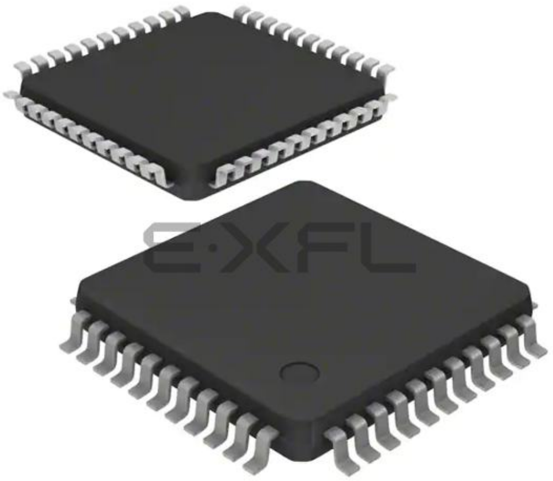
Zilog - Z8F4801AN020EC00TR Datasheet



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	Obsolete
Core Processor	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	31
Program Memory Size	48KB (48K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°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/z8f4801an020ec00tr

DMA_ADC Operation 123
Configuring DMA_ADC for Data Transfer 124
DMA Control Register Definitions 124
DMAx Control Register 124
DMAx I/O Address Register 125
DMAx Address High Nibble Register 126
DMAx Start/Current Address Low Byte Register 127
DMAx End Address Low Byte Register 128
DMA_ADC Address Register 128
DMA_ADC Control Register 130
DMA Status Register 131
Analog-to-Digital Converter 132
Overview 132
Architecture 132
Operation 133
Automatic Power-Down 133
Single-Shot Conversion 133
Continuous Conversion 134
DMA Control of the ADC 135
ADC Control Register Definitions 135
ADC Control Register 135
ADC Data High Byte Register 137
ADC Data Low Bits Register 137
Flash Memory 138
Overview 138
Operation 139
Flash Operation Timing Using the Flash Frequency Registers . . 141
Flash Code Protection Against External Access 141
Flash Code Protection Against Accidental Program and Erasure 141
Byte Programming 142
Page Erase 143
Mass Erase 143
Flash Controller Bypass 143
Flash Control Register Definitions 144
Flash Control Register 144
Flash Status Register 145
Flash Page Select Register 146
Flash Frequency High and Low Byte Registers 147

External Pin Reset

The $\overline{\text{RESET}}$ pin has a Schmitt-triggered input and an internal pull-up. Once the $\overline{\text{RESET}}$ pin is asserted, the device progresses through the Short Reset sequence. While the $\overline{\text{RESET}}$ input pin is asserted Low, the Z8F640x family device continues to be held in the Reset state. If the $\overline{\text{RESET}}$ pin is held Low beyond the Short Reset time-out, the device exits the Reset state immediately following $\overline{\text{RESET}}$ pin deassertion. Following a Short Reset initiated by the external $\overline{\text{RESET}}$ pin, the `EXT` status bit in the Watch-Dog Timer Control (WDTCTL) register is set to 1.

Stop Mode Recovery

Stop mode is entered by execution of a `STOP` instruction by the eZ8 CPU. Refer to the **Low-Power Modes** chapter for detailed Stop mode information. During Stop Mode Recovery, the Z8F640x family device is held in reset for 514 cycles of the Watch-Dog Timer oscillator followed by 16 cycles of the system clock (crystal oscillator). Stop Mode Recovery does not affect any values in the Register File, including the Stack Pointer, Register Pointer, Flags and general-purpose RAM.

The eZ8 CPU fetches the Reset vector at Program Memory addresses `0002H` and `0003H` and loads that value into the Program Counter. Program execution begins at the Reset vector address. Following Stop Mode Recovery, the `STOP` bit in the Watch-Dog Timer Control Register is set to 1. Table 9 lists the Stop Mode Recovery sources and resulting actions. The text following provides more detailed information on each of the Stop Mode Recovery sources.

