Crimzon® Infrared Microcontrollers

ZLP12840 OTP MCU

with Learning Amplification

Product Specification

PS024410-0108

PRELIMINARY

www.zilog.com

Warning: DO NOT USE IN LIFE SUPPORT

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer

 Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZILOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZILOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering.

Z8, Z8 Encore!, Z8 Encore! XP, Z8 Encore! MC, Crimzon, eZ80, and ZNEO are trademarks or registered trademarks of Zilog, Inc. All other product or service names are the property of their respective owners.

Zilog products are designed and manufactured under an ISO registered 9001:2000 Quality Management System. For more details, please visit www.zilog.com/quality.

ISO 9001:2000
FS 507510

Revision History

Each instance in Revision History reflects a change to this document from its previous revision. For more details, refer to the corresponding pages and appropriate links in the table below.

Date	Revision Level	Description	Page No
January 2008	10	Updated Table 61.	129
September 2007	09	Updated Features section, Figure 2, Figure 3, SMR1 Register Events, and Ordering Information section. Added Applications and Support Tools section.	1, 5, 8, 103, and 141
July 2007	08	Updated Disclaimer page and implemented style guide.	All
February 2007	07	Updated Voltage Detection section.	97
January 2006	06	Removed the trademark symbol (TM) from LXM.	All

Table of Contents

Architectural Overview

Zilog's ZLP12840 one-time-programmable (OTP) MCU is a member of the Crimzon® family of infrared microcontrollers. It provides a directly-compatible code upgrade path to other Crimzon MCUs, offers a robust learning function, and features up to 128 KB OTP read-only memory (ROM) and 1004 bytes of general-purpose random access memory (RAM). Two timers allow the generation of complex signals while performing other counting operations. A UART allows the ZLP12840 MCU to be a Slave/Master database chip. When the UART is not in use, the Baud Rate Generator can be used as a third timer. Enhanced Stop Mode Recovery (SMR) features allow the ZLP12840 MCU to awaken from STOP mode on any change of logic, and on any combination of the 12 SMR inputs. The SMR source can also be used as an interrupt source.

Many high-end remote control units offer a learning function. Simply stated, a learning function allows a replacement remote unit to learn most infrared signals from the original remote unit and regenerate the signal. However, the amplifying circuits of many learning remotes are expensive, are not tuned well. ZLP12840 MCU is the first chip dedicated to solve this problem because it offers a built-in tuned amplification circuit in a wide range of positions and battery voltages. The only external component required is a photodiode.

The ZLP12840 MCU greatly reduces system cost, yet improves learning function reliability. With all new features, the ZLP12840 MCU is excellent for infrared remote control and other MCU applications.

Features

Table 1 lists the memory, input/output (I/O), and power features of the ZLP12840 one-time-programmable microcontroller.

Table 1. ZLP12840 OTP MCU Features

Device	OTP ROM (KB)	RAM* (Bytes)	I/O Lines	Voltage Range
ZLP12840 MCU	32, 64, 96, 128	1004	24 or 16	2.0–3.6V

*General-purpose registers implemented as random access memory.

The ZLP12840 MCU supports 20 interrupt sources with 6 interrupt vectors that are listed below:

- Two from T8, T16 time-out and capture
- Three from UART Tx, UART Rx, UART BRG

- One from LVD
- 14 from SMR source P20-P27, P30-P33, P00, P07
  - Any change of logic from P20-P27, P30-P33 can generate an interrupt or SMR

Additional features include:

- IR learning amplifier
- Low power consumption—11 mW (typical)
- Three standby modes:
  - STOP—2 µA (typical)
  - HALT—0.8 mA (typical)
  - Low-Voltage Reset
- Intelligent counter/timer architecture to automate generation or reception and demodulation of complex waveform and pulsed signals:
  - One programmable 8-bit counter/timer with two capture registers and two load registers
  - One programmable 16-bit counter/timer with one 16-bit capture register pair and one 16-bit load register pair
  - Programmable input glitch filter for pulse reception
  - The UART baud rate generator can be used as another 8-bit timer when the UART is not in use
- Six priority interrupts
  - Three external/UART interrupts
  - Two assigned to counter/timers
  - One low-voltage detection interrupt
- 8-bit UART
  - R_X, T_X interrupts
  - 4800, 9600, 19200 and 38400 baud rates
  - Parity Odd/Even/None
  - Stop bits 1/2
- Low-Voltage Detection and High-Voltage Detection Flags
- Programmable Watchdog Timer/Power-On Reset circuits
- Two on-board analog comparators with independent reference voltages and programmable interrupt polarity
- One-time programmable EPROM option bits (ON/OFF)
  - Port 0 pins 0–3 pull-up transistors
  - Port 0 pins 4–7 pull-up transistors

– Port 2 pins 0–7 pull-up transistors
– EPROM Protection
– Watchdog timer enabled at Power-On Reset

Note: *All signals with an overline, "‾", are active Low. For example, B/$\overline{W}$, in which WORD is active Low, and $\overline{B}$/W, in which BYTE is active Low.*

Power connections use the conventional descriptions listed in Table 2.

Table 2. Power Connections

Connection	Circuit	Device
Power	V_{CC}	V_{DD}
Ground	GND	V_{SS}

Functional Block Diagram

Figure 1 displays the functional blocks of the ZLP12840 microcontroller.

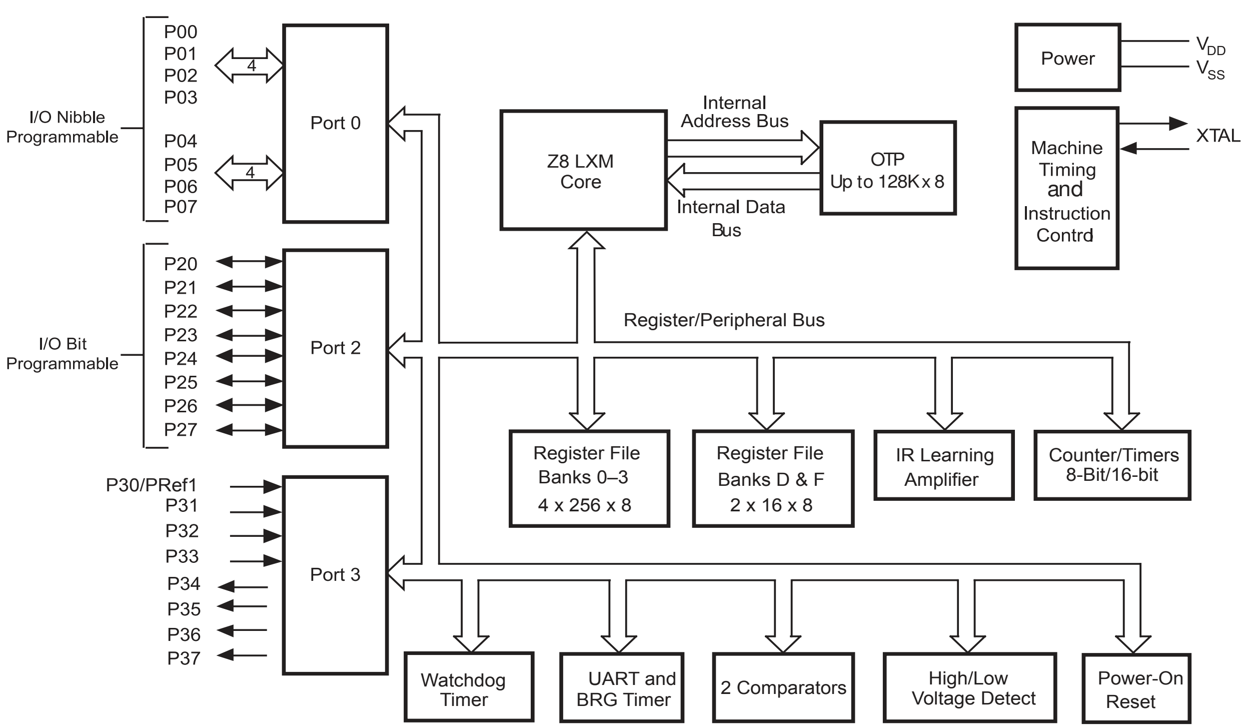


Figure 1. ZLP12840 MCU Functional Block Diagram

Pin Description

Figure 2 displays the pin configuration of the ZLP12840 device in the 20-pin PDIP, SOIC, and SSOP packages.

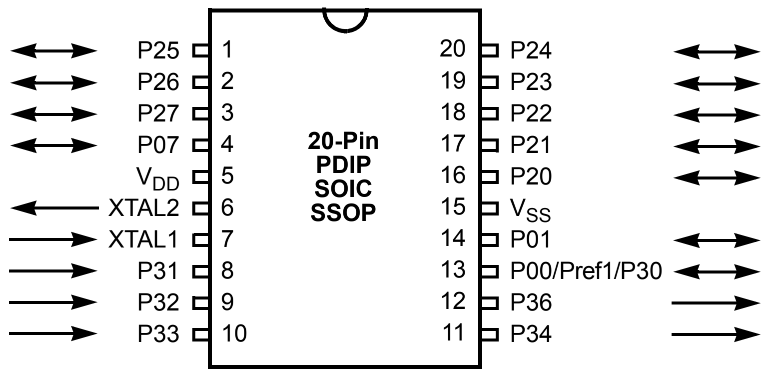


Figure 2. ZLP12840 MCU 20-Pin PDIP/SOIC/SSOP Pin Configuration

Table 3 describes the functions and signal directions of each pin within the 20-pin PDIP, SOIC, and SSOP packages sequentially by pin.

Table 3. ZLP12840 MCU 20-Pin PDIP/SOIC/SSOP Sequential Pin Identification

Pin No	Symbol	Function	Direction
1	P25	Port 2, bit 5	Input/Output
2	P26	Port 2, bit 6	Input/Output
3	P27	Port 2, bit 7	Input/Output
4	P07	Port 0, bit 7	Input/Output
5	V_{DD}	Power Supply	
6	XTAL2	Crystal oscillator	Output
7	XTAL1	Crystal oscillator	Input
8	P31	Port 3, bit 1	Input
9	P32	Port 3, bit 2	Input
10	P33	Port 3, bit 3	Input
11	P34	Port 3, bit 4	Output
12	P36	Port 3, bit 6	Output

Table 3. ZLP12840 MCU 20-Pin PDIP/SOIC/SSOP Sequential Pin Identification (Continued)

Pin No	Symbol	Function	Direction
13[1]	P00	Port 0, bit 0	Input/Output
	P30	Port 3, bit 0	Input
14	P01	Port 0, bit 1	Input/Output
15	V_{SS}	Ground	
16	P20	Port 2, bit 0	Input/Output
17	P21	Port 2, bit 1	Input/Output
18	P22	Port 2, bit 2	Input/Output
19	P23	Port 2, bit 3	Input/Output
20	P24	Port 2, bit 4	Input/Output

[1]When the Port 0 high-nibble pull-up option is enabled and the P30 input is Low, current flows through the pull-up to Ground.

Table 4 describes the functions and signal direction of each pin within the 20-pin PDIP, SOIC, and SSOP packages by function.

Table 4. ZLP12840 MCU 20-Pin PDIP/SOIC/SSOP Functional Pin Identification

Pin No	Symbol	Function	Direction
13[1]	P00	Port 0, bit 0	Input/Output
	P30	Port 3, bit 0	Input
14	P01	Port 0, bit 1	Input/Output
4	P07	Port 0, bit 7	Input/Output
16	P20	Port 2, bit 0	Input/Output
17	P21	Port 2, bit 1	Input/Output
18	P22	Port 2, bit 2	Input/Output
19	P23	Port 2, bit 3	Input/Output
20	P24	Port 2, bit 4	Input/Output
1	P25	Port 2, bit 5	Input/Output
2	P26	Port 2, bit 6	Input/Output
3	P27	Port 2, bit 7	Input/Output
8	P31	Port 3, bit 1	Input
9	P32	Port 3, bit 2	Input
10	P33	Port 3, bit 3	Input
11	P34	Port 3, bit 4	Output
12	P36	Port 3, bit 6	Output
5	V_{DD}	Power Supply	
15	V_{SS}	Ground	
7	XTAL1	Crystal oscillator	Input
6	XTAL2	Crystal oscillator	Output

[1]When the Port 0 high-nibble pull-up option is enabled and the P30 input is Low, current flows through the pull-up to Ground.

Figure 3 displays the pin configuration of the ZLP12840 device in the 28-pin PDIP, SOIC, and SSOP packages.

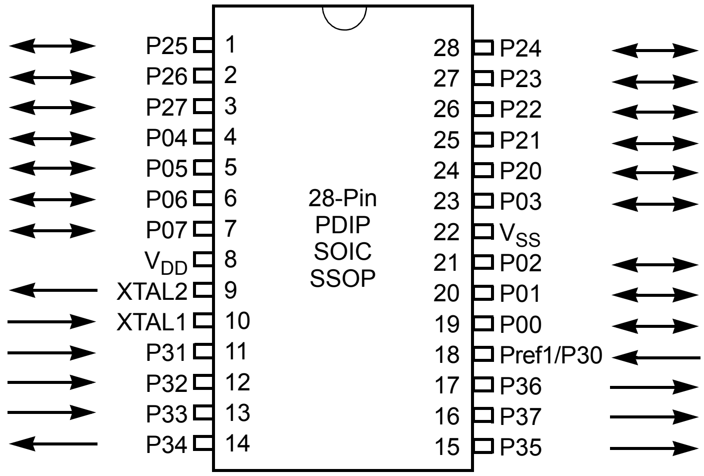


Figure 3. ZLP12840 MCU 28-Pin PDIP/SOIC/SSOP Pin Configuration

Table 5 describes the functions and signal directions of each pin within the 28-pin PDIP, SOIC, and SSOP packages sequentially by pin.

Table 5. ZLP12840 MCU 28-Pin PDIP/SOIC/SSOP Sequential Pin Identification

Pin	Symbol	Function	Direction
1	P25	Port 2, bit 5	Input/Output
2	P26	Port 2, bit 6	Input/Output
3	P27	Port 2, bit 7	Input/Output
4	P04	Port 0, bit 4	Input/Output
5	P05	Port 0, bit 5	Input/Output
6	P06	Port 0, bit 6	Input/Output
7	P07	Port 0, bit 7	Input/Output
8	V_{DD}	Power supply	
9	XTAL2	Crystal oscillator	Output
10	XTAL1	Crystal oscillator	Input
11	P31	Port 3, bit 1	Input
12	P32	Port 3, bit 2	Input
13	P33	Port 3, bit 3	Input
14	P34	Port 3, bit 4	Output
15	P35	Port 3, bit 5	Output
16	P37	Port 3, bit 7	Output
17	P36	Port 3, bit 6	Output
18	P30	Port 3, bit 0; connect to V_{CC} if not used	Input
19	P00	Port 0, bit 0	Input/Output
20	P01	Port 0, bit 1	Input/Output
21	P02	Port 0, bit 2	Input/Output
22	V_{SS}	Ground	
23	P03	Port 0, bit 3	Input/Output
24	P20	Port 2, bit 0	Input/Output
25	P21	Port 2, bit 1	Input/Output
26	P22	Port 2, bit 2	Input/Output
27	P23	Port 2, bit 3	Input/Output
28	P24	Port 2, bit 4	Input/Output

Table 6 describes the functions and signal directions of each pin within the 28-pin PDIP, SOIC, and SSOP packages by function.

Table 6. ZLP12840 MCU 28-Pin PDIP/SOIC/SSOP Functional Pin Identification

Pin	Symbol	Function	Direction
19	P00	Port 0, bit 0	Input/Output
20	P01	Port 0, bit 1	Input/Output
21	P02	Port 0, bit 2	Input/Output
23	P03	Port 0, bit 3	Input/Output
4	P04	Port 0, bit 4	Input/Output
5	P05	Port 0, bit 5	Input/Output
6	P06	Port 0, bit 6	Input/Output
7	P07	Port 0, bit 7	Input/Output
24	P20	Port 2, bit 0	Input/Output
25	P21	Port 2, bit 1	Input/Output
26	P22	Port 2, bit 2	Input/Output
27	P23	Port 2, bit 3	Input/Output
28	P24	Port 2, bit 4	Input/Output
1	P25	Port 2, bit 5	Input/Output
2	P26	Port 2, bit 6	Input/Output
3	P27	Port 2, bit 7	Input/Output
18	P30	Port 3, bit 0; connect to V_{CC} if not used	Input
11	P31	Port 3, bit 1	Input
12	P32	Port 3, bit 2	Input
13	P33	Port 3, bit 3	Input
14	P34	Port 3, bit 4	Output
15	P35	Port 3, bit 5	Output
17	P36	Port 3, bit 6	Output
16	P37	Port 3, bit 7	Output
8	V_{DD}	Power supply	
22	V_{SS}	Ground	
10	XTAL1	Crystal oscillator	Input
9	XTAL2	Crystal oscillator	Output

I/O Port Pin Functions

The ZLP12840 MCU features three 8-bit ports, which are described below.

