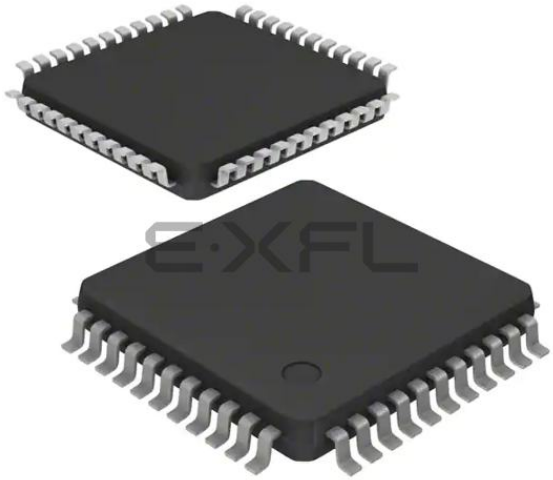

Welcome to E-XFL.COM

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	32KB (32K 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/z8f3281an024xk

Table 8. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page #
F76	ADC Data High	ADCD_H	00	458
F77	ADC Data Low	ADCD_L	00	458
F78	Sample Time	ADCST	00	459
F79	ADC Upper Window Threshold High	ADCUWINH	FF	460
F7A	ADC Upper Window Threshold Low	ADCUWINL	FF	461
F7B	ADC Lower Window Threshold High	ADCLWINH	00	462
F7C	ADC Lower Window Threshold Low	ADCLWINL	00	463
Digital-to-Analog Converter (DAC)				
F7D	DAC Control	DACCTL	00	468
F7E	DAC Data High	DACD_H	00	470
F7F	DAC Data Low	DACD_L	00	470
Low-Power Control				
F80	Power Control 0	PWRCTL0	10	52
F81	Power Control 1	PWRCTL1	00	54
Clock System				
F82	Clock Control 0	CLKCTL0	00	115
F83	Clock Control 1	CLKCTL1	01	116
F84	Clock Control 2	CLKCTL2	00	117
F85	Clock Control 3	CLKCTL3	08	119
F86	Clock Control 4	CLKCTL4	00	119
F87	Clock Control 5	CLKCTL5	05	120
F88	Clock Control 6	CLKCTL6	00	121
F89	Clock Control 7	CLKCTL7	00	122
F8A	Clock Control 8	CLKCTL8	XX	122
F8B	Clock Control 9	CLKCTL9	XX	123
F8C	Clock Control A	CLKCTLA	00	123
F8D	Clock Control B	CLKCTLB	00	125
F8E	Clock Control C	CLKCTLC	00	126
Comparators				
F8F	Comparator Control	CMPCTL	00	492
F90	Comparator 0 Control 0	CMP0CTL0	00	493
F91	Comparator 0 Control 1	CMP0CTL1	00	494

- If enabled for operation in Stop Mode by the associated Flash option bits, the VBO protection circuit continues operating; the LVD circuit continues to operate if enabled by the Power Control Register 0.
- Operational Amplifiers, comparators, and Temperature Sensor continue to operate if both enabled by the Power Control Register 0 and FRECOV=1.
- LCD continue to operate if enabled by the Power Control Register 0.
- All other on-chip peripherals are idle.

To minimize current in Stop Mode, all GPIO pins which are configured as digital inputs must be driven to one of the supply rails (V_{DD} or GND). The device is brought out of Stop Mode using Stop-Mode Recovery. To learn more about Stop-Mode Recovery, see the Reset, Stop-Mode Recovery and Low-Voltage Detection chapter on page 38.

6.2. Halt Mode

Executing the eZ8 CPU's Halt instruction places the device into Halt Mode. In Halt Mode, the operating characteristics are:

- Any enabled crystal oscillator continues to operate
- System clock is enabled and continues to operate
- eZ8 CPU is stopped
- Program counter (PC) stops incrementing
- If enabled, the WDT continues to operate
- All other on-chip peripherals continue to operate

The eZ8 CPU can be brought out of Halt Mode by any of the following operations:

- Interrupt
- Watchdog Timer time-out (Interrupt or Reset)
- Power-On Reset
- Voltage Brown-Out Reset
- External $\overline{\text{RESET}}$ pin assertion

To minimize current in Halt Mode, all GPIO pins which are configured as inputs must be driven to one of the supply rails (V_{DD} or GND).