Table 9. Stop Mode Recovery Sources and Resulting Action

Operating Mode	Stop Mode Recovery Source	Action
Stop mode	Watch-Dog Timer time-out when configured for Reset	Stop Mode Recovery
	Watch-Dog Timer time-out when configured for interrupt	Stop Mode Recovery followed by interrupt (if interrupts are enabled)
	Data transition on any GPIO Port pin enabled as a Stop Mode Recovery source	Stop Mode Recovery

Stop Mode Recovery Using Watch-Dog Timer Time-Out

If the Watch-Dog Timer times out during Stop mode, the Z8F640x family device undergoes a STOP Mode Recovery sequence. In the Watch-Dog Timer Control register, the `WDT` and STOP bits are set to 1. If the Watch-Dog Timer is configured to generate an interrupt upon time-out and the device is configured to respond to interrupts, the Z8F640x family device services the Watch-Dog Timer interrupt request following the normal Stop Mode Recovery sequence.

Interrupt Control Register

The Interrupt Control (IRQCTL) register (Table 37) contains the master enable bit for all interrupts.

Table 37. Interrupt Control Register (IRQCTL)

BITS	7	6	5	4	3	2	1	0
FIELD	IRQE	Reserved						
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R
ADDR	FCFH							

IRQE—Interrupt Request Enable
This bit is set to 1 by execution of an EI (Enable Interrupts) or IRET (Interrupt Return) instruction, or by a direct register write of a 1 to this bit. It is reset to 0 by executing a DI instruction, eZ8 CPU acknowledgement of an interrupt request, or Reset.
0 = Interrupts are disabled.
1 = Interrupts are enabled.

Reserved
These bits must be 0.

Capture mode
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
When the timer is disabled, the Timer Output signal is set to the value of this bit. When the timer is enabled, the Timer Output signal is complemented upon timer Reload.

Gated mode
0 = Timer counts when the Timer Input signal is High (1) and interrupts are generated on the falling edge of the Timer Input.
1 = Timer counts when the Timer Input signal is Low (0) and interrupts are generated on the rising edge of the Timer Input.

Capture/Compare mode
0 = Counting is started 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 is started on the first falling edge of the Timer Input signal. The current count is captured on subsequent falling edges of the Timer Input signal.

PRES—Prescale value.
The timer input clock is divided by 2^{PRES}, where PRES can be set from 0 to 7. The prescaler is reset each time the Timer is disabled. This insures proper clock division each time the Timer is restarted.
000 = Divide by 1
001 = Divide by 2
010 = Divide by 4
011 = Divide by 8
100 = Divide by 16
101 = Divide by 32
110 = Divide by 64
111 = Divide by 128

TMODE—Timer mode
000 = One-Shot mode
001 = Continuous mode
010 = Counter mode
011 = PWM mode
100 = Capture mode
101 = Compare mode
110 = Gated mode
111 = Capture/Compare mode

7. Write to the UART Control 0 register to:
   - Set the receive enable bit (REN) to enable the UART for data reception
   - Enable parity, if desired, and select either even or odd parity.
8. Execute an EI instruction to enable interrupts.

The UART is now configured for interrupt-driven data reception. When the UART Receiver interrupt is detected, the associated interrupt service routine (ISR) should perform the following:

9. Check the UART Status 0 register to determine the source of the interrupt - error, break, or received data.
10. If the interrupt was due to data available, read the data from the UART Receive Data register. If operating in Multiprocessor (9-bit) mode, first read the Multiprocessor Receive flag (MPRX) to determine if the data was directed to this UART before reading the data.
11. Clear the UART Receiver interrupt in the applicable Interrupt Request register.
12. Execute the IRET instruction to return from the interrupt-service routine and await more data.