- Port 0 is nibble-programmable as either input or output
- Port 2 is bit-programmable as either input or output
- Port 3 features four inputs on the lower nibble and four outputs on the upper nibble

Note: *Port 0 and 2 internal pull-ups are disabled on any pin or group of pins when programmed into output mode.*

Caution: *The CMOS input buffer for each port 0 or 2 pin is always connected to the pin, even when the pin is configured as an output. If the pin is configured as an open-drain output and no external signal is applied, a High output state can cause the CMOS input buffer to float. This might lead to excessive leakage current of more than 100 µA. To prevent this leakage, connect the pin to an external signal with a defined logic level or ensure its output state is Low, especially during STOP mode.*

Port 0, 1, and 2 have both input and output capability. The input logic is always present no matter whether the port is configured as input or output. When doing a READ instruction, the MCU reads the actual value at the input logic but not from the output buffer. In addition, the instructions of OR, AND, and XOR have the Read-Modify-Write sequence. The MCU first reads the port, and then modifies the value and load back to the port.

Precaution must be taken if the port is configured as open-drain output or if the port is driving any circuit that makes the voltage different from the desired output logic. For example, pins P00–P07 are not connected to anything else. If it is configured as open-drain output with output logic as ONE, it is a floating port and reads back as ZERO. The following instruction sets P00-P07 all Low.

```
AND P0,#%F0
```

Table 7 summarizes the registers used to control I/O ports. Some port pin functions can also be affected by control registers for other peripheral functions.

Table 7. I/O Port Control Registers

Address (Hex)						
12-Bit	Bank	8-Bit	Register Description	Mnemonic	Reset	Page No
000	0–3	00	Port 0	P0	XXh	21
002	0–3	02	Port 2	P2	XXh	23
003	0–3	03	Port 3	P3	0Xh	25
0F6	All	F6	Port 2 Mode Register	P2M	FFh	22
0F7	All	F7	Port 3 Mode Register	P3M	XXXX_X000b	24
0F8	All	F8	Port 0 Mode Register	P01M	X1XX_XXX1b	20
F00	F	00	Port Configuration Register	PCON	XXXX_X1X0b	20

Port 0

Port 0 is an 8-bit, bidirectional, CMOS-compatible port. Its eight I/O lines are configured under software control to create a nibble I/O port. The output drivers are push/pull or open-drain, controlled by bit 2 of the PCON register.

If one or both nibbles are required for I/O operation, they must be configured by writing to the Port 0 Mode Register (P01M). After a hardware reset or a Stop Mode Recovery, Port 0 is configured as an input port.

Port 0, bit 7 is used as the transmit output of the UART when UART Tx is enabled.The I/O function of Port 0, bit 7 is overridden by the UART serial output (TxD) when UART Tx is enabled (UCTL[7] = 1). The pin must be configured as an output for TxD data to reach the pin (P0M[6] = 0).

An optional pull-up transistor is available as an OTP option on all Port 0 bits with nibble select. For information on configuration, see Figure 4 on page 13.

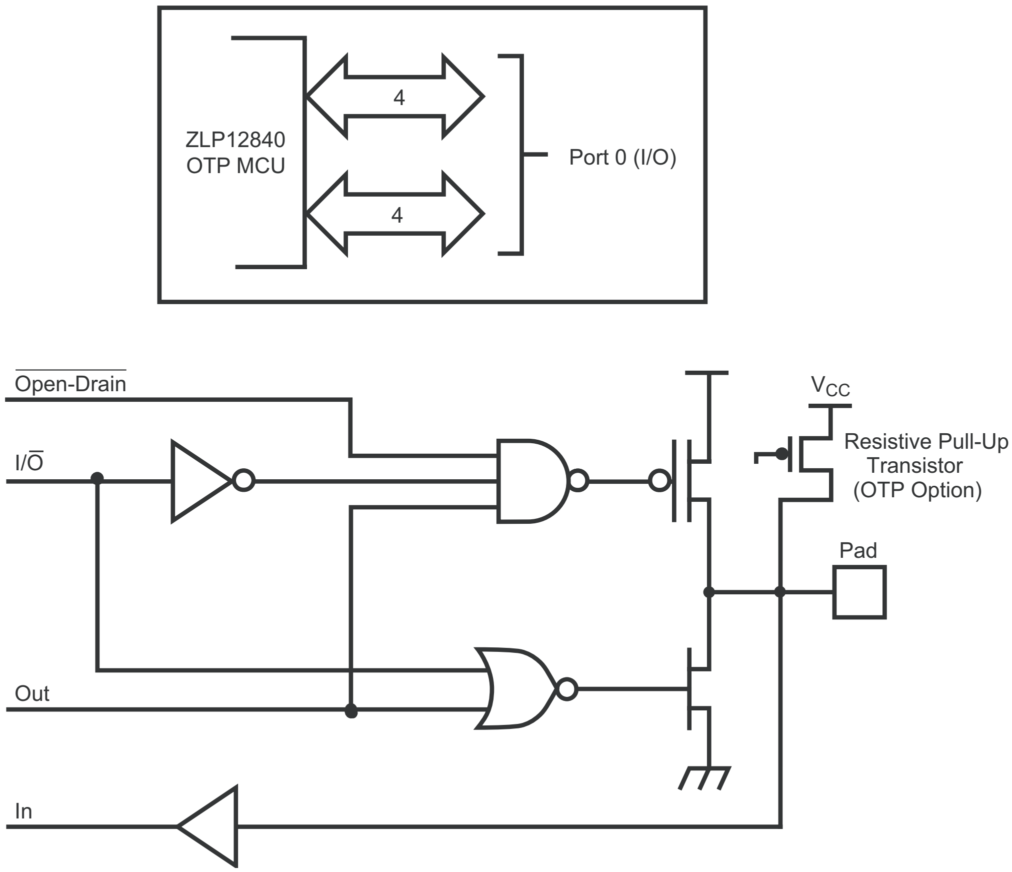


Figure 4. Port 0 Configuration

Port 2

Port 2 is an 8-bit, bidirectional, CMOS-compatible I/O port. Its eight I/O lines can be independently configured under software control as inputs or outputs. Port 2 is always available for I/O operation. An EPROM option bit is available to connect eight pull-up transistors on this port. Bits programmed as outputs are globally programmed as either push/pull or open-drain. The Power-On Reset function resets with the eight bits of Port 2 [P27:20] configured as inputs.

Port 2 also has an 8-bit input OR and AND gate and edge detection circuitry, which can be used to wake up the part. P20 can be programmed to access the edge-detection circuitry in DEMODULATION mode. For information on configuration, see Figure 5 on page 14.

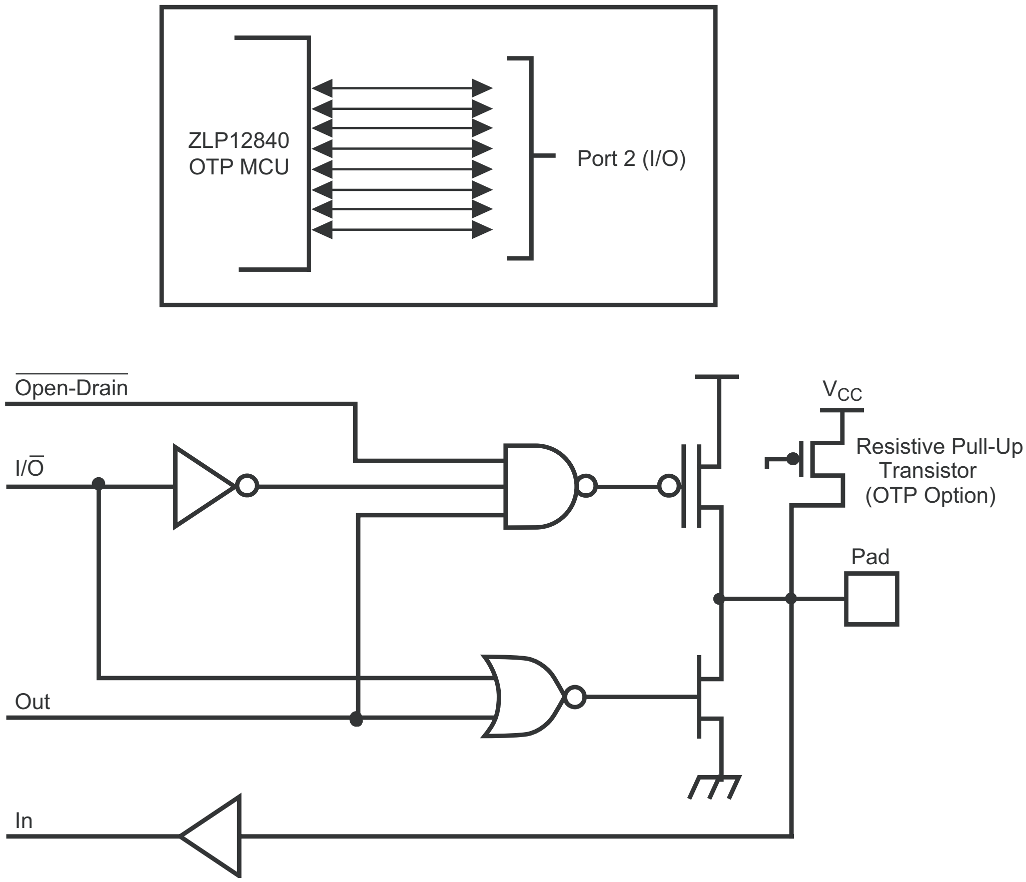


Figure 5. Port 2 Configuration

Port 3

Port 3 is a 8-bit, CMOS-compatible fixed I/O port. Port 3 consists of four fixed inputs (P33:P30) and four fixed outputs (P37:P34). P30, P31, P32, and P33 are standard CMOS inputs, and can be configured under software control as interrupts, as receive data input to the UART block, as input to comparator circuits, or as input to the IR learning AMP. P34, P35, P36, and P37 are push/pull outputs, and can be configured as outputs from the counter/timers. For information on configuration, see Figure 6 on page 15.

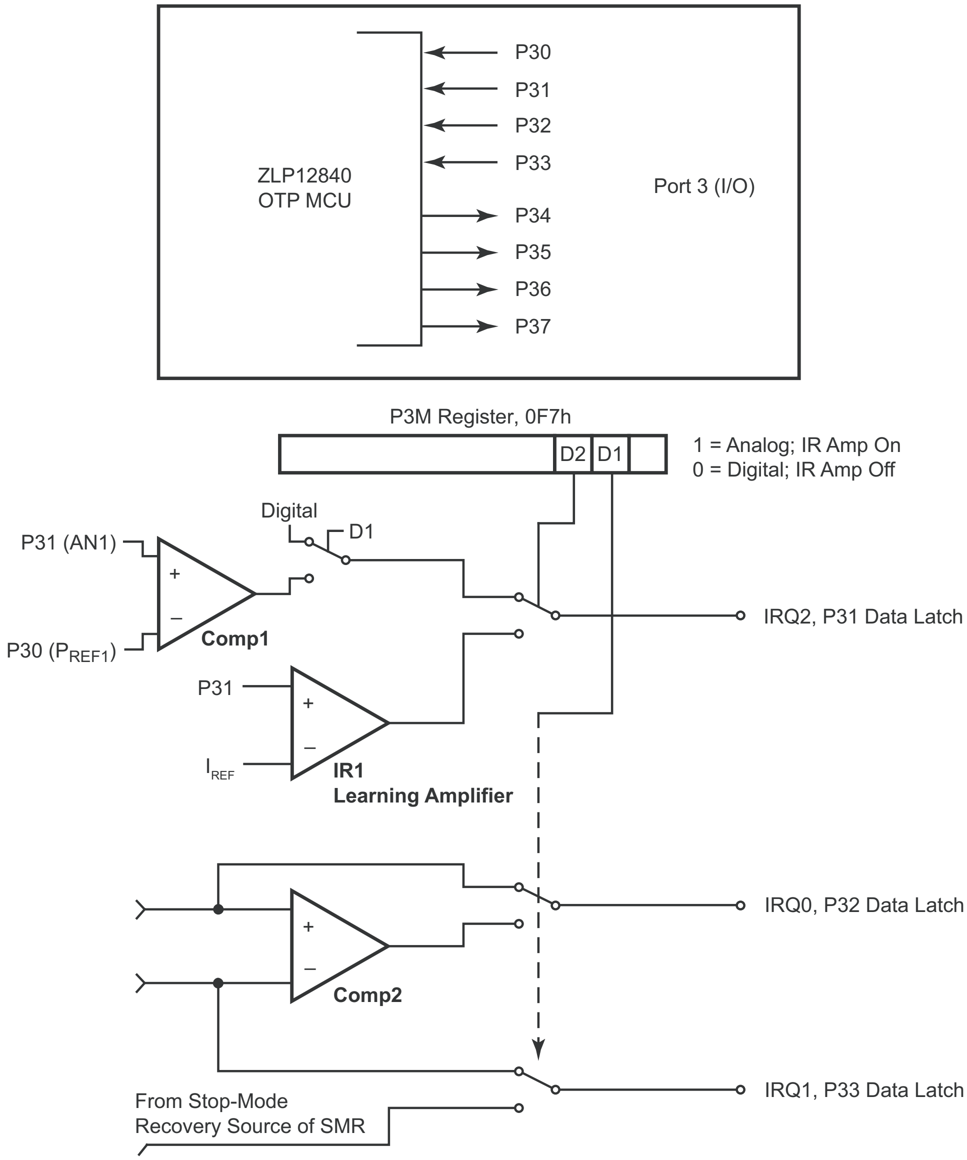


Figure 6. Port 3 Configuration

P31 can be used as an interrupt, analog comparator input, infrared learning amplifier input, normal digital input pin and as a Stop Mode Recovery source. When bit 2 of the Port 3 Mode Register (P3M) is set, P31 is used as the infrared learning amplifier, IR1. The reference source for IR1 is GND. The infrared learning amplifier is disabled during STOP mode. When bit 1 of P3M is set, the part is in ANALOG mode and the analog comparator, COMP1 is used. The reference voltage for COMP1 is P30 (P_{REF1}). When in ANALOG mode, P30 cannot be read as a digital input when the CPU reads bit 0 of the Port 3

Register; such reads always return a value of 1. Also, when in ANALOG mode, P31 cannot be used as a Stop Mode Recovery source because in STOP mode, the comparator is disabled, and its output will not toggle. The programming of Bit 2 of the P3M register takes precedence over the programming of Bit 1 in determining the function of P31. If both bits are set, P31 functions as an IR learning amplifier instead of an analog comparator. The output of the function selected for P31 can be used as a source for IRQ2 interrupt assertion (see Figure 6 on page 15). The IRQ2 interrupt can be configured to be based upon detecting a rising, falling, or edge-triggered input change using bit 6 and bit 7 of the IRQ register. The P31 output stage signal also goes to the Counter/Timer edge detection circuitry similar to P20.

P32 can be used as an interrupt, analog comparator, UART receiver, normal digital input and as a Stop Mode Recovery source. When bit 6 of UCTL is set, P32 functions as a receive input for the UART. When bit 1 of the P3M Register is set, thereby placing the part into ANALOG mode, P32 functions as an analog comparator, Comp2. The reference voltage for Comp2 is P33 ($P_{REF}2$). P32 can be used as a rising, falling or edge-triggered interrupt, IRQ0, using IRQ register bits 6 and 7. If UART receiver interrupts are not enabled, the UART receive interrupt is used as the source of interrupts for IRQ0 instead of P32. When in ANALOG mode P32 cannot be used as a Stop Mode Recovery source because the comparators are turned OFF in STOP mode.

When in ANALOG mode, P33 cannot be read through bit 3 of the Port 3 Register as a digital input by the CPU. In this case, a read of bit 3 of the Port 3 Register indicates whether a Stop Mode Recovery condition exists. Reading a value of 0 indicates that a Stop Mode Recovery condition does exist; if the ZLP12840 MCU is presently in STOP mode, it will exit STOP mode. Reading a value of 1 indicates that no condition exists to remove the ZLP12840 from STOP mode. Additionally, when in ANALOG mode, P33 cannot be used as an interrupt source. Instead, the existence of a Stop Mode Recovery condition can generate an interrupt, if enabled. P33 can be used as a falling-edge interrupt, IRQ1, when not in ANALOG mode. IRQ1 is also used as the UART T_X interrupt and the UART BRG interrupt. Only one source is active at a time. If bits 7 and 5 of UCTL are set to 1, IRQ1 will transmit an interrupt when the Transmit Shift Register is empty. If bits 0 and 5 of UCTL are set to 1 and bit 6 of UCTL is cleared to 0, the BRG interrupts will activate IRQ1.

