB`), used to indicate presence of noise

Architecture

Figure 20 displays how the noise filter is integrated with the LIN-UART for use on a LIN network.

Bits	Description (Continued)
0	**IREN – Infrared Encoder/Decoder Enable** 0 = Infrared encoder/decoder is disabled. LIN-UART operates normally. 1 = Infrared encoder/decoder is enabled. The LIN-UART transmits and receives data through the Infrared encoder/decoder.

Noise Filter Control Register (LIN-UART Control1 Register with MSEL = 001b).

When `MSEL` = `001b`, this register provides control for the digital noise filter.

Table 78. Noise Filter Control Register (UxCTL1 with MSEL = 001b)

Bits	7	6	5	4	3	2	1	0
Field	NFEN	NFCTL			Reserved			
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R
ADDR	FF_E203H, FF_E213H with `MSEL` = 001b							

Bits	Description
7	**NFEN – Noise Filter Enable** 0 = Noise filter is disabled. 1 = Noise filter is enabled. Receive data is preprocessed by the noise filter.
6:3	**NFCTL – Noise Filter Control** This field controls the delay and noise rejection characteristics of the noise filter. The wider the counter the more delay that is introduced by the filter and the wider the noise event that is filtered. 000 = 4-bit up/down counter 001 = 5-bit up/down counter 010 = 6-bit up/down counter 011 = 7-bit up/down counter 100 = 8-bit up/down counter 101 = 9-bit up/down counter 110 = 10-bit up/down counter 111 = 11-bit up/down counter
2:0	These bits are reserved.

The LIN-UART data rate is calculated using the following equation for standard UART modes. For LIN protocol, the baud rate registers must be programmed with the baud period rather than 1/16 baud period.

Note: The UART must be disabled when updating the baud rate registers because High and Low registers must be written independently.

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

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

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

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

For a given LIN-UART data rate, the integer baud rate divisor value is calculated using the following equation for standard UART operation:

$$\textbf{UART Baud Rate Divisor Value (BRG)} = \textbf{Round}\left(\frac{\textbf{System Clock Frequency (Hz)}}{\textbf{16} \times \textbf{UART Data Rate (bits/s)}}\right)$$

For a given LIN-UART data rate, the integer baud rate divisor value is calculated using the following equation for LIN mode UART operation:

$$\textbf{UART Baud Rate Divisor Value (BRG)} = \textbf{Round}\left(\frac{\textbf{System Clock Frequency (Hz)}}{\textbf{UART Data Rate (bits/s)}}\right)$$

The baud rate error relative to the appropriate baud rate is calculated using the following equation:

$$\textbf{UART Baud Rate Error (\%)} = \textbf{100} \times \left(\frac{\textbf{Actual Data Rate} - \textbf{Desired Data Rate}}{\textbf{Desired Data Rate}}\right)$$

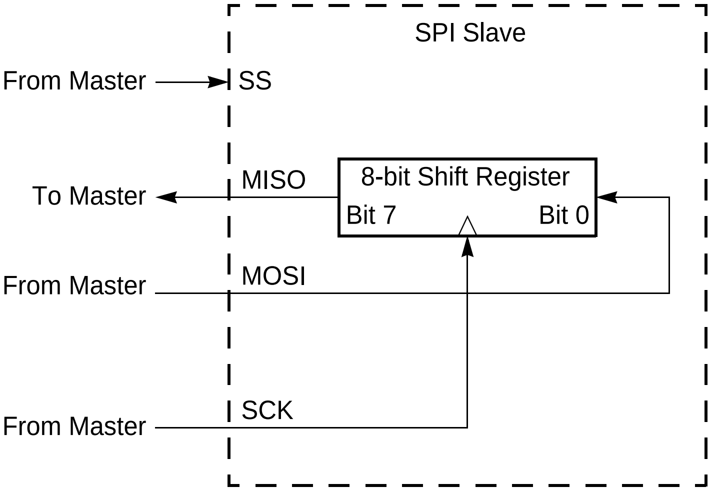

Figure 32. ESPI Configured as an SPI Slave

Error Detection

Error events detected by the ESPI block are described in this section. Error events generate an ESPI interrupt and set a bit in the ESPI status register. The error bits of the ESPI Status Register are read/write 1 to clear.

Transmit Underrun

A transmit underrun error occurs for a master with SSMD = 10 or 11 when a character transfer completes and TDRE = 1. In these modes when a transmit underrun occurs the transfer is aborted (SCK will halt and SSV will be deasserted). For a master in SPI mode (`SSMD = 00`), a transmit underrun is not signaled because SCK will pause and wait for the data register to be written.