Receiving Data using the Direct Memory Access Controller (DMA)

The DMA and UART can coordinate automatic data transfer from the UART Receive Data register to general-purpose Register File RAM. This reduces the eZ8 CPU processing overhead required to support UART data reception. The UART Receiver interrupt must then only notify the eZ8 CPU of error conditions. Follow these steps to configure the UART and DMA for automatic data handling:

1. Write to the DMA control registers to configure the DMA to transfer data from the UART Receive Data register to general-purpose Register File RAM.
2. Write to the UART Baud Rate High and Low Byte registers to set the desired baud rate.
3. Enable the UART pin functions by configuring the associated GPIO Port pins for alternate function operation.
4. Write to the Interrupt control registers to enable the UART Receiver interrupt and set the desired priority.
5. Write to the UART Control 1 register to:
   - Enable Multiprocessor (9-bit) mode functions, if desired.
   - Disable the UART interrupt for received data by clearing $\overline{\text{RDAIRQ}}$ to 0.

When the DMA is configured for two-byte word transfers, the DMA*x* I/O Address register must contain an even numbered address.

Table 72. DMA*x* I/O Address Register (DMA*x*IO)

BITS	7	6	5	4	3	2	1	0
FIELD	DMA_IO							
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
ADDR	FB1H, FB9H							

DMA_IO—DMA on-chip peripheral control register address
This byte sets the low byte of the on-chip peripheral control register address on Register File Page `FH` (addresses `F00H` to `FFFH`).

DMA*x* Address High Nibble Register

The DMA*x* Address High register specifies the upper four bits of address for the Start/Current and End Addresses of DMA*x*.

Table 73. DMA*x* Address High Nibble Register (DMA*x*H)

BITS	7	6	5	4	3	2	1	0
FIELD	DMA_END_H				DMA_START_H			
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
ADDR	FB2H, FHAH							

DMA_END_H—DMA*x* End Address High Nibble
These bits, used with the DMA*x* End Address Low register, form a 12-bit End Address. The full 12-bit address is given by {DMA_END_H[3:0], DMA_END[7:0]}.

DMA_START_H—DMA*x* Start/Current Address High Nibble
These bits, used with the DMA*x* Start/Current Address Low register, form a 12-bit Start/Current Address. The full 12-bit address is given by {DMA_START_H[3:0], DMA_START[7:0]}.

DMA*x* Start/Current Address Low Byte Register

The DMA*x* Start/Current Address Low register, in conjunction with the DMA*x* Address High Nibble register, forms a 12-bit Start/Current Address. Writes to this register set the Start Address for DMA operations. Each time the DMA completes a data transfer, the 12-bit Start/Current Address increments by either 1 (single-byte transfer) or 2 (two-byte word transfer). Reads from this register return the low byte of the Current Address to be used for the next DMA data transfer.

Table 74. DMA*x* Start/Current Address Low Byte Register (DMA*x*START)

BITS	7	6	5	4	3	2	1	0
FIELD	DMA_START							
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
ADDR	FB3H, FHBH							

DMA_START—DMA*x* Start/Current Address Low
These bits, with the four lower bits of the DMA*x*_H register, form the 12-bit Start/Current address. The full 12-bit address is given by {DMA_START_H[3:0], DMA_START[7:0]}.

DMA*x* End Address Low Byte Register

The DMA*x* End Address Low Byte register, in conjunction with the DMA*x*_H register, forms a 12-bit End Address.

Table 75. DMA*x* End Address Low Byte Register (DMA*x*END)

BITS	7	6	5	4	3	2	1	0
FIELD	DMA_END							
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
ADDR	FB4H, FBCH							

DMA_END—DMA*x* End Address Low
These bits, with the four upper bits of the DMA*x*_H register, form a 12-bit address. This address is the ending location of the DMA*x* transfer. The full 12-bit address is given by {DMA_END_H[3:0], DMA_END[7:0]}.

DMA_ADC Address Register

The DMA_ADC Address register points to a block of the Register File to store ADC conversion values as illustrated in Table 76. This register contains the seven most-significant bits of the 12-bit Register File addresses. The five least-significant bits are calculated from the ADC Analog Input number (5-bit base address is equal to twice the ADC Analog Input number). The 10-bit ADC conversion data is stored as two bytes with the most significant byte of the ADC data stored at the even numbered Register File address.