Table 21. Port Alternate Function Mapping, 80-Pin Parts (Continued)

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Subregisters
Port B[1]	PB0	Reserved		AFS1[0]: 0
		ANA0/AMPAOUT	ADC Analog Input/Op Amp A Output	AFS1[0]: 1
	PB1	Reserved		AFS1[1]: 0
		ANA1/AMPAINN	ADC Analog Input/Op Amp A Input (N)	AFS1[1]: 1
	PB2	Reserved		AFS1[2]: 0
		ANA2/AMPAINP	ADC Analog Input/Op Amp A Input (P)	AFS1[2]: 1
	PB3	Reserved		AFS1[3]: 0
		V_{REF}–	Voltage Reference (M)	AFS1[3]: 1
	PB4	Reserved		AFS1[4]: 0
		V_{REF}+	Voltage Reference (P)	AFS1[4]: 1
	PB5	Reserved		AFS1[5]: 0
		ANA9	ADC Analog Input	AFS1[5]: 1
	PB6	Reserved		AFS1[6]: 0
		ANA10	ADC Analog Input	AFS1[6]: 1
	PB7	Reserved		AFS1[7]: 0
		Reserved		AFS1[7]: 1

Notes
1. Because there are at most two choices of alternate function for some pins of Ports A, B, D, E, F and G, the Alternate Function Set Subregister AFS2 is not implemented. Also, 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 pin in ports H and J, the Alternate Function Set subregisters are not implemented for these two ports. Also, 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).

not recommend DMA for One-Shot Compare operation, because upon a channel interrupt, the channel is disabled, thereby clearing its DMA request.

- DMA request behavior is a function of channel mode. The behavior in the first paragraph under DMA would pertain to all modes except Capture Mode.

11.5. Low-Power Modes

11.5.1. Operation in Halt Mode

When the eZ8 CPU is operating in Halt Mode, the Multi-Channel Timer will continue to operate if enabled. To minimize current in Halt Mode, the Multi-Channel Timer must be disabled by clearing the TEN control bit.

11.5.2. Operation in Stop Mode

When the eZ8 CPU is operating in Stop Mode, the Multi-Channel Timer ceases to operate as the System Clock is stopped. The registers are not reset and operation will resume after Stop-Mode Recovery occurs.

11.5.3. Power Reduction During Operation

Deassertion of the TEN bit will inhibit clocking of the entire Multi-Channel Timer block. Deassertion of the CHEN bit of individual channels will inhibit clocking of channel specific logic to minimize power consumption of unused channels. The CPU can still read/write registers when the enable bit(s) are deasserted.

11.6. Multi-Channel Timer Application Examples

11.6.1. PWM Programmable Deadband Generation

The Count Up/Down Mode supports motor control applications that require dead time between output signals. Figure 24 shows dead-time generation between two channels operating in Count Up/Down Mode.

is enabled. When MCT is disabled, reads from MCTL read the register directory. The MCT High and Low Byte registers are not reset when TEN=0.

Table 90. MCT High Byte Register (MCTH)

Bit	7	6	5	4	3	2	1	0
Field	MCTH							
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	FA0h							

Table 91. MCT Low Byte Register (MCTL)

Bit	7	6	5	4	3	2	1	0
Field	MCTL							
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	FA1h							

Bit	Description
[7:0] MCTH, MCTL	**MCT High and Low Bytes** These 2 bytes, {MCTH[7:0], MCTL[7:0]}, contain the current 16-bit MCT count value.

11.7.3. MCT Reload High and Low Byte Registers

The MCT Reload High and Low Byte (MCTRH and MCTRL) registers, shown in Tables 92 and 93, store a 16-bit reload value, {MCTRH[7:0], MCTRL[7:0]}. When TEN=0, writes to this address update the register on the next clock cycle. When TEN=1, writes to this register are buffered and transferred into the register when the counter reaches the end of the count cycle.

$$\text{Modulo Mode Period} = \frac{\text{Prescale Value} \times (\text{Reload Value} + 1)}{f_{MCTCLK}}$$

$$\text{Up/Down Mode Period} = \frac{2 \times \text{Prescale Value} \times \text{Reload Value}}{f_{MCTCLK}}$$

A value written to the MCTRH is stored in a temporary holding register. When a write to the MCTRL occurs, the temporary holding register value is written to the MCTRH. This operation allows simultaneous updates of the 16-bit MCT reload value.