Note: *Comparators and the IR amplifier are powered down by entering STOP mode. For P30:P33 to be used as a Stop Mode Recovery source during STOP mode, these inputs must be placed into DIGITAL mode. When in ANALOG mode, do not configure any Port 3 input as a Stop Mode Recovery source. The configuration of these inputs must be re-initialized after Stop Mode Recovery or Power-On Reset.*

Table 8. Summary of Port 3 Pin Functions

Pin	I/O	Counter/Timers	Comparator	Interrupt	IRAMP	UART
P30	IN		REF1			
P31	IN	IN	AN1	IRQ2	IR1	
P32	IN		AN2	IRQ0		UART Rx
P33	IN		REF2	IRQ1		
P34	OUT	T8	AO1		IROUT	
P35	OUT	T16				
P36	OUT	T8/T16				
P37	OUT		AO2			

Port 3 also provides output for each of the counter/timers and the AND/OR Logic (see Figure 7). Control is performed by programming CTR1 bits 5 and 4, CTR0 bit 0, and CTR2 bit 0.

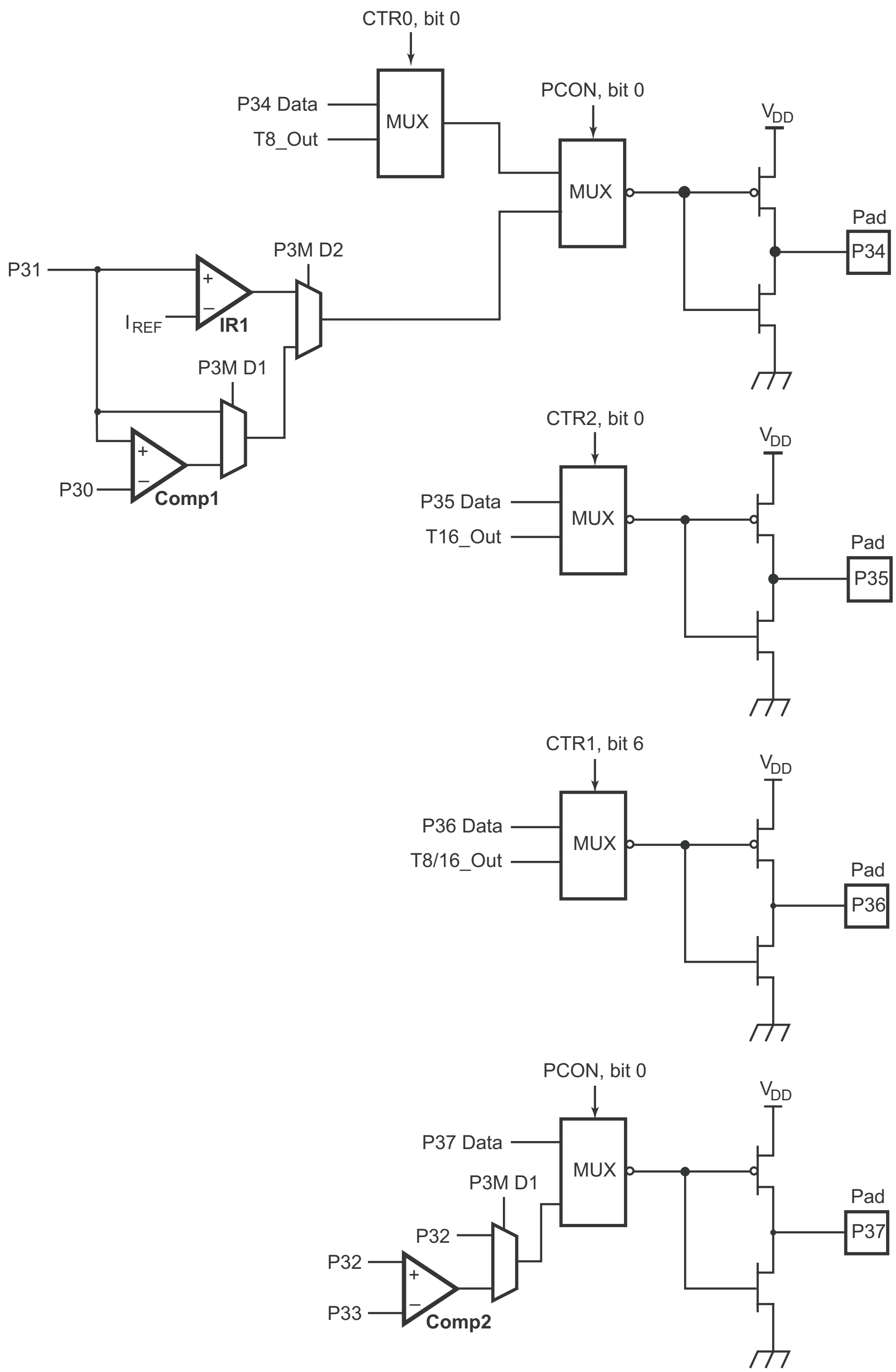


Figure 7. Port 3 Counter/Timer Output Configuration

Comparator Inputs

In ANALOG mode, P31 and P32 have a comparator front end. The comparator reference is supplied by P33 and P_{REF1}. In this mode, the P33 internal data latch and its corresponding IRQ1 are diverted to the Stop Mode Recovery sources (excluding P31, P32, and P33) as displayed in Figure 6 on page 15. In DIGITAL mode, P33 is used as bit 3 of the Port 3 input register, which then generates IRQ1.

Note: *Comparators are powered down by entering STOP mode. For P30:P33 to be used as a Stop Mode Recovery source, these inputs must be placed into DIGITAL mode.*

Comparator Outputs

The comparators can be programmed to be output on P34 and P37 by setting bit 0 of the PCON Register.

Port Configuration Register (PCON)

The Port Configuration (PCON) register (Table 9), configures the Port 0 output mode and the comparator output on Port 3. The PCON register is located in expanded register Bank F, address 00h.

Table 9. Port Configuration Register (PCON)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved					Port 0 Output Mode	Reserved	Comp./IR Amp. Output Port 3
Reset	X	X	X	X	X	1	X	0
R/W	—					W	—	W
Address	Bank F: 00h; Linear: F00h							

Bit Position	Value	Description
[7:3]	—	Reserved—Writes have no effect; reads 11111b.
[2]		Port 0 Output Mode—Controls the output mode of port 0. Write only; reads return 1.
	0	Open-drain
	1	Push/pull
[1]	—	Reserved—Writes have no effect; reads 1.
[0]		Comparator or IR Amplifier Output Port 3—Select digital outputs or comparator and IR amplifier outputs on P34 and P37. Write only; reads return 1.
	0	P34 and P37 outputs are digital.
	1	P34 is Comparator 1 or IR Amplifier output, P37 is Comparator 2 output.

Note: *This register is not reset after a SMR.*

Port 0 Mode Register

The Port 0 Mode Register determines the I/O direction of Port 0. The Port 0 direction is nibble-programmable. Bit 6 controls the upper nibble of Port 0, bits [7:3]. Bit 0 controls the lower nibble of Port 0, bits [3:0] (Table 10).

Table 10. Port 0 Mode Register (P01M)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	P07:P04 Mode	Reserved					P03:P00 Mode
Reset	X	1	X	X	X	X	X	1
R/W	—	W	—	—	—	—	—	W
Address	Bank Independent: F8h; Linear: 0F8h							

Bit Position	Value	Description
7	0	Reserved—Writes have no effect. Reads 1b.
[6]		P07:P04 Mode
	0	Output.
	1	Input.
[5:1]	—	Reserved—Writes have no effect. Reads 11111b.
[0]		P00:P03 Mode
	0	Output.
	1	Input.

Note: *Only P00, P01, and P07 are available on ZLP12840 MCU 20-pin configurations.*

Port 0 Register

The Port 0 Register allows read and write access to the Port 0 pins (Table 11).

Table 11. Port 0 Register (P0)

Bit	7	6	5	4	3	2	1	0
Field	P07	P06	P05	P04	P03	P02	P01	P00
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	Bank 0–3: 00h; Linear: 000h							

Bit Position	R/W	Description
[7]		Port 0 Pin 7—Available for I/O if UART Tx is disabled.
	Read	(Pin configured as input or output in P01M register).
	0	Pin level is Low.
	1	Pin level is High.
	Write	(Pin configured as output in P01M register, UCTL[7]=0).
	0	Assert pin Low.
	1	Assert pin High if configured as push-pull; make pin high-impedance if it is open-drain.
[6:0]		Port 0 Pins 6–0—Each bit provides access to the corresponding Port 0 pin.
	Read	(Pin configured as input or output in P01M register).
	1	Pin level is Low.
	Write	Pin level is High.
	0	(Pin configured as output in P01M register).
	1	Assert pin Low.
		Assert pin High if configured as push-pull; make pin high-impedance if it is open-drain.

Note: *Only P00, P01, and P07 are available on ZLP12840 MCU 20-pin configurations.*

Port 2 Mode Register

The Port 2 Mode Register determines the I/O direction of each bit on Port 2. Bit 0 of the Port 3 Mode Register determines whether the output drive is push/pull or open-drain (Table 12).

Table 12. Port 2 Mode Register (P2M)

Bit	7	6	5	4	3	2	1	0
Field	P27 I/O Definition	P26 I/O Definition	P25 I/O Definition	P24 I/O Definition	P23 I/O Definition	P22 I/O Definition	P21 I/O Definition	P20 I/O Definition
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Bank Independent: F6h; Linear: 0F6h							

Bit Position	Value	Description
[7]	0 1	Defines P27 as output. Defines P27 as input.
[6]	0 1	Defines P26 as output. Defines P26 as input.
[5]	0 1	Defines P25 as output. Defines P25 as input.
[4]	0 1	Defines P24 as output. Defines P24 as input.
[3]	0 1	Defines P23 as output. Defines P23 as input.
[2]	0 1	Defines P22 as output. Defines P22 as input.
[1]	0 1	Defines P21 as output. Defines P21 as input.
[0]	0 1	Defines P20 as output. Defines P20 as input.

Note: *This register is not reset after a SMR.*

Port 2 Register

The Port 2 Register allows read and write access to the Port 2 pins (Table 13).

Table 13. Port 2 Register (P2)

Bit	7	6	5	4	3	2	1	0
Field	P27	P26	P25	P24	P23	P22	P21	P20
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	Bank 0–3: 02h; Linear: 002h							

Bit Position	Value	Description
[7:0]		Port 2 Pins 7–0—Each bit provides access to the corresponding Port 2 pin.
	Read	(Pin configured as input or output in P2M register).
	0	Pin level is Low.
	1	Pin level is High.
	Write	(Pin configured as output in P2M register).
	0	Assert pin Low.
	1	Assert pin High if configured as push-pull; make pin high-impedance if it is open-drain.

Port 3 Mode Register

The Port 3 Mode Register is used primarily to configure the functionality of the Port 3 inputs. When bit 2 is set, the IR Learning Amplifier is used instead of the COMP1 comparator, regardless of the value of bit 1 (Table 14).

Table 14. Port 3 Mode Register (P3M)

Bit	7	6	5	4	3	2	1	0
Field	Reserved					IR Learning Amplifier	DIGITAL/ANALOG Mode	Port 2 Open-Drain
Reset	X	X	X	X	X	0	0	0
R/W	—	—	—	—	—	W	W	W
Address	Bank Independent: F7h; Linear 0F7h							

Bit Position	R/W	Value	Description
[7:3]	—	—	Reserved—Writes have no effect. Reads return 11111b.
[2]	W	0 1	IR Learning Amplifier disabled. IR Learning Amplifier enabled with P31 configured as amplifier input.
[1]	W	 0 1	DIGITAL/ANALOG Mode P30, P31, P32, P33 are digital inputs. P30, P32, and P33 are comparator inputs. If P3M[2]=0, P31 is also a comparator input. If P3M[2]=1, P31 is the IR amplifier input.
[0]	W	0 1	Port 2 open-drain. Port 2 push/pull.

Note: *This register is not reset after a SMR.*

Port 3 Register

The Port 3 Register allows read access to port pins P33 through P30 and write access to the port pins P37 through P34 (Table 15).

Table 15. Port 3 Register (P3)

Bit	7	6	5	4	3	2	1	0
Field	P37	P36	P35	P34	P33	P32	P31	P30
Reset	0	0	0	0	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Banks 0–3: 03h; Linear: 003h							

Bit Position	Value	Description
[7]	Write 0 1	Port 3, pin 7 Output—Writes to this bit do not affect the pin state if write-only register bit PCON[0] has been written with a 1, which configures P37 as the Comparator 1 or IR Amplifier output. P37 asserted Low if PCON[0]=0. P37 asserted High if PCON[0]=0. A read returns the last value written to this bit.
[6]	Write 0 1	Port 3, pin 6 Output—Writes to this bit do not affect the pin state if register bits CTR1[7:6]=01, which configures P36 as the Timer 8 and Timer 16 combined logic output. P36 asserted Low. P36 asserted High. A read returns the last value written to this bit.
[5]	Write 0 1	Port 3, pin 5 Output—Writes to this bit do not affect the pin state if register bit CTR2[0]=1, which configures P35 as the Timer 16 output. P35 asserted Low. P35 asserted High. A read returns the last value written to this bit.
[4]	Write 0 1	Port 3, pin 4 Output—Writes to this bit do not affect the pin state if write only register bit PCON[0]=1, which configures P34 as Comparator 2 output, or register bit CTR0[0]=1, which configures P34 as Timer 8 output. P34 asserted Low. P34 asserted High. A read returns the last value written to this bit.

Bit Position	Value	Description
[3]	Read	Port 3, pin 3 Input—Writing this bit has no effect.
		If P3M[1]=0:
	0	P33 is Low.
	1	P33 is High.
		If P3M[1]=1 or SMR4[4]=1:
	0	SMR condition exists.
	1	SMR condition does not exist.
[2]	Read	Port 3, pin 2 Input—Writing this bit has no effect.
		If P3M[1]=0:
	0	P32 input is Low.
	1	P32 input is High.
		If P3M[1]=1:
	0	Comparator 2 output is Low.
	1	Comparator 2 output is High.
[1]	Read	Port 3, pin 1 Input—Writing this bit has no effect.
		If P3M[2:1]=00:
	0	P31 input is Low.
	1	P31 input is High.
		If P3M[2:1]=01:
	0	Comparator 1 output is Low.
	1	Comparator 1 output is High.
		If P3M[2:1]=10 or 11:
	0	IR amplifier output is Low.
	1	IR amplifier output is High.
[0]	Read	Port 3, pin 0 Input—Writing this bit has no effect.
		If P3M[1]=00:
	0	P30 input is Low.
	1	P30 input is High.
		If P3M[1]=1:
	1	Reads as 1.

Note: *This register is not reset after a SMR.*

Memory and Registers

The Z8 LXM CPU used in the ZLP12840 family of devices incorporates special features to extend the available memory space while maintaining the benefits of a Z8® CPU core in consumer and battery-operated applications.

OTP Program/Constant Memory

The ZLP12840 family of devices can address up to 128 KB of one-time programmable (OTP) memory, used for object code (program instructions and immediate data) and constant data (ROM tables and data constants). The amount of OTP implemented depends on the specific device. The OTP memory space is organized in 64 KB pages with the following characteristics.

Page 0 can contain up to 64 KB of program instructions and constant data. The first 12 bytes of Page 0 are reserved for the six available 16-bit interrupt request (IRQ) vectors. Upon reset, program execution begins at address 000Ch in Page 0. Execution rolls over to the bottom of Page 0 if the program counter exceeds the highest Page 0 address (FFFFh).

Page 1, if implemented, can contain up to 64 KB of data constants and tables only. Page 1 cannot contain program instructions or immediate data. Constant data in either page can be accessed only by the Load Constant (LDC and LDCI) instructions. LDC and LDCI use 16-bit addresses to access OTP memory.

For example, if a ZLP12840 family device contains 96 KB of OTP memory, only the first 64 KB (Page 0) can contain object code; the remaining 32 KB (in Page 1) is available for constant data. For a ZLP12840 family device with 64 KB or less of total OTP memory, all OTP memory is available for object code or constant data.

The page accessed by LDC or LDCI depends on the value of Program Memory Page Register bit 0 (PMPR[0]). Page 0 is accessed if PMPR[0]=0; Page 1 is accessed if PMPR[0]=1. PMPR[7] enables the page toggle feature. For example, if PMPR[0]=0, PMPR[7]=1, and a Load Constant and Increment (LDCI) instruction address increments past FFFFh, the state of PMPR[0] is toggled from 0 to 1, and the next LDCI instruction addresses 0000h on Page 1. Figure 8 on page 28 displays the Program/Constant memory map for a 128 KB device.