Flash Frequency High and Low Byte Registers

The Flash Frequency High and Low Byte registers combine to form a 16-bit value, FFREQ, to control timing for Flash program and erase operations. The 16-bit binary Flash Frequency value must contain the system clock frequency (in kHz) and is calculated using the following equation:.

$$\text{FFREQ[15:0]} = \{\text{FFREQH[7:0]},\text{FFREQL[7:0]}\} = \frac{\text{System Clock Frequency}}{1000}$$

⚠ **Caution:** Flash programming and erasure is not supported for system clock frequencies below 32KHz (32768Hz) or above 20MHz. The Flash Frequency High and Low Byte registers must be loaded with the correct value to insure proper operation of the Z8F640x family device.

Table 88. Flash Frequency High Byte Register (FFREQH)

BITS	7	6	5	4	3	2	1	0
FIELD	FFREQH							
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
ADDR	FFAH							

FFREQH—Flash Frequency High Byte
High byte of the 16-bit Flash Frequency value.

Table 89. Flash Frequency Low Byte Register (FFREQL)

BITS	7	6	5	4	3	2	1	0
FIELD	FFREQL							
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
ADDR	FFBH							

FFREQL—Flash Frequency Low Byte
Low byte of the 16-bit Flash Frequency value.

Option Bits

Overview

Option Bits allow user configuration of certain aspects of Z8F640x family device operation. The feature configuration data is stored in the Program Memory and read during Reset. The features available for control via the Option Bits are:

- Watch-Dog Timer time-out response selection–interrupt or Short Reset.
- Watch-Dog Timer 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 all or a portion of the user code in Program Memory.

Operation

Option Bit Configuration By Reset

Each time the Option Bits are programmed or erased, the Z8F640x family 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 Z8F640x family device. Option Bit control of the Z8F640x family device is established before the device exits Reset and the eZ8 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 two bytes of Program Memory at addresses `0000H` and `0001H` are reserved for the user Option Bits. The byte at Program Memory address `0000H` is used to configure user options. The byte at Program Memory address `0001H` is reserved for future use and must be left in its unprogrammed state.

Program Memory Address 0000H

Table 90. Option Bits At Program Memory Address 0000H

BITS	7	6	5	4	3	2	1	0
FIELD	WDT_RES	WDT_AO	Reserved			RP	FHSWP	FWP
RESET	U	U	U	U	U	U	U	U
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	Program Memory 0000H							
Note: U = Unchanged by Reset. R/W = Read/Write.								

WDT_RES—Watch-Dog Timer Reset
0 = Watch-Dog Timer time-out generates an interrupt request. Interrupts must be globally enabled for the eZ8 CPU to acknowledge the interrupt request.
1 = Watch-Dog Timer time-out causes a Short Reset. This setting is the default for unprogrammed (erased) Flash.

WDT_AO—Watch-Dog Timer Always On
0 = Watch-Dog Timer is automatically enabled upon application of system power. Watch-Dog Timer can not be disabled.
1 = Watch-Dog Timer is enabled upon execution of the WDT instruction. Once enabled, the Watch-Dog Timer can only be disabled by a Reset or Stop Mode Recovery. This setting is the default for unprogrammed (erased) Flash.

Reserved
These Option Bits are reserved for future use and must always be set to 1. This setting is the default for unprogrammed (erased) Flash.

RP—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 the default for unprogrammed (erased) Flash.

If the OCD receives a Serial Break (nine or more continuous bits Low) the Auto-Baud Detector/Generator resets. The Auto-Baud Detector/Generator can then be reconfigured by sending `80H`.

OCD Serial Errors

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

- Serial Break (a minimum of nine continuous bits Low)
- Framing Error (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 four character long Serial Break back to the host, and resets the Auto-Baud Detector/Generator. A Framing Error or Transmit Collision may be caused by the host sending a Serial Break to the OCD. Because of the open-drain nature of the interface, returning a Serial Break break back to the host only extends the length of the Serial Break if the host releases the Serial Break early.