11.7.5. MCT Subregister x

The MCT Subregister x, in which x = 0, 1, or 2 (see Table 95), stores an 8-bit data write to a subregister or an 8-bit data read from a subregister. The MCT Subaddress Register selects the subregister to be written to or read from.

Table 95. MCT Subregister *x* (MCTSR*x*)

Bit	7	6	5	4	3	2	1	0
Field	MCTSR*x*							
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	MCTSR0 @ FA5h, MCTSR1 @ FFA6h, MCTSR2 @ FFA7h							
Note: *x* references bits in the range [2:0].								

Bit	Description
[7:0] MCTSR*x*	

11.7.6. Multi-Channel Timer Control 0 and Control 1 Registers

The Multi-Channel Timer Control 0 and 1 registers (MCTCTL0, MCTCTL1), shown in Tables 96 and 97, control multi-channel timer operation. Writes to the PRES field of the MCTCTL1 Register are buffered when TEN=1, and will not take effect until the next end-of-cycle count occurs.

Table 96. Multi-Channel Timer Control 0 Register (MCTCTL0)

Bit	7	6	5	4	3	2	1	0
Field	TCTST	CHST	TCIEN	Reserved		TCLKS		
RESET	0	0	0	0	0	0	0	0
R/W	R/W1C	R	R/W	R	R	R/W	R/W	R/W
Address	00h in Subaddress Register, accessible through Subregister 0							

Bit	Description
[7] TCTST	**Timer Count Status** This bit indicates if a timer count cycle is complete and is cleared by writing 1 to the bit and is cleared when TEN=0. 0: Timer count cycle is not complete. 1: Timer count cycle is complete.

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.

the associated DMA request. The desired endpoint buffer space is selected using the EPSEL field in the USBDMA0CTL (or USBDMA1CTL) Register. A DMA request is asserted by setting the STARTDMA bit. When asserted, a DMA request will remain asserted until the DMA Controller transfers the last byte (based on the DMA count). A DMA request will also be deasserted if the DMA Controller attempts to transfer more than 64 bytes.

Software typically initiates a DMA transfer from an OUT endpoint following a USB interrupt to service the OUT endpoint buffer memory space indicated by OUTxIRQ in the USBOUTIRQ Subregister. Software reads the corresponding USB OUT x Byte Count Subregister (USBOxBC) containing the number of bytes to be transferred, and will configure the appropriate DMA x Count Subregister in the DMA Controller. In the DMA Controller, the DMA source address should be configured with the USBDMADATA Register address, and fixed-source addressing should be selected. Typically, the destination address is configured to be the Register RAM. Software then writes the USBDMA0CTL (or USBDMA1CTL) Register to select the endpoint buffer memory and initiate assertion of a DMA request.

Software typically initiates a DMA transfer to an IN endpoint buffer following a USB interrupt to service the IN endpoint indicated by INxIRQ in the USBINIRQ Subregister. The USBDMADATA Register address is the DMA destination address, and should be accessed with fixed addressing. When the next IN endpoint buffer data is available, software configures a DMA channel and writes to the USBDMA0CTL (or USBDMA1CTL) Register to select the appropriate IN endpoint and initiate assertion of a DMA request. When the DMA completes, software should write to the USBIxBC Subregister to arm the USB IN endpoint.

17.3. USB Control Register Definitions

Seven registers provide access to the USB Module: three registers for USB special function registers (SFRs) and endpoint buffer memory, three registers for DMA control and data, and one register for interrupt control. Table 169 lists these USB registers. The USB Subaddress Register (USBSA), USB Subdata Register (USBSD), and USB Control Register (USBCTL) together provide access to the subregisters that control USB SFRs and endpoint buffer spaces.

Table 169. USB Registers and Subregisters

USB Register Mnemonic	Address	USB Register Name
USBSA	F59h	USB Subaddress Register
USBSD	F5Ah	USB Subdata Register
USBCTL	F5Bh	USB Control Register
USBDMA0CTL	F5Ch	USB DMA 0 Control Register

Note: *The DMASA bit in the DMASA Register contains the subregister address.