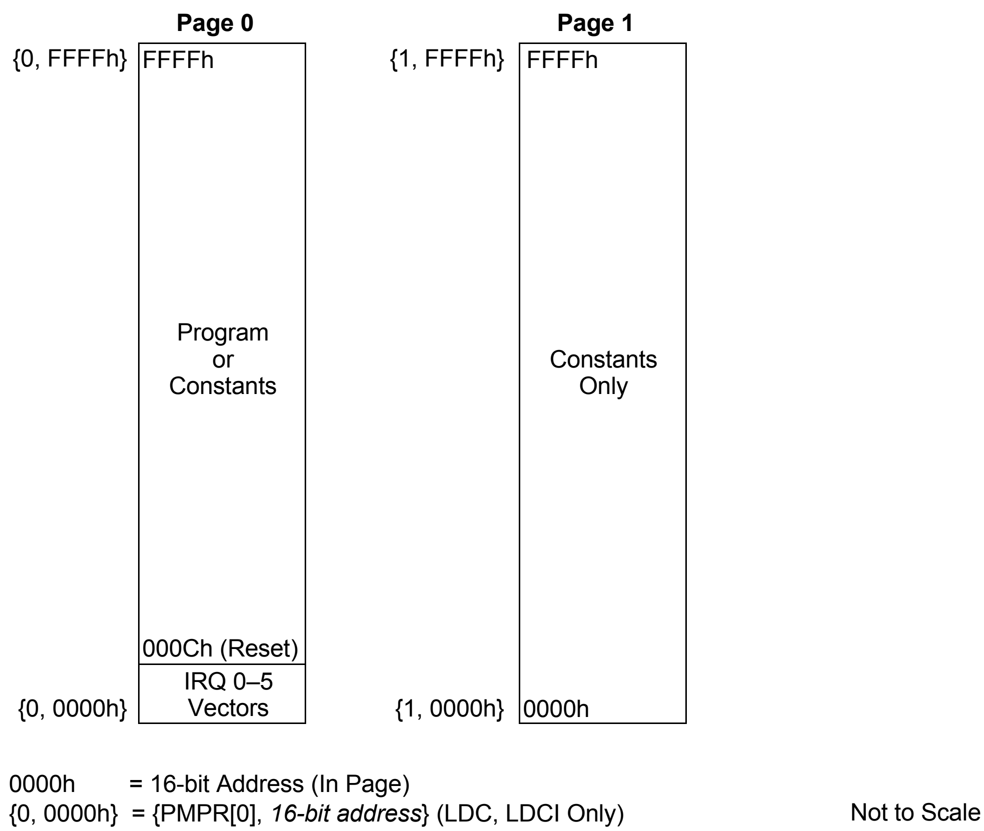


Figure 8. Program/Constant Memory Map (128 KB Device)

Register File

This device features 1056 bytes of register file space, organized in 256 byte banks. Bank 0 contains 237 bytes of RAM addressed as general-purpose registers, 4 port addresses (of which one is reserved), and 16 control register addresses. Banks 1, 2, and 3 each contain 256 general-purpose register bytes. Banks D and F each contain 16 addresses for control registers. All other banks are reserved and must not be selected.

The current bank is selected for 8-bit direct or indirect addressing by writing Register Pointer bits RP[3:0]. In the current bank, a 16-byte working register group (addressed as R0–R15) is selected by writing RP[7:4]. A working register operand requires only 4 bits of program memory. There are 16 working register groups per bank. See Figure 9 on page 30 and Figure 10 on page 31.

8-bit addresses in the range F0h–FFh (and the equivalent 4-bit addresses) are bank-independent, meaning they always access the control registers in Bank 0, regardless of the RP[3:0] value. Addresses in the range 00h–03h always access the Bank 0 Port registers unless Bank D or F is selected. (Port 01h is not implemented in this device.) When Bank D or F is selected, addresses 10h–EFh access the Bank 0 general-purpose registers.

The LDX and LDXI instructions or indirect addressing can be used to access the Bank 1–3 registers not accessible by 8-bit or working register addresses (12-bit addresses 100h–103h, 1F0h–1FFh, 200h–203h, 2F0h–2FFh, 300h–303h, and 3F0h–3FFh). See Linear Memory Addressing on page 32.

Stack

The Stack Pointer register (SPL) is Bank 0 register FFh. Operations that use the stack pointer always addresses Bank 0, regardless of the RP[3:0] setting. For details on stack, refer to *Z8 LXM CPU Core User Manual (UM0183)*.

This device does not use a stack pointer high byte. Bank 0 register FEh can be used to store user data, see User Data Register on page 36.

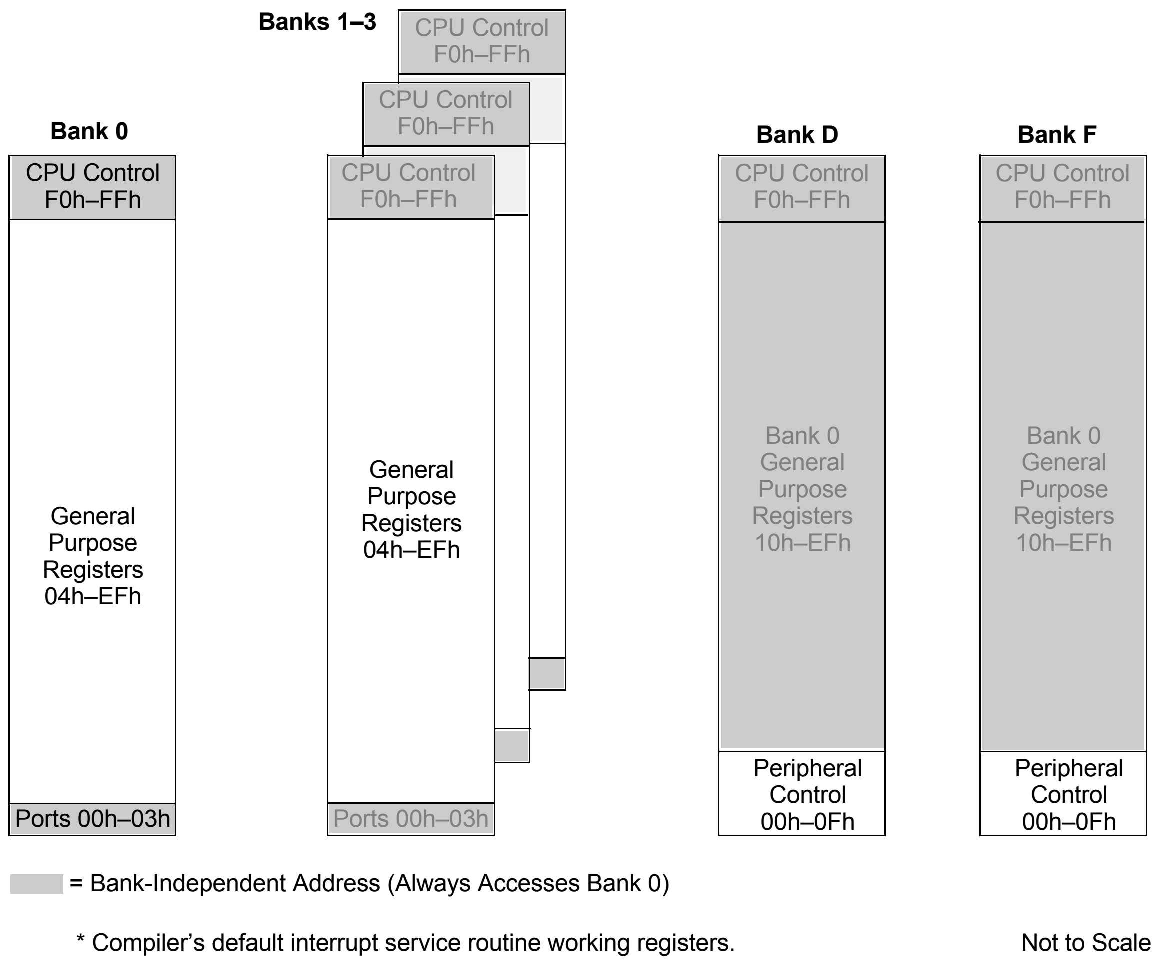


Figure 9. Register File 8-Bit Banked Address Map

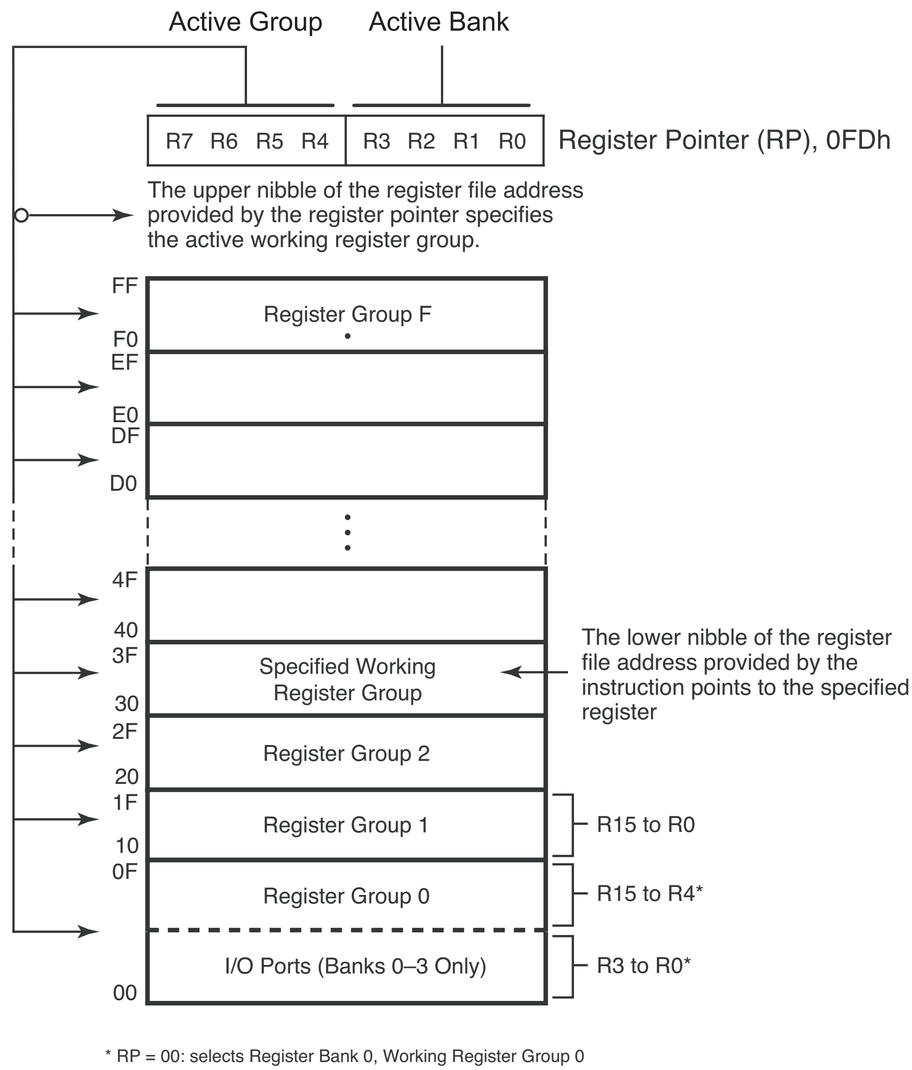


Figure 10. Register Pointer—Detail

Register Pointer Example

```
R253 RP = 00h
R0 = Port 0
R1 = Port 1
R2 = Port 2
R3 = Port 3
```

But if:

```
R253 RP = 0Dh
R0 = CTR0
R1 = CTR1
R2 = CTR2
R3 = CTR3
```

The counter/timers are mapped into ERF group D. Access is easily performed using the following code segment.

```
LD RP, #0Dh     ; Select ERF D for access to Bank D
                ; (working register group 0)
LD R0,#xx       ; load CTR0
LD 1, #xx       ; load CTR1
LD R1, 2        ; CTR2 → CTR1

LD RP, #7Dh     ; Select Expanded Register Bank D and working
                ; register group 7 of Bank 0 for access.
LD 71h, 2       ; CTR2 → register 71h
LD R1, 2        ; CTR2 → register 71h
```

Linear Memory Addressing

In addition to using the RP Register to designate a bank and working register group for 8-bit or 4-bit addressing, programs can use 12-bit linear addressing to load a register in any other bark to or from a register in the current bank. Linear addressing is implemented in the LDX and LDXI instructions only. Linear addressing treats the register file as if all of the registers are logically ordered end-to-end, as opposed to being grouped into banks and working register groups, as displayed in Figure 11 on page 33. For linear addressing, register file addresses are numbered sequentially from Bank 0, register 00h to Bank 0, register FFh, then continuing with Bank 1, register 00h, and so on up to Bank F, register FFh.

Using the LDX and/or the LDXI instructions, either the target or destination register location can be addressed through a 12-bit linear address value stored in a general-purpose register pair. For example, the following code uses linear addressing for the source of a register transfer operation and uses a working register address for the target.

```
SRP #%23     ;Set working register group 2 in bank 3
LD R0, #%55  ;Load 55 into working register R0 in the current
             ;group and bank (linear address 320h)
SRP #%12     ;Set working register group 1 in bank 2
LD R6, #%03  ;Load high byte of source linear address (0320h)
LD R7, #%20  ;Load low byte of source linear address (0320h)
LD R0, @RR6  ;Load linear address 320h contents (55h) into
             ;working register R0 in the current group and
             ;bank (linear address 210h)
```

As it can be seen in the above example, the source register is referenced via a linear address value contained within registers R6 and R7, whereas the destination is referenced via the SRP setting and a working register. For more information on the use of the LDX and LDXI instructions, refer to *Z8 LXM CPU Core User Manual (UM0183)*.

Note: *The LDE and LDEI instructions that existed in the Z8® CPU are no longer valid; they are replaced by the LDX and LDXI instructions.*

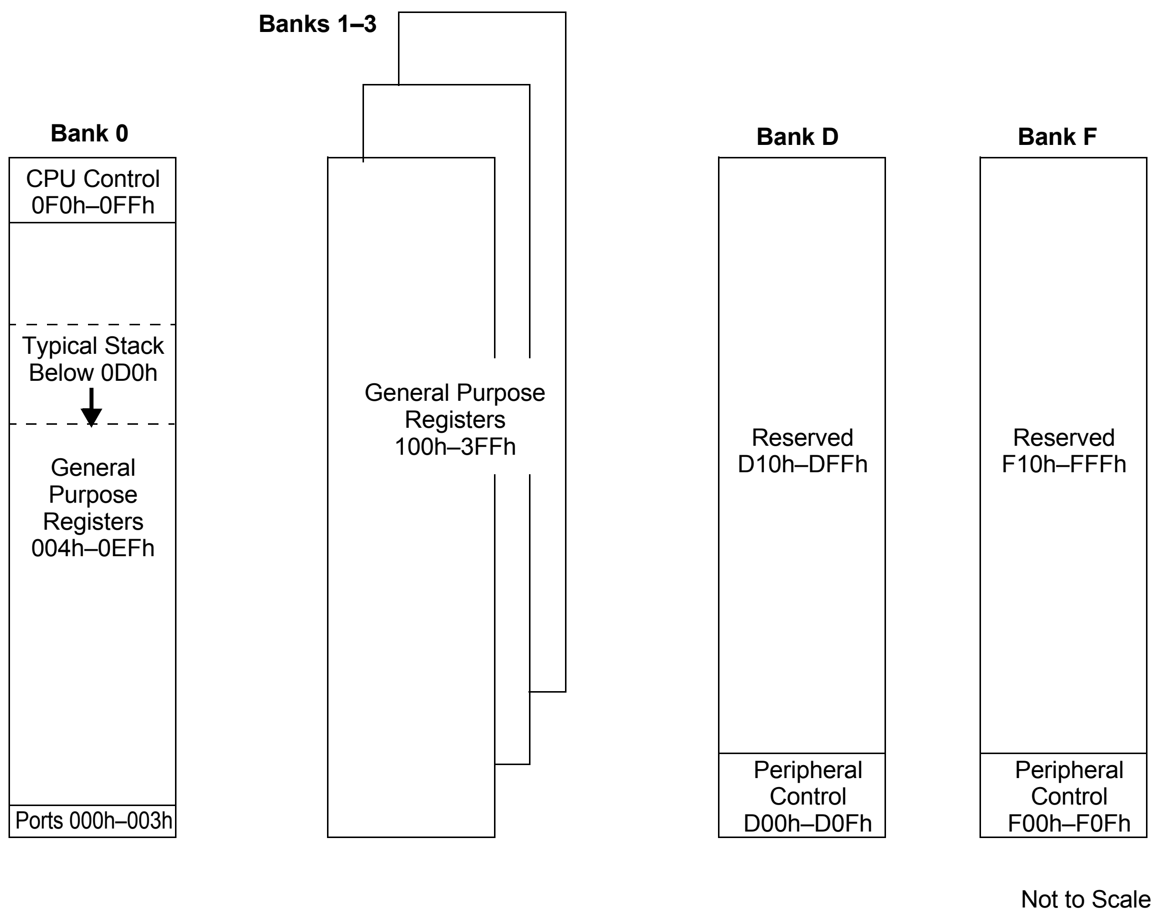


Figure 11. Register File LDX, LDXI Linear 12-Bit Address Map

Program Memory Paging Register

Bit 0 of the Program Memory Paging Register determines which 64 KB bank of program memory is read during the execution of the LDC and LDCI instructions (Table 16).