In SLAVE mode, a transmit underrun error occurs if TDRE = 1 at the start of a transfer. When a transmit underrun occurs in SLAVE mode, ESPI transmits a character of all 1s.

A transmit underrun sets the `TUND` bit in the ESPI status register to 1. Writing 1 to `TUND` clears this error flag.

Mode Fault (Multi-Master Collision)

A mode fault indicates when more than one master is trying to communicate at the same time (a multi-master collision) in SPI mode. The mode fault is detected when the enabled master's $\overline{\text{SS}}$ input pin is asserted. For this to happen the control and mode registers must be configured with MMEN = 1, SSIO = 0 ($\overline{\text{SS}}$ is an input) and $\overline{\text{SS}}$ input = 0. A mode fault sets the `COL` bit in the ESPI status register to 1. Writing a 1 to `COL` clears this error flag.

I^2C Baud Rate High and Low Byte Registers

The I^2C Baud Rate High and Low Byte registers (see Tables 102 and 103) combine to form a 16-bit reload value, `BRG`[15:0], for the I^2C Baud Rate Generator. The baud rate High and Low Byte Registers must be programmed for the I^2C baud rate in slave mode as well as in master mode. In slave mode, the baud rate value programmed must match the master's baud rate within +/- 25% for proper operation.

The I^2C baud rate is calculated using the below equation.

Note: If `BRG` = `0000H`, use `10000H` in the equation.

$$\text{I2C Baud Rate (bps)} = \frac{\text{System Clock Frequency (Hz)}}{4 \times \text{BRG[15:0]}}$$

Table 102. I^2C Baud Rate High Byte Register (I2CBRH)

Bits	7	6	5	4	3	2	1	0
Field	BRH							
RESET	FFH							
R/W	R/W							
ADDR	FF_E243H							

Bits	Description
[7:0] BRH	**I^2C Baud Rate High Byte** Most significant byte, `BRG`[15:8], of the I^2C Baud Rate Generator’s reload value.

Note: If the DIAG bit in the I2C Mode Register is set to 1, a read of the I2CBRH Register returns the current value of the I2C Baud Rate Counter[15:8].

The operational amplifier is a two-input, one-output operational amplifier with a typical open loop gain of 10,000 (80 dB). The general-purpose input pin (OPINP) provides the non-inverting amplifier input, while general-purpose input pin (OPINN) provides the inverting amplifier input. The output is available at the output pin (OPOUT).

The key operating characteristics of the operational amplifier are:

- Frequency compensated for unity gain stability
- Input common-mode-range from GND (0.0 V) to VDD–1 V
- Input offset voltage less than 15 mV
- Output voltage swing from GND + 0.1 V to V_{DD}–0.1 V
- Input bias current less than 1 µA
- Operating the operational amplifier open loop (no feedback) effectively provides another on-chip comparator

Comparator Operation

The comparator output reflects the relationship between the non-inverting input and the inverting (reference) input. If the voltage on the non-inverting input is higher than the voltage on the inverting input, the comparator output is at a high state. If the voltage on the non-inverting input is lower than the voltage on the inverting input, the comparator output is at a low state.

To operate, the comparator must be enabled by setting the `CMPEN` bit in the comparator and op-amp register to 1. In addition the CINP and CINN comparator input alternate functions must be enabled on their respective GPIO pins. For more information, see the GPIO Alternate Functions section on page 39.

The comparator does not automatically power-down. To reduce operating current when not in use, the comparator is disabled by clearing the `CMPEN` bit to 0.

Operational Amplifier Operation

To operate, the operational amplifier must be enabled by setting the `OPEN` bit in the comparator and op-amp register to 1. In addition, the OPINP, OPINN and OPOUT alternate functions must be enabled on their respective general-purpose I/O pins. For more information, see the GPIO Alternate Functions section.

The logical value of the operational amplifier output (OPOUT) is read from the Port 3 data input register if both the operational amplifier and input pin Schmitt trigger are enabled. For more information, see the GPIO Alternate Functions section. The operational amplifier generates an interrupt via the GPIO Port B3 input interrupt, if enabled.