18.2.8.1. Linked List Descriptor Request Priority

Because a linked list descriptor fetch has the highest priority, other DMA requests are ignored while a descriptor is being transferred from the Register File to the DMA registers.

18.2.8.2. Round Robin Priority

Round-robin priority is the default at System Reset, and is selected by clearing the PRIORITY bit. With round-robin priority, each channel request is serviced for the length of its burst size or until it deasserts a DMA request, whichever occurs first. After a channel is serviced, it is assigned lowest priority and is taken out of the rotation until all other requesting channels are serviced.

18.2.8.3. Fixed Priority

Fixed priority is selected by setting the PRIORITY bit. With fixed priority, channel requests are serviced with the following priority order from highest to lowest: DMA0, DMA1, DMA2, DMA3. Lower-priority DMA channels are not serviced until higher-priority DMA channels deassert DMA request.

18.2.9. Interrupts

Independent interrupt control is provided for end-of-count and watermark interrupts. An indication of whether the most recent interrupt was due to an end-of-count or watermark is provided by the IRQS bit in the DMA 0–3 Subregister Selection and Status registers.

18.2.10. End-of-Count Interrupt

An interrupt is generated when the end-of-count is reached. If interrupted on an end-of-count, the EOCIRQE bit is set in the DMAxCTL0 Subregister. The EOCIRQE bit can be cleared if the application is not required to service the buffer, or will service the buffer in conjunction with a future buffer.

18.2.11. Watermark Interrupt

The DMA Controller is able to generate an interrupt prior to an end-of-count being reached. If the value of the WMCNT bit in the DMAxCNTH Subregister matches the current count, {CNTH, CNTL}, in the DMAxCNTH and DMAxCNTL subregisters, a watermark interrupt will be generated. To disable watermark interrupts, configure WMCNT=0h.

Only a single watermark interrupt will be generated for a given count value.

A variety of peripherals and GPIOs can be selected to be an Event System channel signal source. In addition, software can assert a channel (High) by writing CHSRCSEL=01h. In this case, the channel will remain asserted (High) until CHSRCSEL is written with a value other than 01h. When selecting a GPIO as an Event System channel signal source, no additional configuration of the GPIO port alternate function selection registers is required.

The ESCHxSRC subregisters are accessed using the Event System Source Subaddress Register (ESSSA) and Event System Source Subdata Register (ESSSD). An ESCHxSRC Subregister is selected by ESSSA in the ESSSA Register and is accessed by writing/reading ESSSD in the ESSSD Register.

Table 217 lists the available Event System signal sources.

Table 217. Event System Signal Sources

Peripheral	Output Signal
Software	ESCHxCTL=01h
Timer 0	Timer 0 out
	$\overline{\text{Timer 0 out}}$
Timer 1	Timer 1 out
	$\overline{\text{Timer 1 out}}$
Timer 2	Timer 2 out
	$\overline{\text{Timer 2 out}}$
Multi-Channel Timer	Multi-Channel Timer out A
	Multi-Channel Timer out B
	Multi-Channel Timer out C
	Multi-Channel Timer out D
RTC	Prescaled clock
	Alarm
Comparator 0	Comparator 0 out
Comparator 1	Comparator 1 out
Comparator 0/1	Comparator 0/1 window detection (C01)
Port A	Port A[7:0] pin
Port C	Port C[7:4] pin
Port E	Port E[6:3] pin

formed only on the channels selected in ANAINL and ANAINH. Channels that are not selected are skipped.

The ADC configuration is identical for each channel scanned as defined in the ADC control registers. Timing parameters, ST and SST, should be configured for the requirements of the worst-case channel. If single-shot conversion is selected (CONTCONV=0), the ADC performs the scan sequence once. If continuous conversion is enabled, the ADC repeats the scan sequence in a continuous fashion.

An interrupt can be generated for each channel conversion result. The data for each channel result can be moved by software or DMA.

Note: ADC scanning continues while the F6482 Series device is in Debug Mode, during which the CPU fetch unit stops. This activity can result in a loss of synchronization between user code and ADC scanned data.