Table 16. Program Memory Paging Register (PMPR)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Page Toggle Enable	Reserved						Page Register
Reset	0	X	X	X	X	X	X	0
R/W	R/W	—	—	—	—	—	—	R/W
Address	Bank Independent: F0h; Linear: 0F0h							

Bit Position	**Value**	**Description**
[7]		Page Toggle Enable
	0	PMPR[0] changes only when written by software.
	1	If PMPR[0]=0, the CPU toggles PMPR[0] when LDCI increments past FFFFh.
[6:1]	—	Reserved—Reads 111111b; write 000000b for compatibility with possible future devices.
[0]		Page Register
	0	LDC, LDCI instructions access Page 0.
	1	LDC, LDCI instructions access Page 1.

Register Pointer Register

The upper nibble of the register pointer (Table 17) selects which working register group, of 16 bytes in the register file, is accessed out of the possible 256. The lower nibble selects the expanded register file bank and, in the case of the ZLP12840 MCU family, banks 0, 1, 2, 3, F, and D are implemented. A `0h` in the lower nibble allows the normal register file (Bank 0) to be addressed. Any other value from `01h` to `0Fh` exchanges the lower 16 registers to an expanded register bank.

Table 17. Register Pointer Register (RP)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Working Register Group Pointer				Register Bank Pointer			
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	Bank Independent: FDh; Linear 0FDh							

Bit Position	**Value**	**Description**
[7:4]		Working Register Group Pointer
	0h–Fh	Determines which 16 byte working group is addressed.
[3:0]		Register Bank Pointer
	0h–Fh	Determines which bank is active.

User Data Register

Bank-independent register FEh is available for user data storage (Table 18).

Note: *Do not use register FEh as a counter for the DJNZ instruction.*

Table 18. User Data Register (USER)

Bit	7	6	5	4	3	2	1	0
Field	User Data							
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	Bank Independent: FEh; Linear: 0FEh							

Bit Position	Value	Description
[7:0]	00h–FFh	User Data

Stack Pointer Register

The Stack Pointer Register contains the 8-bit address of the stack pointer. The stack pointer resides in Bank 0 of RAM. The stack address is decremented prior to a PUSH operation and incremented after a POP operation. The stack address always points to the data stored at the ‘top’ of the stack (the lowest stack address). During a call instruction, the contents of the Program Counter are saved on the stack. Interrupts cause the contents of the Program Counter and Flags registers to be saved on the stack. An overflow or underflow can occur when the stack address is incremented or decremented during normal operations. You must prevent this occurrence otherwise, it results in unpredictable operations (Table 19).

Table 19. Stack Pointer Register (SPL)

Bit	7	6	5	4	3	2	1	0
Field	Stack Pointer							
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	Bank Independent: FFh; Linear: 0FFh							

Bit Position	Description
[7:0]	Stack Pointer

Register File Summary

Table 20 maps each linear (12-bit) register file address to the associated register, mnemonic, and reset value. The table also lists the register bank (or banks) and corresponding 8-bit address, if any, for each register, plus a page link to the detailed register diagram.

Throughout this document, an "X" in a number denotes an undefined digit. A "—" (dash) in a table cell indicates that the corresponding attribute does not apply to the listed item. Reset value digits highlighted in grey are not reset by a Stop Mode Recovery. Register bit SMR[7] (shown in **boldface**) is set to 1 instead of reset by a Stop Mode Recovery.

Table 20. Register File Address Summary

Address (Hex)						
12-Bit	Bank	8-Bit	Register Description	Mnemonic	Reset	Page No
000	0–3	00	Port 0	P0	XXh	21
001	0–3	01	Reserved	—	—	—
002	0–3	02	Port 2	P2	XXh	23
003	0–3	03	Port 3	P3	0Xh	25
004–00F	0	04–0F	General-Purpose Registers (Bank 0 Only)	—	XXh	—
010–0EF	0,D,F	10–EF	General-Purpose Registers (Banks 0, D, F)	—	XXh	—
0F0	All	F0	Program Memory Paging Register	PMPR	0XXX_XXX0b	34
0F1	All	F1	UART Receive/Transmit Data Register	URDATA/ UTDATA	XXh	54
0F2	All	F2	UART Status Register	UST	0000_0010b	55
0F3	All	F3	UART Control Register	UCTL	00h	56
0F4	All	F4	UART Baud Rate Generator Constant	BCNST	FFh	57
0F5	All	F5	Reserved	—	—	—
0F6	All	F6	Port 2 Mode Register	P2M	FFh	22
0F7	All	F7	Port 3 Mode Register	P3M	XXXX_X000b	24
0F8	All	F8	Port 0 Mode Register	P01M	X1XX_XXX1b	20
0F9	All	F9	Interrupt Priority Register	IPR	XXh	90
0FA	All	FA	Interrupt Request Register	IRQ	00h	92

Table 20. Register File Address Summary (Continued)

Address (Hex) 12-Bit	Bank	8-Bit	Register Description	Mnemonic	Reset	Page No
0FB	All	FB	Interrupt Mask Register	IMR	0XXX_XXXXb	89
0FC	All	FC	Flags Register	FLAGS	XXh	118
0FD	All	FD	Register Pointer	RP	00h	35
0FE	All	FE	User Data Register	USER	XXh	36
0FF	All	FF	Stack Pointer Register	SPL	XXh	37
100–103	—	—	General-Purpose Registers (12-Bit Only)	—	XXh	—
104–1EF	1	04–EF	General-Purpose Registers	—	XXh	—
1F0–203	—	—	General-Purpose Registers (12-Bit Only)	—	XXh	—
204–2EF	2	04–EF	General-Purpose Registers	—	XXh	—
2F0–303	—	—	General-Purpose Registers (12-Bit Only)	—	XXh	—
304–3EF	3	04–EF	General-Purpose Registers	—	XXh	—
3F0–3FF	—	—	General-Purpose Registers (12-Bit Only)	—	XXh	—
400–CFF	—	—	Reserved	—	—	—
D00	D	00	Counter/Timer 8 Control Register	CTR0	0000_0000b	77
D01	D	01	Timer 8 and Timer 16 Common Functions	CTR1	0000_0000b	79
D02	D	02	Counter/Timer 16 Control Register	CTR2	0000_0000b	82
D03	D	03	Timer 8/Timer 16 Control Register	CTR3	0000_0XXXb	83
D04	D	04	Counter/Timer 8 Low Hold Register	TC8L	00h	76
D05	D	05	Counter/Timer 8 High Hold Register	TC8H	00h	76
D06	D	06	Counter/Timer 16 Low Hold Register	TC16L	00h	75
D07	D	07	Counter/Timer 16 High Hold Register	TC16H	00h	75
D08	D	08	Timer 16 Capture Low Register	LO16	00h	74
D09	D	09	Timer 16 Capture High Register	HI16	00h	74
D0A	D	0A	Timer 8 Capture Low Register	LO8	00h	73
D0B	D	0B	Timer 8 Capture High Register	HI8	00h	73
D0C	D	0C	Low-Voltage Detection Register	LVD	1111_1000b	98
D0D–D0F	D	0D–0F	Reserved	—	—	—
D10–DFF	—	—	Reserved (8-Bit access goes to Bank 0)	—	—	—

Table 20. Register File Address Summary (Continued)

Address (Hex)						
12-Bit	**Bank**	**8-Bit**	**Register Description**	**Mnemonic**	**Reset**	**Page No**
F00	F	00	Port Configuration Register	PCON	XXXX_X1X0b	19
F01–F09	F	01–09	Reserved	—	—	—
F0A	F	0A	Stop Mode Recovery Register 4	SMR4	XXX0 0000b _	111
F0B	F	0B	Stop Mode Recovery Register	SMR	**0**010_0000b	102
F0C	F	0C	Stop Mode Recovery Register 1	SMR1	00h	105
F0D	F	0D	Stop Mode Recovery Register 2	SMR2	X0X0_00XXb	107
F0E	F	0E	Stop Mode Recovery Register 3	SMR3	X0h	110
F0F	F	0F	Watchdog Timer Mode Register	WDTMR	XXXX_1101b	112
F10–FFF	—	—	Reserved (8-Bit access goes to Bank 0)	—	—	—

Infrared Learning Amplifier

The ZLP12840 MCU's infrared learning amplifier allows you to detect and decode infrared transmissions directly from the output of the receiving diode without the need for external circuitry. See Port 3 on page 14.

An IR diode can be connected to the IR amplifier as displayed in Figure 12. When the IR amplifier is enabled and an input current is detected on Port 3, pin 1 (P31), the IR amplifier outputs a logical High value. When the input current is below the switching threshold of the IR amplifier, the amplifier outputs a logical Low value.

Within the MCU, the IR amplifier output goes to the capture/timer logic, which can be programmed to demodulate the IR signal. The IR amplifier output can also be read by the CPU, or drive the Port 3, pin 4 (P34) output if write-only register bit PCON[0] is written with a 1.

The IR learning amplifier can demodulate signals up to a frequency of 500 kHz. A special mode exists that allows you to capture the third, fourth, and fifth edges of the IR amplifier output and generate an interrupt.

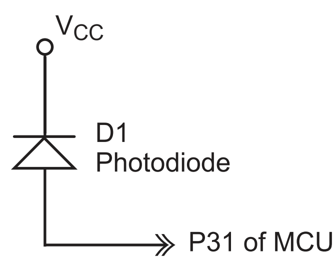


Figure 12. Learning Amplification Circuitry with the ZLP12840 MCU

For details on programming the timers to demodulate a received signal, see Timers on page 59.

Universal Asynchronous Receiver/Transmitter

The universal asynchronous receiver/transmitter (UART) is a full-duplex communication channel capable of handling asynchronous data transfers. The two UARTs use a single 8-bit data mode with selectable parity. Features of the UARTs include:

- 8-bit asynchronous data transfer
- Selectable even- and odd-parity generation and checking
- One or two Stop bits
- Separate transmit and receive interrupts
- Framing, overrun, and break detection
- Separate transmit and receive enables
- 8-bit Baud Rate Generator (BRG)
- Baud Rate Generator timer mode
- UART operational during HALT mode

Table 21. UART Control Registers

Address (Hex)						
12-Bit	**Bank**	**8-Bit**	**Register Description**	**Mnemonic**	**Reset**	**Page No**
0F1	All	F1	UART Receive/Transmit Data Register	URDATA/ UTDATA	XXh	54
0F2	All	F2	UART Status Register	UST	0000_0010b	55
0F3	All	F3	UART Control Register	UCTL	00h	56
0F4	All	F4	UART Baud Rate Generator Constant	BCNST	FFh	57

Architecture

The UARTs consist of three primary functional blocks: transmitter, receiver, and Baud Rate Generator. The UART transmitter and receiver function independently, but employ the same baud rate and data format. Figure 13 on page 46 displays the UART architecture.

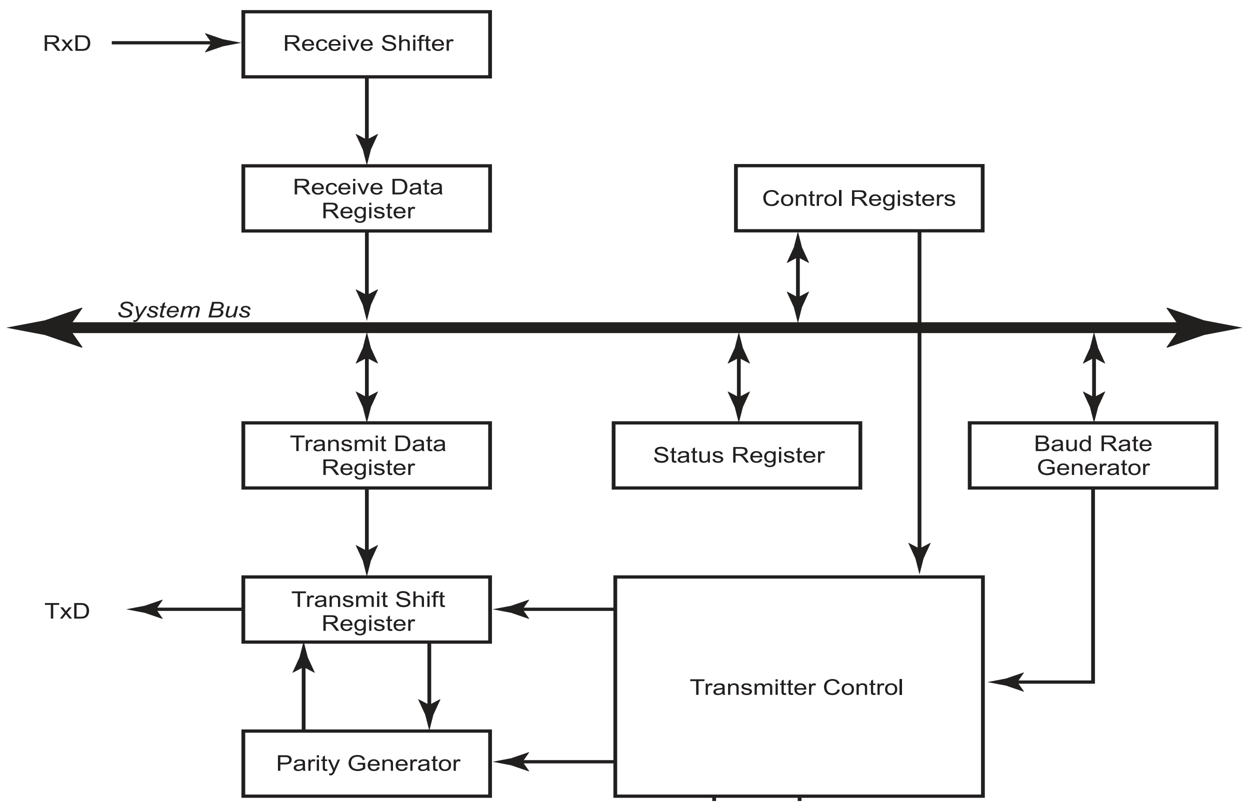


Figure 13. UART Block Diagram

Operation

The UART channel can be used to communicate with a master microprocessor or as a slave microprocessor, both of which exhibit transmit and receive functionality. You can either operate the UART channel by polling the UART Status register or via interrupts. The UART remains active during HALT mode. If neither the transmitter nor the receiver is enabled, the UART baud rate generator can be used as an additional timer. The UART contains a noise filter for the receiver that can be enabled.

Data Format

The UART always transmits and receives data in an 8-bit data format, with the least-significant bit occurring first. An even or odd parity bit can be optionally added to the data stream. Each character begins with an active Low Start bit and ends with either 1 or 2 active High Stop bits. Figure 14 and Figure 15 on page 47 display the asynchronous data format employed by the UARTs without parity and with parity, respectively.

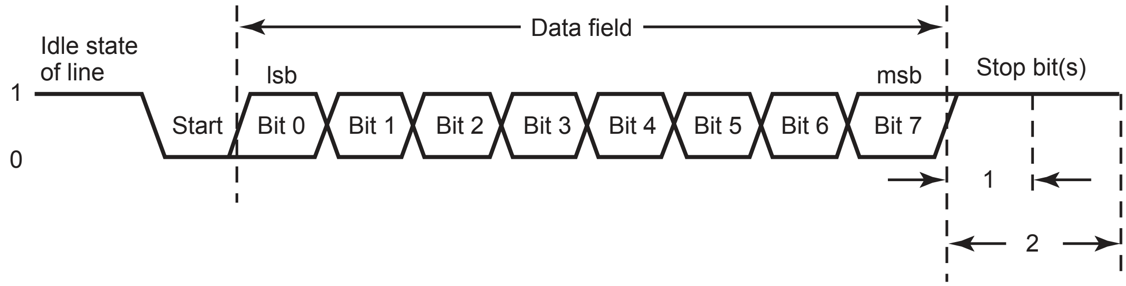


Figure 14. UART Asynchronous Data Format without Parity

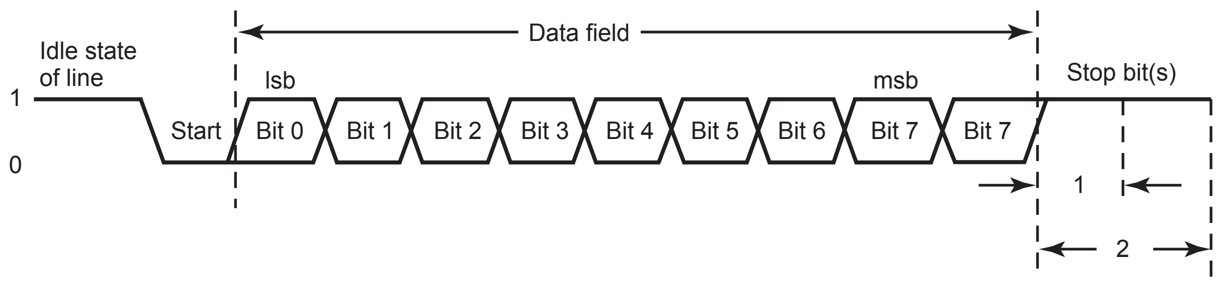


Figure 15. UART Asynchronous Data Format with Parity

Transmitting Data Using the Polled Method

Follow the steps below to transmit data using the polled method of operation:

1. Write to the baud rate generator constant (BCNST) register, address `0F4h`, to set the appropriate baud rate.
2. Write a 0 to bit 6 of the P01M register.
3. Write to the UART control register (UCTL) to:
   (a) Set the transmit enable bit, UCTL[7], to enable the UART for data transmission.
   (b) If parity is appropriate, set the parity enable bit, UCTL[4] to 1 and select either Even or Odd parity (UCTL[3]).
4. Check the Transmit Status register bit, UST[2], to determine if the Transmit Data register is empty (indicated by a 1). If empty, continue to step 6. If the Transmit Data register is full (indicated by a 0), continue to monitor the UST[2] bit until the Transmit Data register becomes available to receive new data.
5. Write the data byte to the UART Transmit Data register, `0F1h`. The transmitter automatically transfers the data to the internal transmit shift register and transmits the data.