The host should transmit a Serial Break on the DBG pin when first connecting to the Z8F640x family device or when recovering from an error. A Serial Break from the host resets the Auto-Baud Generator/Detector but does not reset the OCD Control register. A Serial Break leaves the Z8F640x family device in Debug mode if that is the current mode. The OCD is held in Reset until the end of the Serial Break when the DBG pin returns High. Because of the open-drain nature of the DBG pin, the host can send a Serial Break to the OCD even if the OCD is transmitting a character.

Breakpoints

Execution Breakpoints are generated using the BRK instruction (opcode 00H). When the eZ8 CPU decodes a BRK instruction, it signals the On-Chip Debugger. If Breakpoints are enabled, the OCD enters Debug mode and idles the eZ8 CPU. If Breakpoints are not enabled, the OCD ignores the BRK signal and the BRK instruction operates as an NOP.

Breakpoints in Flash Memory

The BRK instruction is opcode `00H`, which corresponds to the fully programmed state of a byte in Flash memory. To implement a Breakpoint, write `00H` to the desired address, overwriting the current instruction. To remove a Breakpoint, the corresponding page of Flash memory must be erased and reprogrammed with the original data.

Watchpoints

The On-Chip Debugger can set one Watchpoint to cause a Debug Break. The Watchpoint identifies a single Register File address. The Watchpoint can be set to break on reads and/ or writes of the selected Register File address. Additionally, the Watchpoint can be configured to break only when a specific data value is read and/or written from the specified reg-

```
DBG <-- 03H
DBG --> RuntimeCounter[15:8]
DBG --> RuntimeCounter[7:0]
```

- **Write OCD Control Register (04H)**—The Write OCD Control Register command writes the data that follows to the OCDCTL register. When the Read Protect Option Bit is enabled, the DBGMODE bit (OCDCTL[7]) can only be set to 1, it cannot be cleared to 0 and the only method of putting the Z8F640x family device back into normal operating mode is to reset the device.

```
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 Z8F640x family 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 Z8F640x family 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 zero). If the Z8F640x family 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 Flash Control Registers are allowed and all other register write data values are discarded.

```
DBG <-- 08H
DBG <-- {4’h0,Register Address[11:8]}
DBG <-- Register Address[7:0]
DBG <-- Size[7:0]
DBG <-- 1-256 data bytes
```

- **Read Register (09H)**—The Read Register command reads data from the Register File. Data can be read 1-256 bytes at a time (256 bytes can be read by setting size to

- **Write Watchpoint (20H)**—The Write Watchpoint command sets and configures the debug Watchpoint. If the Z8F640x family device is not in Debug mode or the Read Protect Option Bit is enabled, the `WPTCTL` bits are all set to zero.

```
DBG <-- 20H
DBG <-- WPTCTL[7:0]
DBG <-- WPTADDR[7:0]
DBG <-- WPTDATA[7:0]
```

- **Read Watchpoint (21H)**—The Read Watchpoint command reads the current Watchpoint registers.