21.2.3.3. Conversion Averaging

The ADC is capable of processing data from multiple individual conversions to form an averaged result. When averaging is enabled by setting the AVE bit, AVESAMP determines whether 2, 4, 8, or 16 samples are averaged to produce a result. An interrupt will be generated only when a final sample is obtained and processed into the average. If channel scanning is enabled, the averaged result is obtained sequentially for each channel being scanned.

21.2.3.4. Power Control

The ADC is capable of performing conversions at two different power settings, as selected by the POWER bit. When POWER=00, the ADC runs at higher current consumption, and can be clocked at up to 5MHz. When POWER=10, the ADC runs at lower current consumption and can be clocked at up to 1MHz. The lower power consumption setting can reduce overall current consumption for longer sampling times because the current consumption during sampling is reduced.

21.2.3.5. Resolution

When the RESOLUT bit is cleared, 12-bit conversions are performed. For applications that require even higher resolution, 2-pass 14-bit resolution conversions are performed when RESOLUT is set. These conversions involve somewhat longer timings than those described in the 2-Pass 14-Bit Resolution Timing section on page 447. When performing 2-pass 14-bit conversion, Zilog recommends using the following input mode selection: INMODE=01.

21.2.4. Starting and Stopping Conversions

ADC activity is initiated by writing the START bits in the ADC Control 0 Register to perform a conversion (START=01), offset calibration (START=10), or gain calibration

26.2.8.4. 1/4 Duty Mode

1/4 Duty Mode configurations are shown in Table 265 on page 509, and are selected with the LCDMODE bit in the LCDCTL2 Register. In this mode, all commons (COM[3:0]) are used, and each SEGx drives up to four LCD display segments. Example waveforms for 1/4 Duty Mode with 1/3 bias are shown in Figure 92 (Type A) and Figure 93 (Type B).

Memory Map Example									
COM3	COM2	COM1	COM0		COM3	COM2	COM1	COM0	
1	0	1	1	SEG01	1	1	0	1	SEG00