6. To transmit additional bytes, return to step 4.
7. Before disabling the transmitter, read the transmit completion status bit, UST[1]. If UST[1]=0, continue to monitor the bit until it changes to 1, which indicates that all data in the Transmit Data and internal shift registers has been transmitted.

Caution: *Data written while the transmit enable bit is clear (UCTL[7]=0) will not be transmitted. Data written while the transmit data status bit is clear (UST[2]=0) overwrites the previous value written, so the previous written value will not be transmitted. Disabling the UART transmitter while the transmit completion status bit is clear (UST[1]=0) can corrupt the byte being transmitted.*

Transmitting Data Using the Interrupt-Driven Method

The UART transmitter interrupt indicates the availability of the Transmit Data register to accept new data for transmission. Follow the steps below to configure the UART for interrupt-driven data transmission:

1. Write to the BCNST register to set the appropriate baud rate.
2. Write a 0 to bit 6 of the P01M register.
3. Execute a DI instruction to disable interrupts.
4. Write to the Interrupt control registers to enable the UART Transmitter interrupt and set the appropriate priority.
5. Write to the UART Control register to:
   (a) Set the transmit enable bit (UCTL bit 7) to enable the UART for data transmission.
   (b) Enable parity, if appropriate, and select either even or odd parity.
6. Execute an EI instruction to enable interrupts.
7. Because the transmit buffer is empty, an interrupt is immediately executed.
8. Write the data byte to the UART Transmit Data register. The transmitter automatically transfers the data to the internal transmit shift register and transmits the data.
9. Execute the IRET instruction to return from the interrupt-service routine and wait for the Transmit Data register to again become empty.
10. Before disabling the transmitter, read the transmit completion status bit, UST[1]. If UST[1]=0, continue to monitor the bit until it changes to 1, which indicates that all data in the Transmit Data and internal shift registers has been transmitted.

Caution: *Data written while the transmit enable bit is clear (UCTL[7]=0) will not be transmitted. Data written while the transmit data status bit is clear (UST[2]=0) overwrites the previous value written, so the previous written value will not be transmitted. Disabling the UART transmitter while the transmit completion status bit is clear (UST[1]=0) can corrupt the byte being transmitted.*

Receiving Data Using the Polled Method

Follow the steps below to configure the UART for polled data reception:

1. Write to the BCNST register to set the appropriate baud rate.
2. Write to the UART control register (UCTL) to:
   (a) Set the receive enable bit (UCTL[6]) to enable the UART for data reception
   (b) Enable parity, if appropriate and select either even or odd parity
3. Check the receive status bit in the UART Status register, bit UST[7], to determine if the Receive Data register contains a valid data byte (indicated by a 1). If UST[7] is set to 1 to indicate available data, continue to step 4. If the Receive Data register is empty (indicated by a 0), continue to monitor the UST[7] bit awaiting reception of the valid data.
4. Read data from the UART Receive Data register.
5. Return to step 3 to receive additional data.

Receiving Data Using the Interrupt-Driven Method

The UART Receiver interrupt indicates the availability of new data (as well as error conditions). Follow the steps below to configure the UART receiver for interrupt-driven operation:

1. Write to the UART BRG Constant registers to set the appropriate baud rate.
2. Execute a DI instruction to disable interrupts.
3. Write to the interrupt control registers to enable the UART receiver interrupt and set the appropriate priority.
4. Clear the UART Receiver interrupt in the applicable Interrupt Request register.
5. Write to the UART Control register (UCTL) to:
   (a) Set the receive enable bit (UCTL[6]) to enable the UART for data reception.
   (b) Enable parity, if appropriate, and select either even or odd parity.
6. 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) performs the following:

1. Checks the UART Status register to determine the source of the interrupt, whether it is an error, break, or received data.
2. Reads the data from the UART Receive Data register if the interrupt was caused by data available.

3. Clears the UART receiver interrupt in the applicable Interrupt Request register.
4. Executes the IRET instruction to return from the interrupt service routine and await more data.

UART Interrupts

The UART features separate interrupts for the transmitter and the receiver. In addition, when the UART primary functionality is disabled, the Baud Rate Generator can also function as a basic timer with interrupt capability.

Note: *When the UART is set to run at higher baud rates, the UART receiver's service routine might not have enough time to read and manipulate all bits in the UART Status register (especially bits generating error conditions) for a received byte before the next byte is received. Devise your own hand-shaking protocol to prevent the transmitter from transmitting more data while current data is being serviced.*

Transmitter Interrupts

The transmitter generates a single interrupt when the Transmit Status bit, UST[2], is set to 1. This indicates that the transmitter is ready to accept new data for transmission. The Transmit Status interrupt occurs after the internal transmit shift register has shifted the first bit of data out. At this point, the Transmit Data register can be written with the next character to send. This provides 7 bit periods of latency to load the Transmit Data register before the transmit shift register completes shifting the current character. Writing to the UART Transmit Data register clears the UST[2] bit to 0. The interrupt is cleared by writing a 0 to the Transmit Data register.

Receiver Interrupts

The receiver generates an interrupt when any of the following occurs:

- A data byte has been received and is available in the UART Receive Data register—This interrupt can be disabled independent of the other receiver interrupt sources. The received data interrupt occurs once the receive character has been received and placed in the Receive Data register. Software must respond to this received data available condition before the next character is completely received to avoid an overrun error. The interrupt is cleared by reading from the UART Receive Data register.
- A break is received—A break is detected when a 0 is sent to the receiver for the full byte plus the parity and stop bits. After a break is detected, it will interrupt immediately if there is no valid data in the Receive Data register. If data is present in the Receive Data register, an interrupt will occur after the UART Receive Data register is read.
- An overrun is detected—An overrun occurs when a byte of data is received while there is valid data in the UART Receive Data register that is not read. The interrupt

will be generated when the UART Receive Data register is read. The interrupt is cleared by reading the UART Receive Data register. When an overrun error occurs, the additional data byte will not overwrite the data currently stored in the UART Receive Data register.

- A data framing error is detected—A data framing error is detected when the first stop bit is 0 instead of 1. When configured for 2 stop bits, a data framing error is only detected when the first stop bit is 0. A framing error interrupt is generated when the framing error is detected. Reading the UART Receive Data register clears the interrupt.

Note: *It is important to ensure that the transmitter uses the same stop bit configuration as the receiver.*

UART Overrun Errors

When an overrun error condition occurs the UART prevents overwriting of the valid data currently in the Receive Data register. The Break Detect and Overrun status bits are not displayed until after the valid data has been read.

After the valid data has been read, the UART Status (UST) register is updated to indicate the overrun condition (and Break Detect, if applicable). The UST[7] bit is set to 1 to indicate that the Receive Data register contains a data byte. However, because the overrun error occurred, this byte may not contain valid data and should be ignored. The Break Detect bit, UST[3], indicates if the overrun was caused by a break condition on the line. After reading the status byte indicating an overrun error, the Receive Data register must be read again to clear the error bits is the UART Status 0 register. Updates to the Receive Data register occur only when the next data word is received.

UART Data and Error Handling Procedure

Figure 16 on page 52 displays the recommended procedure for use in UART receiver interrupt service routines.

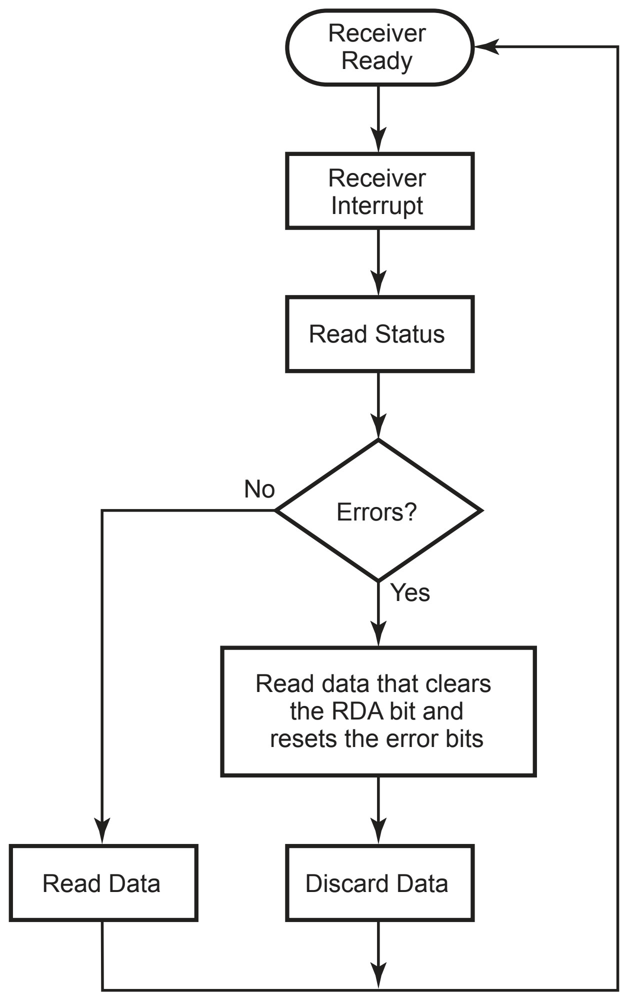


Figure 16. UART Receiver Interrupt Service Routine Flow

Baud Rate Generator Interrupts

If the Baud Rate Generator interrupt enable is set, the UART Receiver interrupt asserts when the UART Baud Rate Generator reloads. This action allows the Baud Rate Generator to function as an additional counter if the UART functionality is not employed.

UART Baud Rate Generator

The UART Baud Rate Generator creates a lower frequency baud rate clock for data transmission. The input to the Baud Rate Generator is the system clock. The UART Baud Rate Constant register contains an 8-bit baud rate divisor value (BCNST[7:0]) that sets the data transmission rate (baud rate) of the UART. For programmed register values other than 00h, the UART data rate is calculated using the following equation:

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

When the UART Baud Rate Low Register is programmed to 00h, the UART data rate is calculated using the following equation:

$$\text{UART Data Rate (bits/s)} = \frac{\text{System Clock Frequency (Hz)}}{4096}$$

When the UART Baud Rate Generator is used as a general-purpose counter, the counters time out period can be computed as follows based upon the counters clock input being a divide by 16 of the system clock and the maximum count value being 255:

$$\text{Time-Out Period } (\mu s) = \frac{16 \times \text{UART Baud Rate Divisor Value (BCNST)}}{\text{System Clock Frequency (MHz)}}$$

Note: *In general, the system clock frequency is the XTAL clock frequency divided by 2.*

When the UART is disabled, the Baud Rate Generator can function as a basic 8-bit timer with interrupt on time-out. Follow the steps below to configure the Baud Rate Generator as a timer with interrupt on time-out:

1. Disable the UART by clearing the receive and transmit enable bits, UCTL[7:6] to 0.
2. Load the appropriate 8-bit count value into the UART Baud Rate Generator Constant register. The count frequency is the system clock frequency in Hz divided by 16.
3. Enable the Baud Rate Generator timer function and associated interrupt by setting the Baud Rate Generator bit (UCTL bit 0) in the UART Control Register to 1. When configured as an 8-bit timer, the count value, instead of the reload value, is read, and the counter begins counting down from its initial programmed value. Upon timing out (reaching a value of 1), if the time-out interrupt is enabled, an interrupt will be produced. The counter will then reload its programmed start value and begin counting down again.

Table 22 lists a number of BCNST register settings at various baud rates and system clock frequencies.

Table 22. BCNST Register Settings Examples

Target UART Data Rate (baud)	System Clock = 4 MHz, Crystal Clock = 8 MHz	System Clock = 3 MHz, Crystal Clock = 6 MHz
2400	BCNST = 01101000 Actual baud rate = 2403	BCNST = 01001110 Actual baud rate = 2403

Table 22. BCNST Register Settings Examples (Continued)

Target UART Data Rate (baud)	System Clock = 4 MHz, Crystal Clock = 8 MHz	System Clock = 3 MHz, Crystal Clock = 6 MHz
4800	BCNST = 00110100 Actual baud rate = 4807	BCNST = 00100111 Actual baud rate = 4807
9600	BCNST = 00011010 Actual baud rate = 9615	BCNST = 00010100 Actual baud rate = 9375
19200	BCNST = 00001101 Actual baud rate = 19230	BCNST = 00001010 Actual baud rate = 18750

UART Receive Data Register/UART Transmit Data Register

The UART Receive/Transmit Data Register is used to send and retrieve data from the UART channel. When the UART receives a byte of data, it can be read from this register. The UART receive interrupt is cleared when this register is used. Data written to this register is transmitted by the UART (Table 23).

Table 23. UART Receive/Transmit Data Register (URDATA/UTDATA)

Bit	7	6	5	4	3	2	1	0
Field	UART Receive/Transmit							
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	Bank Independent: F1h; Linear: 0F1h							

Bit Position	Description
[7:0]	UART Receive/Transmit When read, returns received data. When written, transmits written data.

UART Status Register

The UART Status Register shows the status of the UART. Bits [6:3] are cleared by reading the UART Receive/Transmit Register (`F1h`) (Table 24).

Table 24. UART Status Register (UST)

Bit	7	6	5	4	3	2	1	0
Field	Receive Status	Parity Error	Overrun Error	Framing Error	Break	Transmit Data	Transmit Complete	Noise Filter
Reset	0	0	0	0	0	0	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Bank Independent: F2h; Linear: 0F2h							

Bit Position	Value	Description
[7]		Receive Status—Set when data is received; cleared when URDATA is read.
	0	UART Receive Data Register empty.
	1	UART Receive Data Register full.
[6]		Parity—Set when a parity error occurs; cleared when URDATA is read.
	0	No parity error occurs.
	1	Parity error occurs.
[5]		Overrun—Set when an overrun error occurs; cleared when URDATA is read.
	0	No overrun error occurs.
	1	Overrun error occurs.
[4]		Framing—Set when a framing error occurs; cleared when URDATA is read.
	0	No framing error occurs.
	1	Framing error occurs.
[3]		Break—Set when a break is detected; cleared when URDATA is read.
	0	No break occurs.
	1	Break occurs.
[2]		Transmit Data Status—Set when the UART is ready to transmit; cleared when TRDATA is written.
	0	Do not write to the UART Transmit Data Register.
	1	UART Transmit Data Register ready to receive additional data.
[1]		Transmit Completion Status
	0	Data is currently transmitting.
	1	Transmission is complete.

Bit Position	Value	Description
[0]	Read	Noise Filter—Detects noise during data reception.
	0	No noise detected.
	1	Noise detected.
	Write	
	0	Turn OFF noise filter.
	1	Turn ON noise filter.

UART Control Register

As the name implies, the UART Register controls the UART. In addition to setting bit 5 (see Table 25), also set appropriate bit in the Interrupt Mask Register (see Table 45 on page 92).

Note: *This register is not reset after a Stop Mode Recovery.*

Table 25. UART Control Register (UCTL)

Bit	7	6	5	4	3	2	1	0
Field	Transmitter Enable	Receiver Enable	UART Interrupts Enable	Parity Enable	Parity Select	Send Break	Stop Bits	Baud Rate Generator
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	Bank Independent: F3h; Linear: 0F3h							

Bit Position	Value	Description
[7]	0	Transmitter disabled.
	1	Transmitter enabled.
[6]	0	Receiver disabled.
	1	Receiver enabled.
[5]	0	UART Interrupts disabled.
	1	UART Interrupts enabled.
[4]	0	Parity disabled.
	1	Parity enabled.

Bit Position	Value	Description
[3]	0 1	Even parity selected. Odd parity selected.
[2]	0 1	No break is sent. Send Break (force Tx output to 0).
[1]	0 1	One stop bit. Two stop bits.
[0]	 0 1	Baud Rate Generator—When the transmitter and receiver are disabled, the BRG can be used as an additional timer. When setting this bit, clear bits [7:6] in this register. Also set bit [5] if an interrupt is desired when the BRG is reloaded. BRG used as Baud Rate Generator for UART. BRG used as timer.

Baud Rate Generator Constant Register

The UART baud rate generator determines the frequency at which UART data is received and transmitted. This baud rate is determined by the following equation:

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

The system clock is usually the crystal clock divided by 2.