```
DBG <-- 21H
DBG --> WPTCTL[7:0]
DBG --> WPTADDR[7:0]
DBG --> WPTDATA[7:0]
```

On-Chip Debugger Control Register Definitions

OCD Control Register

The OCD Control register controls the state of the On-Chip Debugger. This register enters or exits Debug mode and enables the BRK instruction. It can also reset the Z8F640x family device.

A “reset and stop” function can be achieved by writing `81H` to this register. A “reset and go” function can be achieved by writing `41H` to this register. If the Z8F640x family device is in Debug mode, a “run” function can be implemented by writing `40H` to this register.

Table 94. OCD Control Register (OCDCTL)

BITS	7	6	5	4	3	2	1	0
FIELD	DBGMODE	BRKEN	DBGACK	Reserved				RST
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R/W

DBGMODE—Debug Mode
Setting this bit to 1 causes the Z8F640x family device to enter Debug mode. When in Debug mode, the eZ8 CPU stops fetching new instructions. Clearing this bit causes the eZ8 CPU to start running again. This bit is automatically set when a BRK instruction is decoded and Breakpoints are enabled or when a Watchpoint Debug Break is detected. If the Read Protect Option Bit is enabled, this bit can only be cleared by resetting the Z8F640x family device, it cannot be written to 0.
0 = The Z8F640x family device is operating in normal mode.
1 = The Z8F640x family device is in Debug mode.

RPEN—Read Protect Option Bit Enabled
0 = The Read Protect Option Bit is disabled (1).
0 = The Read Protect Option Bit is enabled (0), disabling many OCD commands.

Reserved
These bits are always 0.

OCD Watchpoint Control Register

The OCD Watchpoint Control register is used to configure the debug Watchpoint.

Table 96. OCD Watchpoint Control/Address (WPTCTL)

BITS	7	6	5	4	3	2	1	0
FIELD	WPW	WPR	WPDM	Reserved	WPTADDR[11:8]			
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

WPW—Watchpoint Break on Write
This bit cannot be set if the Read Protect Option Bit is enabled.
0 = Watchpoint Break on Register File write is disabled.
1 = Watchpoint Break on Register File write is enabled.

WPR—Watchpoint Break on Read
This bit cannot be set if the Read Protect Option Bit is enabled.
0 = Watchpoint Break on Register File read is disabled.
1 = Watchpoint Break on Register File write is enabled.

WPDM—Watchpoint Data Match
If this bit is set, then the Watchpoint only generates a Debug Break if the data being read or written matches the specified Watchpoint data. Either the `WPR` and/or `WPW` bits must also be set for this bit to affect operation. This bit cannot be set if the Read Protect Option Bit is enabled.
0 = Watchpoint Break on read and/or write does not require a data match.
1 = Watchpoint Break on read and/or write requires a data match.

Reserved
This bit is reserved and must be 0.

RADDR[11:8]—Register address
These bits specify the upper 4 bits of the Register File address to match when generating a Watchpoint Debug Break. The full 12-bit Register File address is given by {WPTCTL3:0], WPTADDR[7:0]}.