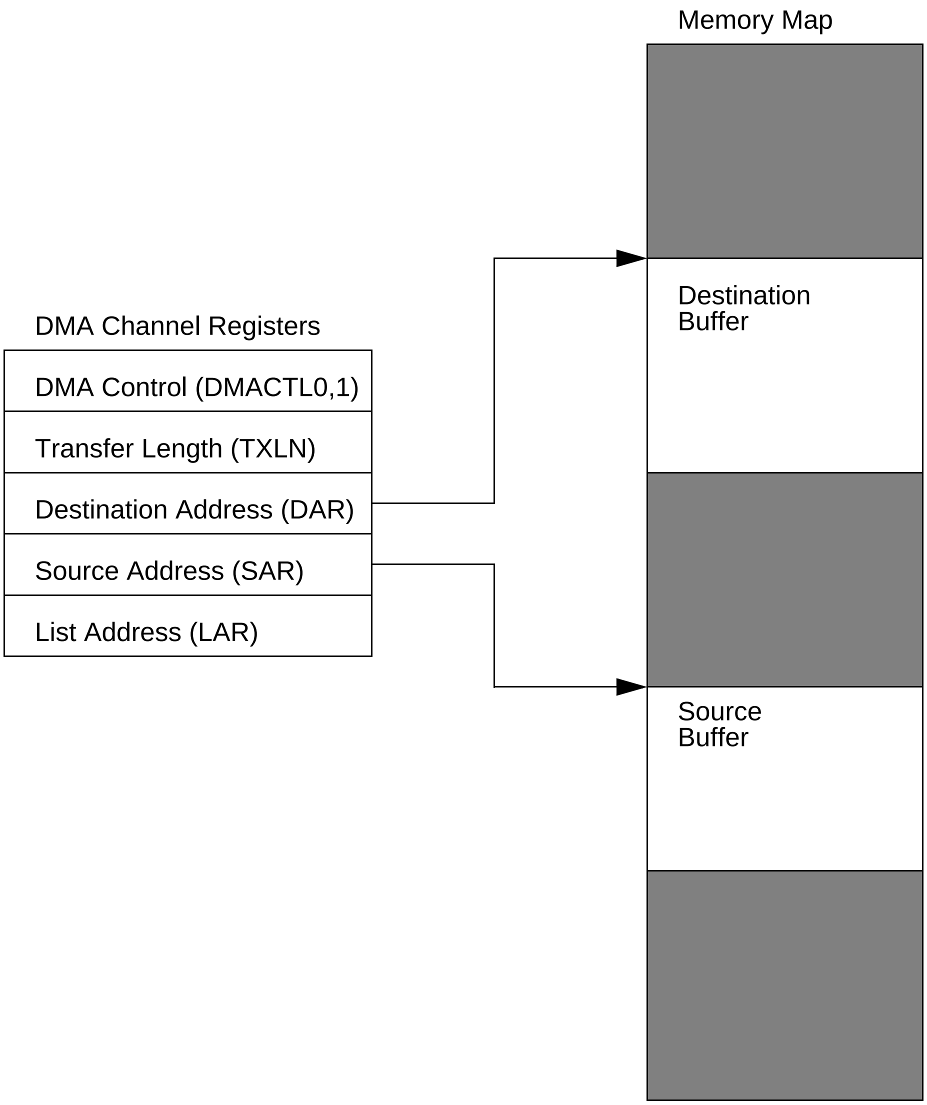

Figure 47. Direct DMA Diagram

Direct DMA Setup and Operation

Observe the following steps to set up the DMA in direct mode:

1. Write the DMAxREQSEL to select the request source.
2. Write the DMAxDAR register with the destination address.
3. Write the DMAxSAR register with the source address.

Flash Control Register

The Flash Control Register selects how the Flash memory is accessed.

Table 128. Flash Control Register (FCTL)

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	INFO	Reserved						
RESET	0	00H						
R/W	R/W	R						
ADDR	FF_E061H							

Bits	**Description**
7 INFO	**Information Area Access** This bit selects access to the information area. 0 = Information Area is not selected. 1 = Information Area is selected. The Information area is mapped into the Program memory address space at addresses `000000H` through `00007FH`.
6:0	**Reserved** These bits are reserved and must be written to zero.

Flash Sector Protect Register

The Flash Sector Protect register (see Table 129) protects Flash memory sectors from being programmed or erased from user code. User code can only write bits in this register to 1 (bits cannot be cleared to 0 by user code).

Table 129. Flash Sector Protect Register (FSECT)

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	SECT7	SECT6	SECT5	SECT4	SECT3	SECT2	SECT1	SECT0
RESET	0	0	0	0	0	0	0	0
R/W	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1
ADDR	FF_E062H							
R/W1 = Register is accessible for Read operations. Register is written to 1 only (via user code).								

Bits	**Description**
[7:0] SECT*n*	**Sector Protect** 0 = Sector *n* is programmed or erased from user code. 1 = Sector *n* is protected and cannot be programmed or erased from user code.
Note: *User code write bits from 0 to 1 only.	