When the UART baud rate generator is used as an additional timer, a Read from this register will return the actual value of the count of the BRG in progress and not the reload value (Table 26).

Note: *This register is not reset after a Stop Mode Recovery.*

Table 26. Baud Rate Generator Constant Register (BCNST)

Bit	7	6	5	4	3	2	1	0
Field	Baud Rate Generator Constant							
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Bank Independent: F4h; Linear: 0F4h							

Bit Position	Description
[7:0]	BRG Constant When read, returns the actual timer count value (when UCTL[0]=1). When written, sets the Baud Rate Generator Constant. The actual baud rate frequency = XTAL ÷ (32 x BCNST).

Timers

The Crimzon® ZLP12840 MCU infrared timer contains a 16-bit and an 8-bit counter/timer, each of which can be used simultaneously for transmitting. In addition, both timers can be used for demodulating an input carrier wave. Both timers share a single input pin.

Figure 17 displays the counter/timer architecture, which is designed to help unburden the program from coping with such real-time problems as generating complex waveforms or receiving and demodulating complex waveforms and pulses.

In addition to the 16-bit and 8-bit timers, the UART's baud rate generator can be used as an additional 8-bit timer when the UART receiver is not in use. See Universal Asynchronous Receiver/Transmitter on page 45.

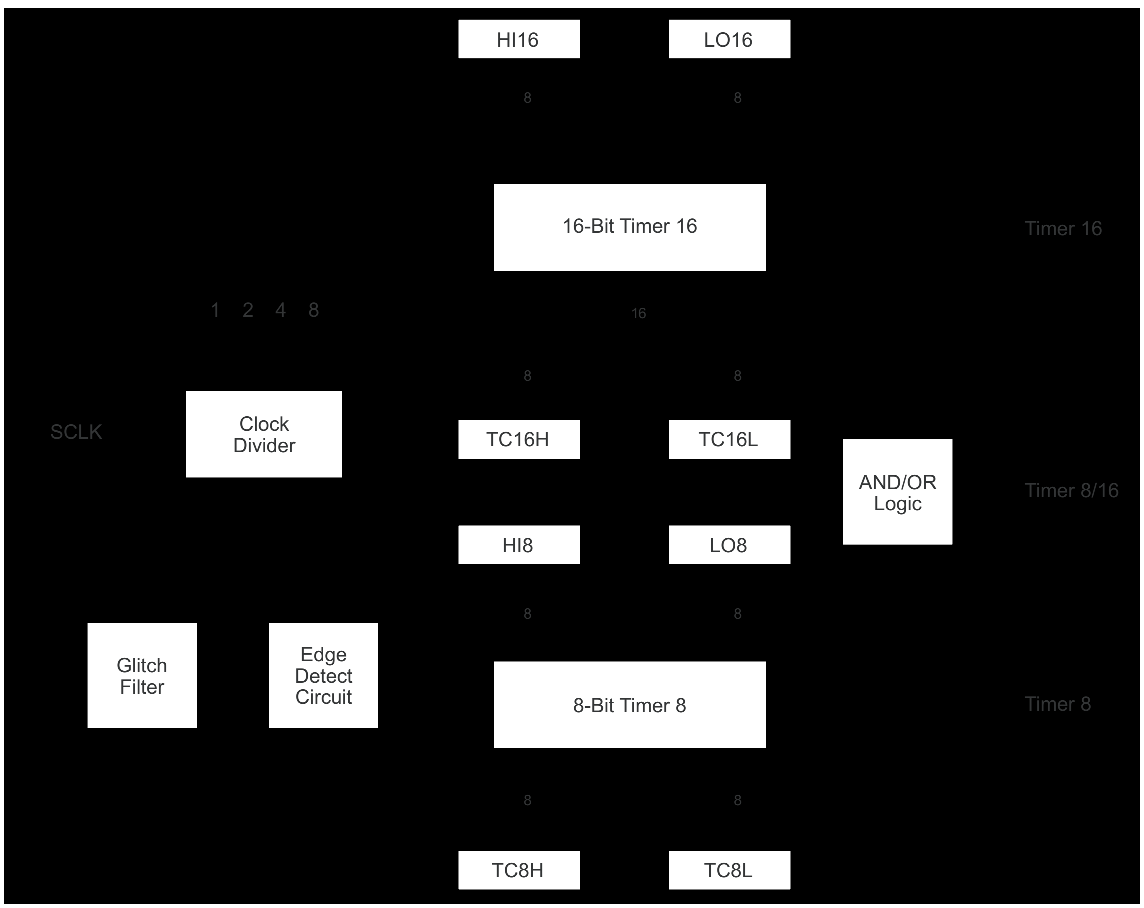


Figure 17. Counter/Timers Block Diagram

Table 27 summarizes the registers used to control timers. Some timer functions can also be affected by control registers for other peripheral functions.

Table 27. Timer Control Registers

Address (Hex)						
12-Bit	Bank	8-Bit	Register Description	Mnemonic	Reset	Page No
D00	D	00	Counter/Timer 8 Control Register	CTR0	0000_0000b	77
D01	D	01	Timer 8 and Timer 16 Common Functions	CTR1	0000_0000b	79
D02	D	02	Counter/Timer 16 Control Register	CTR2	0000_0000b	82
D03	D	03	Timer 8/Timer 16 Control Register	CTR3	0000_0XXXb	83
D04	D	04	Counter/Timer 8 Low Hold Register	TC8L	00h	76
D05	D	05	Counter/Timer 8 High Hold Register	TC8H	00h	76
D06	D	06	Counter/Timer 16 Low Hold Register	TC16L	00h	75
D07	D	07	Counter/Timer 16 High Hold Register	TC16H	00h	75
D08	D	08	Timer 16 Capture Low Register	LO16	00h	74
D09	D	09	Timer 16 Capture High Register	HI16	00h	74
D0A	D	0A	Timer 8 Capture Low Register	LO8	00h	73
D0B	D	0B	Timer 8 Capture High Register	HI8	00h	73

Counter/Timer Functional Blocks

The Crimzon ZLP12840 MCU infrared timer contains a glitch filter for removing noise from the input when demodulating an input carrier. Each timer features its own DEMODULATING mode. The T8 timer has the ability to capture only one cycle of a carrier wave of a high-frequency waveform. Each timer can be simultaneously used to generate a signal output.

Input Circuit

Depending on the setting of register bits P3M[2:1] and CTR1[6], the timer/counter input monitors one of the following conditions:

- The P31 digital signal, if CTR1[6]=0 and P3M[2:1]=00.
- The P31 analog comparator output, if CTR1[6]=0 and P3M[2:1]=01.
- The P31 IR amplifier output, if CTR1[6]=0 and P3M[2]=1.

- The P20 digital signal, if CTR16=1.

Based on register bits CTR1[5:4], a pulse is generated at when a rising edge, falling edge, or any edge is detected. Glitches in the input signal are filtered out if they are shorter than the glitch filter width specified in register bits CTR1[3:2]. The input circuit is displayed in Figure 18.

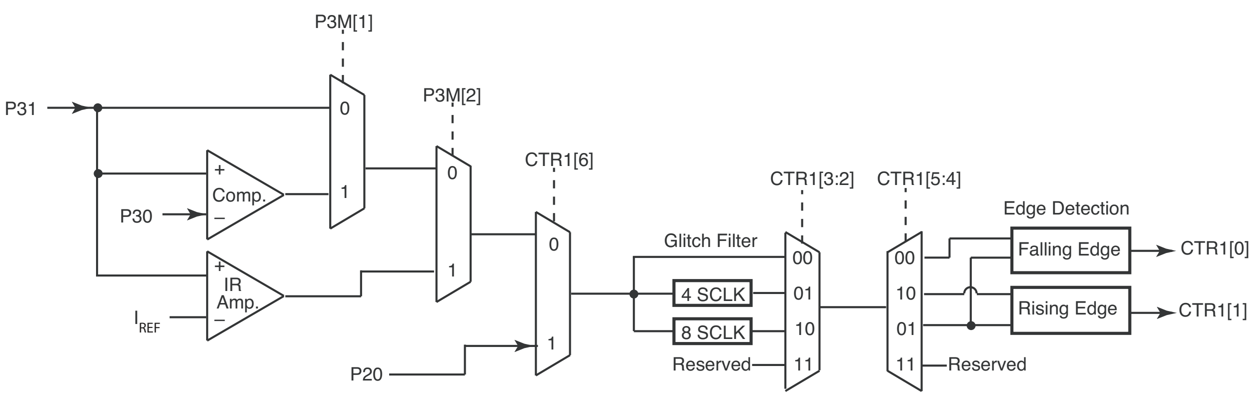


Figure 18. Counter/Timer Input Circuit

T8 TRANSMIT Mode

Before T8 is enabled, the output of T8 depends on CTR1, bit 1. If it is 0, T8_OUT is 1; if it is 1, T8_OUT is 0. See Figure 19.

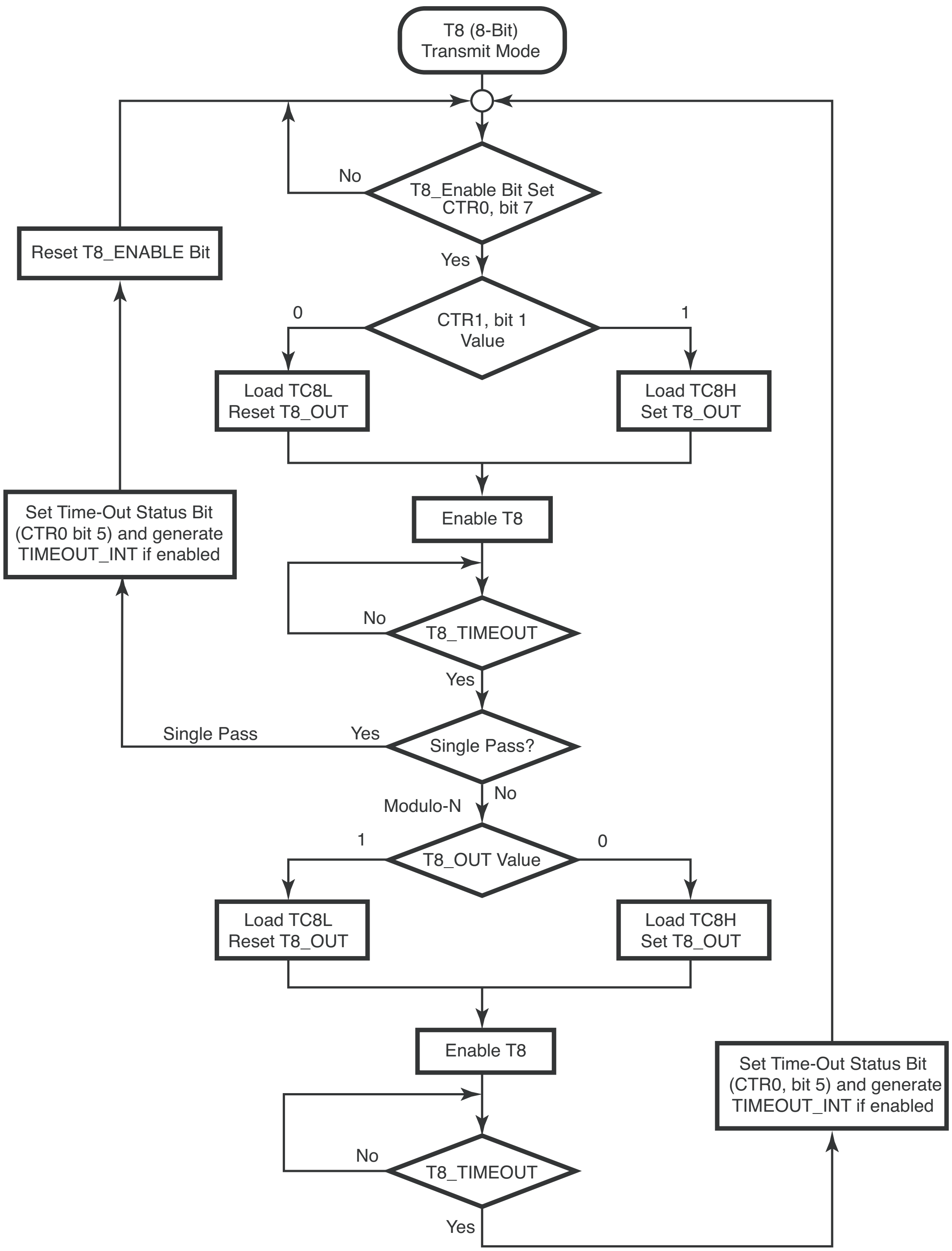


Figure 19. TRANSMIT Mode Flowchart

When T8 is enabled, the output T8_OUT switches to the initial value (CTR1, bit 1). If the initial value (CTR1, bit 1) is 0, TC8L is loaded; otherwise, TC8H is loaded into the counter. In SINGLE-PASS mode (CTR0, Bit 6), T8 counts down to 0 and stops, T8_OUT toggles, the time-out status bit (CTR0, bit 5) is set, and a time-out interrupt can be generated if it is enabled (CTR0, bit 1). In MODULO-N mode, upon reaching terminal count, T8_OUT is toggled, but no interrupt is generated. From that point, T8 loads a new count (if the T8_OUT level now is 0), TC8L is loaded; if it is 1, TC8H is loaded. T8 counts down to 0, toggles T8_OUT, and sets the time-out status bit (CTR0, bit 5), thereby generating an interrupt if enabled (CTR0, bit 1). One cycle is thus completed. T8 then loads from TC8H or TC8L according to the T8_OUT level and repeats the cycle (see Figure 20).

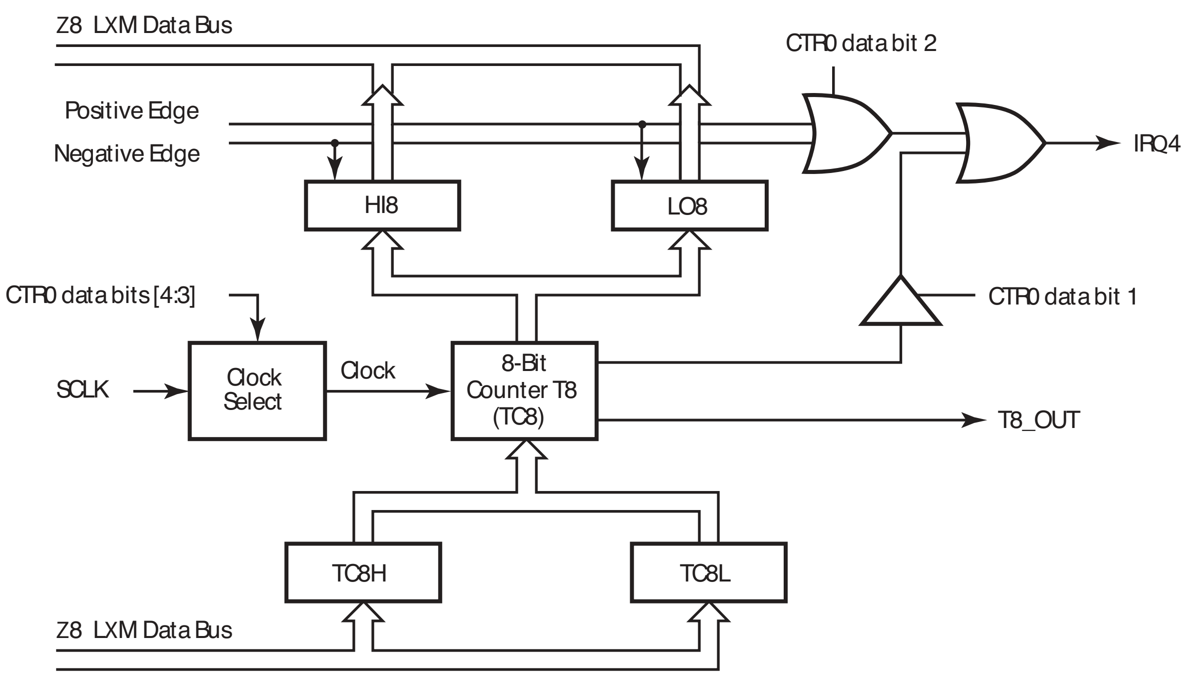


Figure 20. 8-Bit Counter/Timer Circuits

You can modify the values in TC8H or TC8L at any time. The new values take effect when they are loaded.

Caution: *An initial count of 1 is not allowed (a non-function occurs). An initial count of 0 causes TC8 to count from 0 to* FFh *to* FEh.

Note: *The 'h' suffix denotes hexadecimal values.*

Note: *Transition from 0 to* FFh *is not a time-out condition.*

Caution: *Using the same instructions for stopping the counter/timers and setting the status bits is not recommended.*

Two successive commands are necessary. First, the counter/timers must be stopped. Second, the status bits must be reset. These commands are required because it takes one counter/timer clock interval for the initiated event to actually occur (see Figure 21 and Figure 22).