General Purpose I/O Port Output Timing

Figure 94 and Table 108 provide timing information for GPIO Port pins.

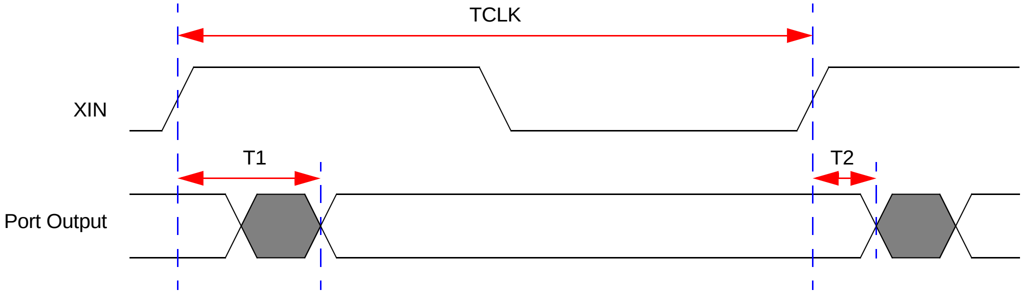


Figure 94. GPIO Port Output Timing

Table 108. GPIO Port Output Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
T_1	XIN Rise to Port Output Valid Delay	–	15
T_2	XIN Rise to Port Output Hold Time	2	–

On-Chip Debugger Timing

Figure 95 and Table 109 provide timing information for DBG pins. The timing specifications presume a rise and fall time on DBG of less than 4µs.

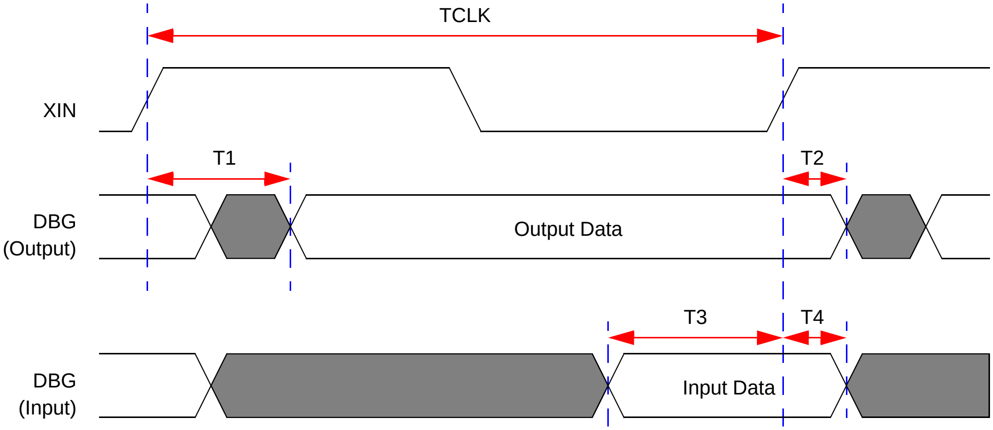


Figure 95. On-Chip Debugger Timing

Table 109. On-Chip Debugger Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
DBG			
T_1	XIN Rise to DBG Valid Delay	–	15
T_2	XIN Rise to DBG Output Hold Time	2	–
T_3	DBG to XIN Rise Input Setup Time	10	–
T_4	DBG to XIN Rise Input Hold Time	5	–
	DBG frequency		System Clock / 4

Table 126. eZ8 CPU Instruction Summary (Continued)

Assembly		Address Mode		Opcode(s)	Flags						Fetch	Instr.
Mnemonic	Symbolic Operation	dst	src	(Hex)	C	Z	S	V	D	H	Cycles	Cycles
ADD dst, src	dst ← dst + src	r	r	02	*	*	*	*	0	*	2	3
		r	Ir	03							2	4
		R	R	04							3	3
		R	IR	05							3	4
		R	IM	06							3	3
		IR	IM	07							3	4
ADDX dst, src	dst ← dst + src	ER	ER	08	*	*	*	*	0	*	4	3
		ER	IM	09							4	3
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
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
Flags Notation:	* = Value is a function of the result of the operation. - = Unaffected X = Undefined				0 = Reset to 0 1 = Set to 1							

Table 126. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode		Opcode(s) (Hex)	Flags						Fetch Cycles	Instr. Cycles
		dst	src		C	Z	S	V	D	H		
TCM dst, src	(NOT dst) AND src	r	r	62	-	*	*	0	-	-	2	3
		r	Ir	63							2	4
		R	R	64							3	3
		R	IR	65							3	4
		R	IM	66							3	3
		IR	IM	67							3	4
TCMX dst, src	(NOT dst) AND src	ER	ER	68	-	*	*	0	-	-	4	3
		ER	IM	69							4	3
TM dst, src	dst AND src	r	r	72	-	*	*	0	-	-	2	3
		r	Ir	73							2	4
		R	R	74							3	3
		R	IR	75							3	4
		R	IM	76							3	3
		IR	IM	77							3	4
TMX dst, src	dst AND src	ER	ER	78	-	*	*	0	-	-	4	3
		ER	IM	79							4	3
TRAP Vector	SP ← SP – 2 @SP ← PC SP ← SP – 1 @SP ← FLAGS PC ← @Vector		Vector	F2	-	-	-	-	-	-	2	6
WDT				5F	-	-	-	-	-	-	1	2
Flags Notation:	* = Value is a function of the result of the operation. - = Unaffected X = Undefined						0 = Reset to 0 1 = Set to 1					

Upper Nibble (Hex)	Lower Nibble (Hex)															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2																
3																
4																
5																
6																
7																
8																
9																
A			3.3 **CPC** r1,r2	3.4 **CPC** r1,Ir2	4.3 **CPC** R2,R1	4.4 **CPC** IR2,R1	4.3 **CPC** R1,IM	4.4 **CPC** IR1,IM	5.3 **CPCX** ER2,ER1	5.3 **CPCX** IM,ER1						
B																
C	3.2 **SRL** R1	3.3 **SRL** IR1														
D																
E																
F																

Figure 102. Second Opcode Map after 1FH