Option Bits

Option bits allow user configuration of certain aspects of the Z16FMC operation. The feature configuration data is stored in the Program memory and read during Reset. The features available for control using the option bits are:

- WDT time-out response selection – interrupt or Reset
- WDT enabled at Reset
- The ability to prevent unwanted read access to user code in Program memory
- The ability to prevent accidental programming and erasure of the user code in Program memory
- Voltage Brownout (VBO) configuration – always enabled or disabled during STOP mode to reduce STOP mode power consumption
- Oscillator mode selection for high, medium and low power crystal oscillators, or external RC oscillator
- PWM pin set up for motor control application

Option Bit Configuration By Reset

Each time the option bits are programmed or erased, the device must be Reset for the change to take place. During any reset operation (System Reset, Short Reset, or Stop Mode Recovery), the option bits are automatically read from the Program memory and written to Option Configuration registers. The Option Configuration registers control operation of the device. Option Bit control register are loaded before the device exits Reset and the CPU begins code execution. The Option Configuration registers are not part of the Register file and are not accessible for read or write access.

Option Bit Address Space

The first four bytes of Program Memory at addresses `0000H` through `0003H` (see Table 132) are reserved for the user option bits. These bytes are used to configure user specific options. You can change the option bits to meet the application needs.

Program Memory Address 0000H

Option bits in this space are altered to change the chip configuration at reset.

Baud Rate Reload Register

The Baud Rate Reload Register (DBGBR) is used to configure the baud rate of the serial communication stream. This register is automatically set by the Auto-Baud Detector. This register cannot be written by the CPU when `OCDLOCK` is set.

Table 142. Baud Rate Reload Register (DBGBR)

Bits	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	RELOAD															
RESET	0000H															
R/W	R/W															
ADDR	FF_E082-FF_E083															

Bits	**Description**
[15:0] RELOAD	This value is the baud rate reload value used to generate a bit clock. It is calculated as: $\text{RELOAD} = \frac{\text{SYSTEM CLOCK}}{\text{BAUD RATE}} \times 8$

Line Control Register

The Line Control Register (DBGLCR) controls the state of the UART. This register cannot be written by the CPU when `OCDLOCK` is set.

Table 143. Line Control Register (DBGLCR)

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	OE	TDH	HDS	TXFC	NBEN	NB	OUT	PIN
RESET	0	0	0	0	0	0	1	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
ADDR	FF_E084							

Bits	**Description**
7 OE	**Output Enable** This bit controls the output driver. If the UART is enabled, this bit controls the output driver during transmission only. 0 = Pin is open-drain during UART transmit. Pin behaves as an input if UART is disabled. 1 = Pin is driven during transmission if UART is enabled. Pin is an output if UART is disabled.

Table 153. Oscillator Control Register (OSCCTL)

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	INTEN	XTLEN	WDTEN	POFEN	WDFEN	FLPEN	SCKSEL	
RESET	1	0	1	0	0	0*	00	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ADDR	FF_E0A0H							

Note: *The reset value is 1 if the option bit LPOPT is 0.

Bit Position	**Value (H)**	**Description**
[7] INTEN		Internal Precision Oscillator Enable
	0	Internal precision oscillator is disabled.
	1	Internal precision oscillator is enabled.
[6] XTLEN		Crystal Oscillator Enable
	0	Crystal oscillator is disabled.
	1	Crystal oscillator is enabled.
[5] WDTEN		WDT Oscillator Enable
	0	WDT oscillator is disabled.
	1	WDT oscillator is enabled.
[4] POFEN		Primary Oscillator Failure Detection Enable
	0	Failure detection and recovery of primary oscillator is disabled. This bit is cleared automatically if a primary oscillator failure is detected.
	1	Failure detection and recovery of primary oscillator is enabled.
[3] WDFEN		WDT Oscillator Failure Detection Enable
	0	Failure detection of WDT oscillator is disabled.This bit is cleared automatically if a WDT oscillator failure is detected.
	1	Failure detection of WDT oscillator is enabled.
[2] FLPEN		Flash Low Power Mode Enable
	0	Flash Low Power Mode is disabled.
	1	Flash Low Power Mode is enabled. The Flash will be powered down during idle periods of the clock and powered up during Flash reads. This bit must only be set if the frequency of the primary oscillator source is 8 MHz or lower. The reset value of this bit is controlled by the LP0PT option bit during reset.
[1:0] SCKSEL		System Clock Oscillator Select
	00	Internal precision oscillator functions as system clock at 5.6 MHz.
	01	Crystal oscillator or external clock driver functions as system clock.
	10	Reserved.
	11	Watchdog Timer oscillator functions as system clock.