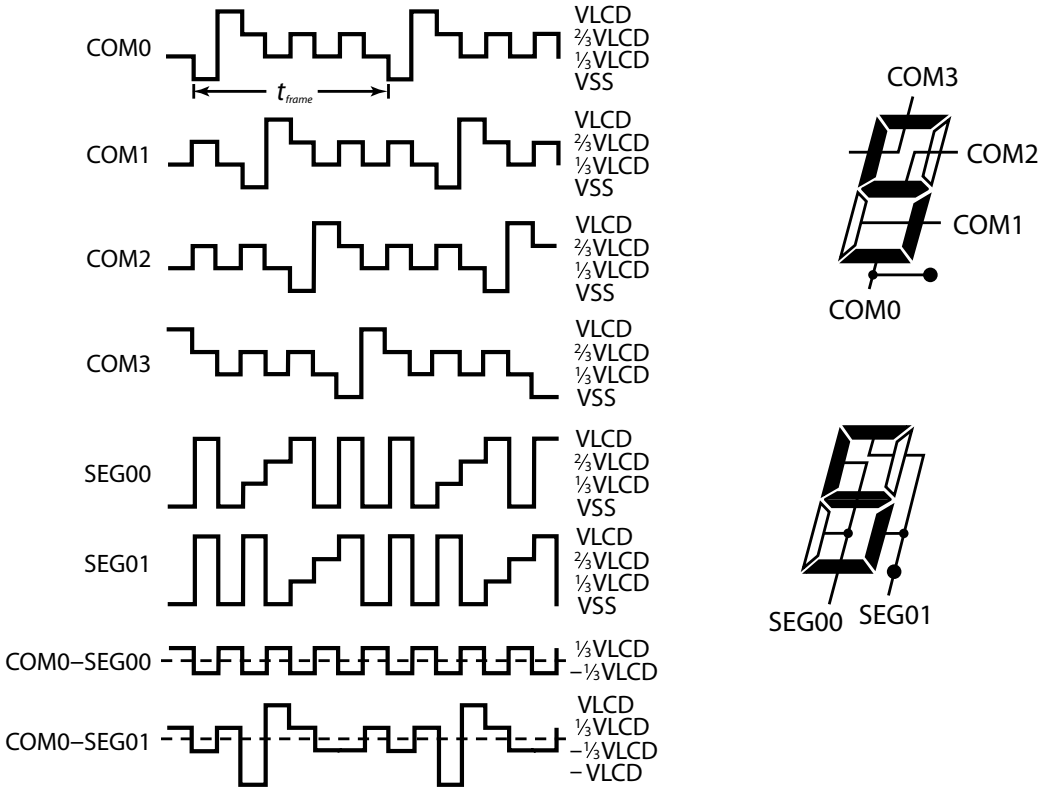

Figure 92. 1/4 Duty Mode with 1/3 Bias Type A Example Waveforms

Bit	Description (Continued)
[3] BIASGSEL	**Bias Generator Selection** 0: The normal transition bias drive is selected, nominally 360kΩ for 1/3 LCD waveform biasing and 240kΩ for 1/2 LCD waveform biasing. Normal internal charge pump output current is also selected. 1: The high transition bias drive is selected, nominally90kΩ for 1/3 LCD waveform biasing and 60 kΩ for 1/2 LCD waveform biasing. High internal charge pump output current is also selected.
[2:0] CONTRAST	**Contrast Control** CONTRAST is a function of CPEN and waveform type (LCDMODE[2]). CPEN = 0, LCDMODE[2] = 0 (type B waveform): CONTRAST sets the number of frame clock cycles that all LCD Controller outputs are driven to V_{SS}. 000: 0 dead cycles. 001: 1 dead cycles. 010: 2 dead cycles. 011: 3 dead cycles. 100: 4 dead cycles. 101: 6 dead cycles. 110: 8 for 1/2 and 1/4 duty. Reserved (not supported) for 1/3 duty. 111: Reserved. CPEN = 0, LCDMODE[2] = 1 (type A waveform): CONTRAST sets the number of frame clock cycles that all LCD Controller outputs are driven to V_{SS}. 000: 0 dead cycles. 001: 1 dead cycles. 010: 2 dead cycles. 011: 3 dead cycles. 100: 4 dead cycles. 101: 5 dead cycles. 110: 6 dead cycles. 111: 8 for 1/2 and 1/4 duty. Reserved (not supported) for 1/3 duty. CPEN = 1: CONTRAST sets t7he V_{LCD} level generated by the internal charge pump. 000: V_{LCD} = 2.50V (nominal). 001: V_{LCD} = 2.64V (nominal). 010: V_{LCD} = 2.78V (nominal). 011: V_{LCD} = 2.92V (nominal). 100: V_{LCD} = 3.06V (nominal). 101: V_{LCD} = 3.20V (nominal). 110: V_{LCD} = 3.35V (nominal). 111: V_{LCD} = 3.50V (nominal).

Chapter 27. Flash Memory

The products in the F6482 Series feature either 64KB (65536), 60KB (61440), 32KB (32768), or 16KB (16384) of nonvolatile Flash memory with read/write/erase capability. This Flash memory can be programmed and erased in-circuit by either user code or through the On-Chip Debugger.

The Flash memory array is arranged in pages with 512 bytes per page. The 512-byte page is the minimum Flash memory area that can be erased. Each page is divided into 4 rows of 128 bytes.

For program/data protection, a block of Flash memory can be protected. The size of the protected block is configured to be at the desired page boundary.

The first 2 bytes of Flash Program Memory are used as Flash option bits. For more information about their operation, see the Flash Option Bit Address Space section on page 544.

Table 276 lists the Flash memory configuration for each device in the F6482 Series; Figure 94 shows the Flash memory arrangement.

Table 276. F6482 Series Flash Memory Configurations

Part Number	Flash Size in KB (Bytes)	Flash Pages	Program Memory Addresses
Z8F6482, Z8F6481	64 (65536)	128	0000h–FFFFh
Z8F6082, Z8F6081	60 (61440)	120	0000h–EFFFh
Z8F3282, Z8F3281	32 (32768)	64	0000h–7FFFh
Z8F1682, Z8F1681	16 (16384)	32	0000h–3FFFh

28.2.1. Trim Bit Address Register

The Trim Bit Address Register, shown in Table 283, contains the target address for access to the trim option bits. Trim Bit addresses in the range 00h–1Fh map to the Information Area address range 20h–3Fh, as indicated in Table 284.

Table 283. Trim Bit Address Register (TRMADR)

Bit	7	6	5	4	3	2	1	0
Field	Reserved			TRMADR				
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	FF6h							

Bit	Description
[7:5]	**Reserved** These bits are reserved.
[4:0] TRMADR	**Trim Bit Address** 00–1F: Selects the trim option bit register accessed when TRMDR is written; see the map in Table 284.

Table 284. Trim Bit Address Map

Trim Bit Address	Information Area Address
00h	20h
01h	21h
02h	22h
03h	23h
:	:
1Fh	3Fh

munication will only work when using an external clock source. To operate in high-speed synchronous mode, simply Auto-Baud to the desired speed. The Auto-Baud generator will automatically run at the desired baud rate.

Slow bus rise times due to the pull-up resistor become a limiting factor when operating at high speeds. To compensate for slow rise times, the output driver can be configured to drive the line High. If the Transmit Drive (TXD) bit is set, the line will be driven both High and Low during transmission. The line starts being driven at the beginning of the start bit and stops being driven at the middle of the stop bit. If the Transmit Drive High (TXDH) bit is set, the line will be driven High until the input is High or until the center of the bit occurs, whichever occurs first. If both TXD and TXDH are set, the pin will be driven High for one clock period at the beginning of each 0-to-1 transition. An example of a high-speed synchronous interface is shown in Figure 101.

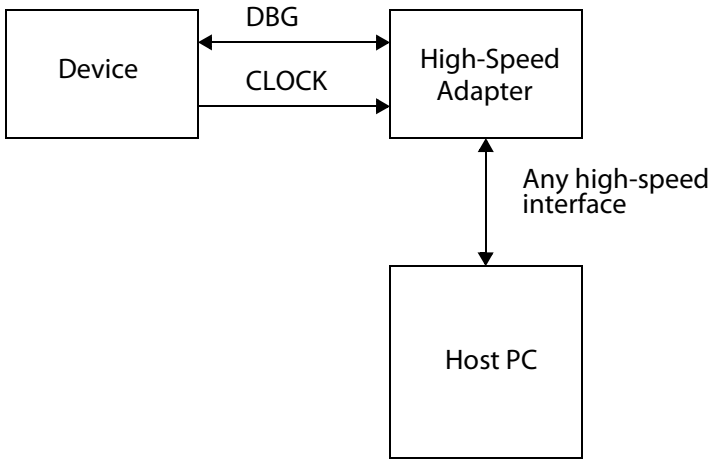

Figure 101. Synchronous Operation

30.2.6. OCD Serial Errors

The On-Chip Debugger can detect any of the following error conditions on the DBG pin:

- Serial Break (a minimum of ten continuous bits Low)
- Framing Error (the received stop bit is Low)
- Transmit Collision (OCD and host simultaneous transmission detected by the OCD)

When the OCD detects one of these errors, it aborts any command currently in progress, transmits a Serial Break 4096 system clock cycles long back to the host, and resets the Auto-Baud Detector/Generator. A Framing Error or Transmit Collision can be caused by the host sending a Serial Break to the OCD. Because of the open-drain nature of the inter-

Tables 317 through 324 contain the instructions belonging to each group and the number of operands required for each instruction. Some instructions appear in more than one table, because these instructions should be considered as a subset of more than one category. Within these tables, the source operand is identified as *src*, the destination operand is *dst*, and the condition code is *cc*.

Table 317. Arithmetic Instructions

Mnemonic	Operands	Instruction
ADC	dst, src	Add with Carry
ADCX	dst, src	Add with Carry using Extended Addressing
ADD	dst, src	Add
ADDX	dst, src	Add using Extended Addressing
CP	dst, src	Compare
CPC	dst, src	Compare with Carry
CPCX	dst, src	Compare with Carry using Extended Addressing
CPX	dst, src	Compare using Extended Addressing
DA	dst	Decimal Adjust
DEC	dst	Decrement
DECW	dst	Decrement Word
INC	dst	Increment
INCW	dst	Increment Word
MULT	dst	Multiply
SBC	dst, src	Subtract with Carry
SBCX	dst, src	Subtract with Carry using Extended Addressing
SUB	dst, src	Subtract
SUBX	dst, src	Subtract using Extended Addressing

B