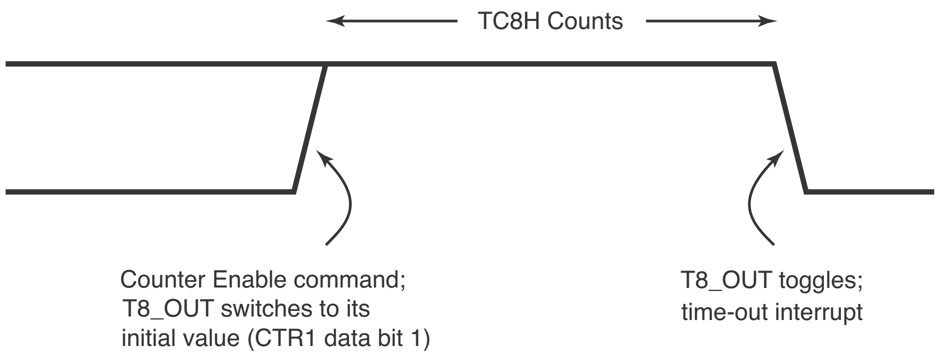


Figure 21. T8_OUT in SINGLE-PASS Mode

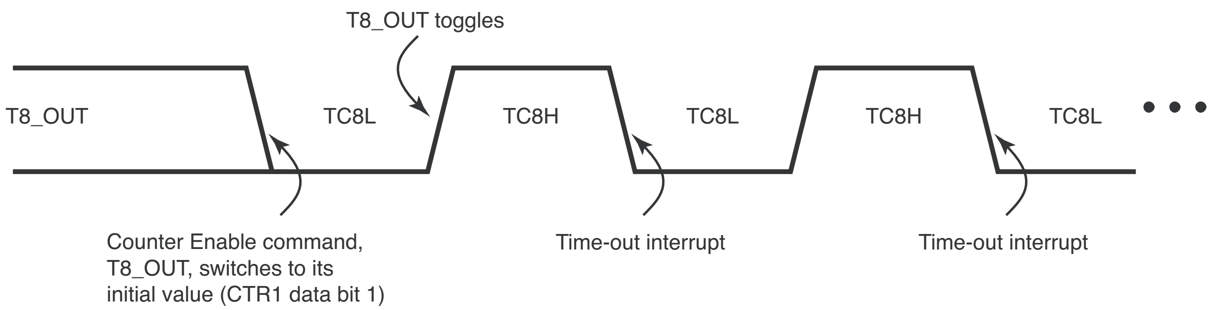


Figure 22. T8_OUT in MODULO-N Mode

T8 DEMODULATION Mode

You must program TC8L and TC8H to FFh. After T8 is enabled, when the first edge (rising, falling, or both depending on CTR1 bits [5:4]) is detected, it starts to count down. When a subsequent edge (rising, falling, or both depending on CTR1 bits [5:4]) is detected during counting, the current value of T8 is complemented and put into one of the capture registers. If it is a positive edge, data is put into LO8; if it is a negative edge, data is put into HI8. From that point, one of the edge detect status bits (CTR1, bits [1:0]) is set, and an interrupt can be generated if enabled (CTR0, bit 2). Meanwhile, T8 is loaded with FFh and starts counting again. If T8 reaches 0, the time-out status bit (CTR0, bit 5) is set, and an interrupt can be generated if enabled (CTR0, bit 1). T8 then continues counting from FFh (see Figure 23).

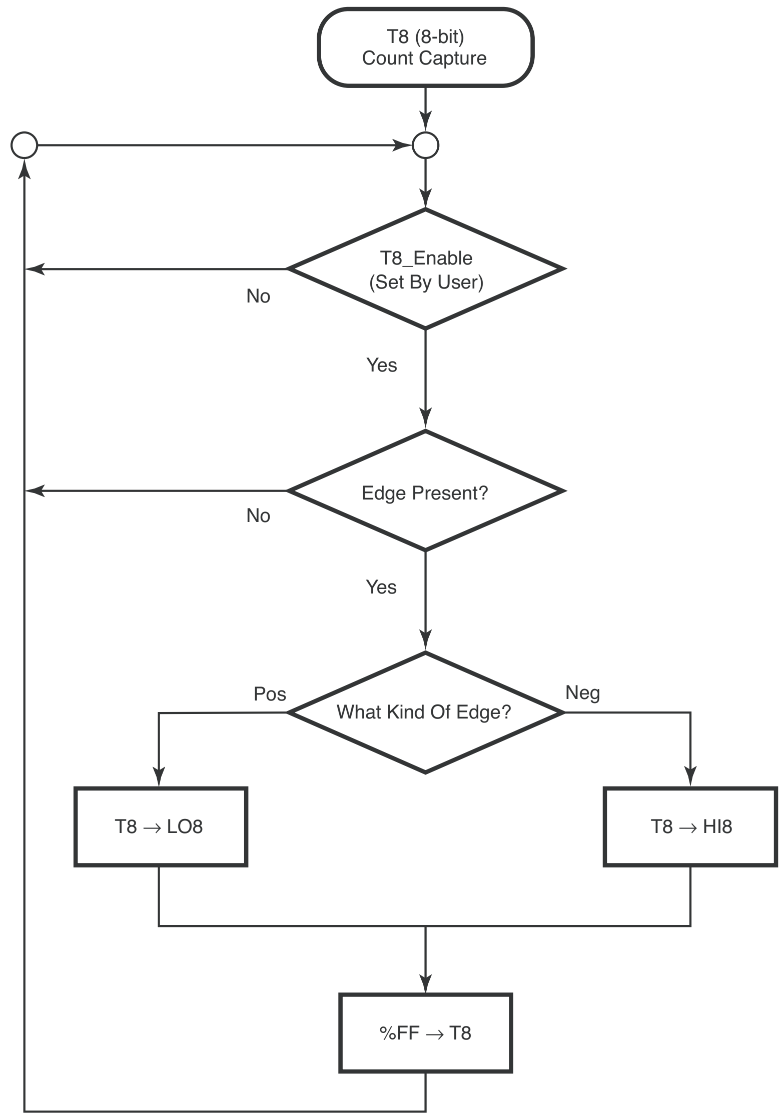


Figure 23. DEMODULATION Mode Count Capture Flowchart

When bit 4 of CTR3 is enabled, the flow of the demodulation sequence is altered. The third edge makes T8 active, and the fourth and fifth edges are captured. The capture interrupt is activated after the fifth event occurs. This mode is useful for capturing the carrier duty cycle as well as the frequency at which the first cycle is corrupted (see Figure 24 and Figure 25 on page 67).

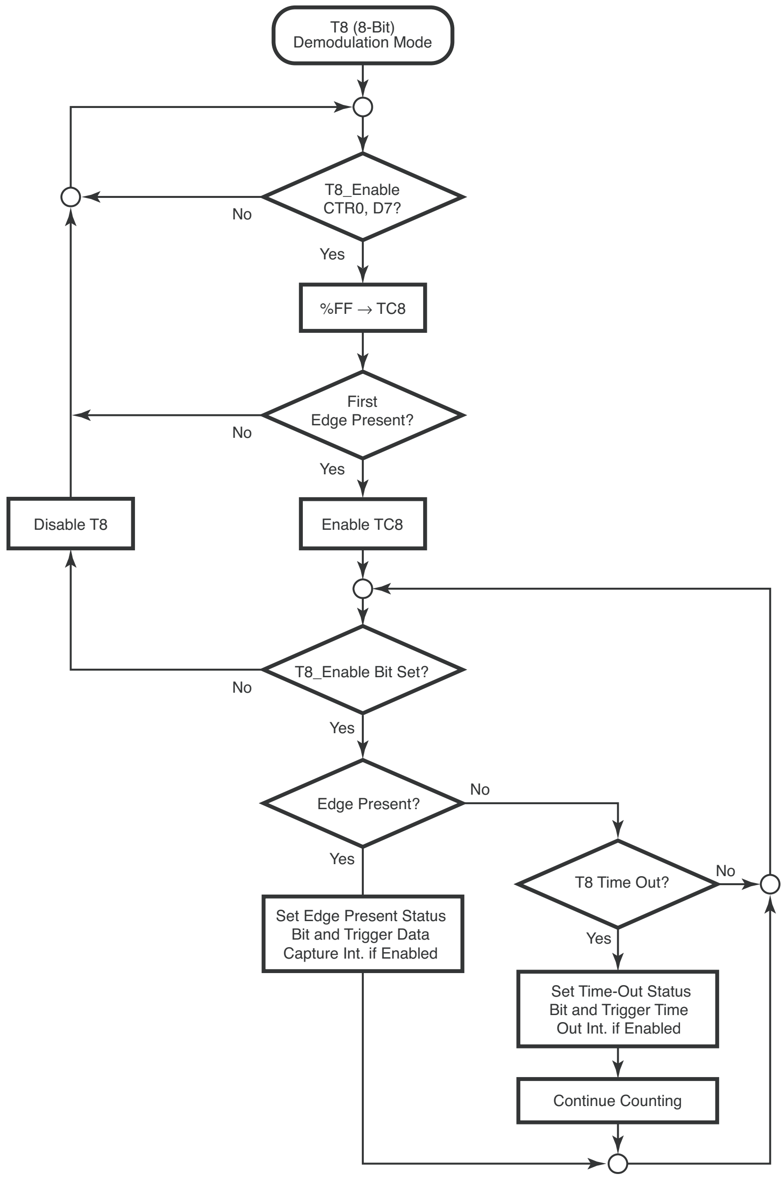


Figure 24. DEMODULATION Mode Flowchart

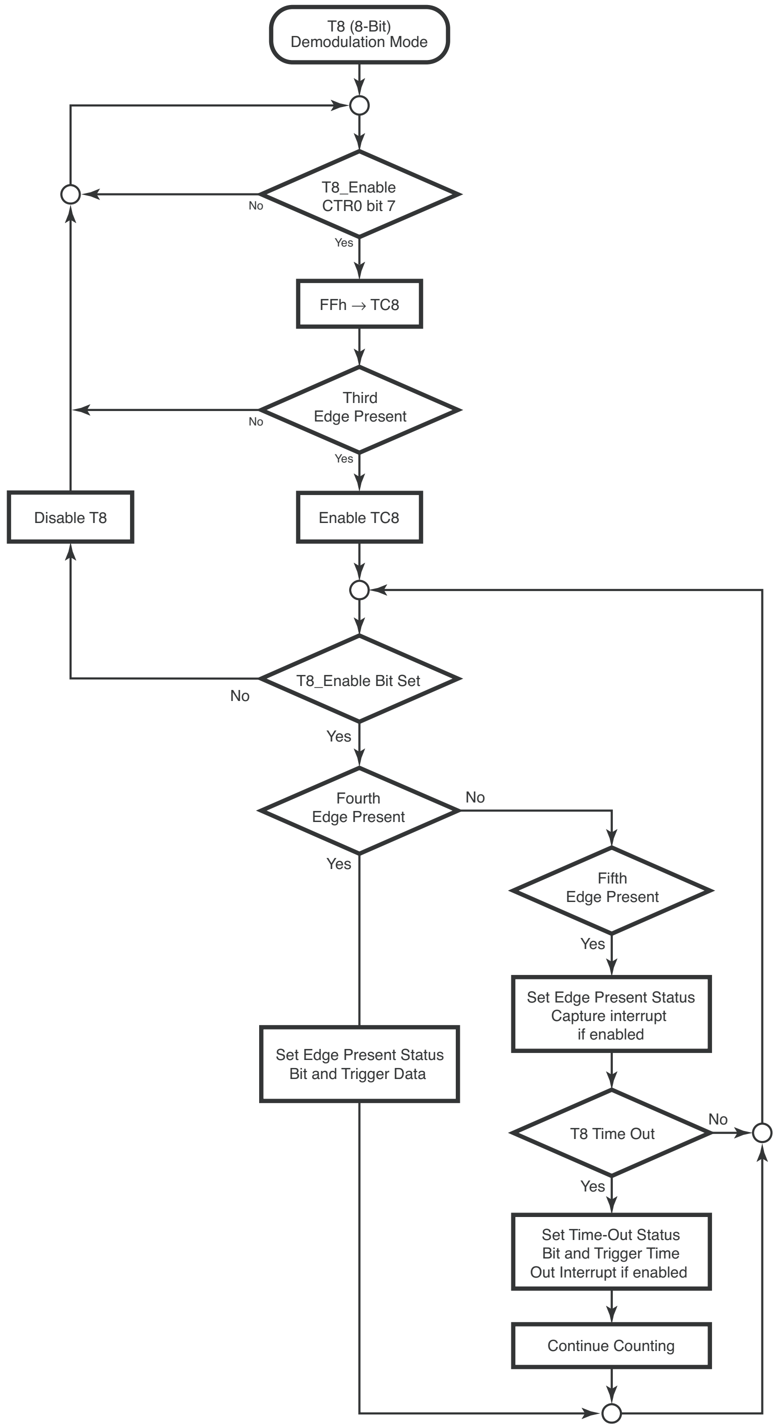


Figure 25. DEMODULATION Mode Flowchart with Bit 4 of CTR3 Set

T16 TRANSMIT Mode

In NORMAL or PING-PONG mode, the output of T16 when not enabled, is dependent on CTR1, bit 0. If it is a 0, T16_OUT is a 1; if it is a 1, T16_OUT is 0. You can force the output of T16 to either a 0 or 1 whether it is enabled or not by programming CTR1 bits [3:2] to a 10 or 11.

When bit 4 of CTR3 is set, the T16 output does not update. However, time-out interrupts (Flags) are still updated. In addition, the T8 carrier is not disrupted by timing out of the T16 timer.

When T16 is enabled, TC16H * 256 + TC16L is loaded, and T16_OUT is switched to its initial value (CTR1, bit 0). When T16 counts down to 0, T16_OUT is toggled (in NORMAL or PING-PONG mode), an interrupt (CTR2, bit 1) is generated (if enabled), and a status bit (CTR2, bit 5) is set. See Figure 26.

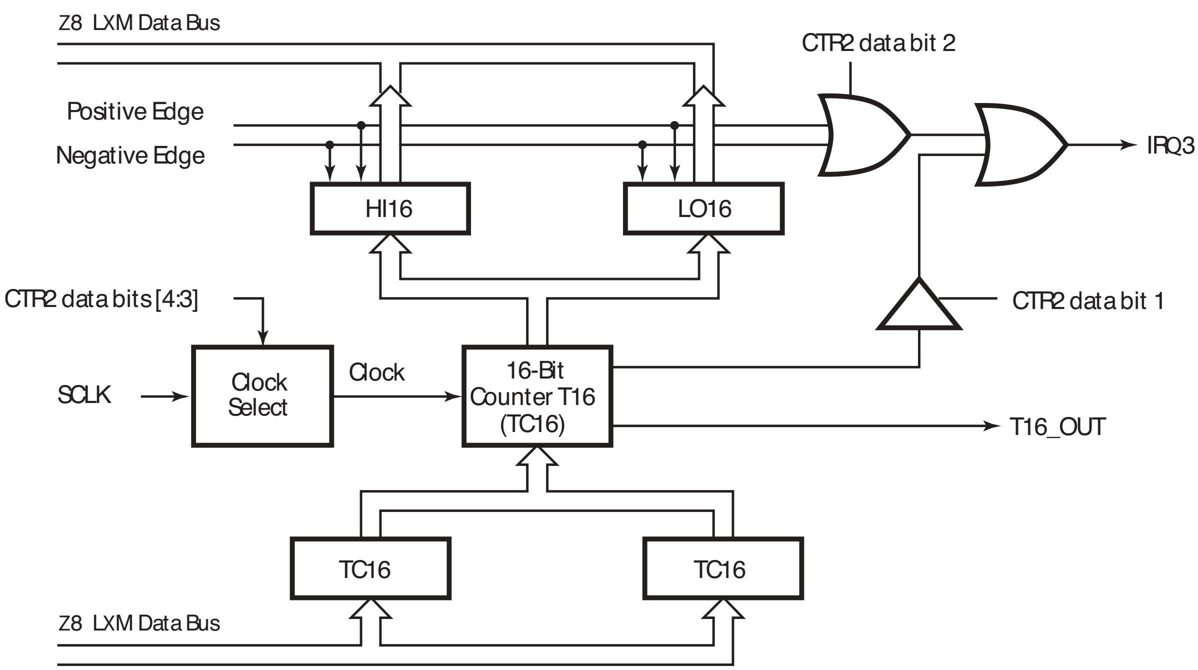


Figure 26. 16-Bit Counter/Timer Circuits

Note: *Global interrupts override this function as described in the* Interrupts on page 85.

If T16 is in SINGLE-PASS mode, it is stopped at this point (see Figure 27 on page 69). If it is in MODULO-N mode, it is loaded with TC16H * 256 + TC16L, and the counting continues (see Figure 28 on page 69).

You can modify the values in TC16H and TC16L at any time. The new values take effect when they are loaded.

Caution: *Do not load these registers at the time the values are to be loaded into the counter/timer to ensure known operation. An initial count of 1 is not allowed. An initial count of 0 causes T16 to count from 0 to* FFFEh. *Transition from 0 to* FFFFh *is not a time-out condition.*

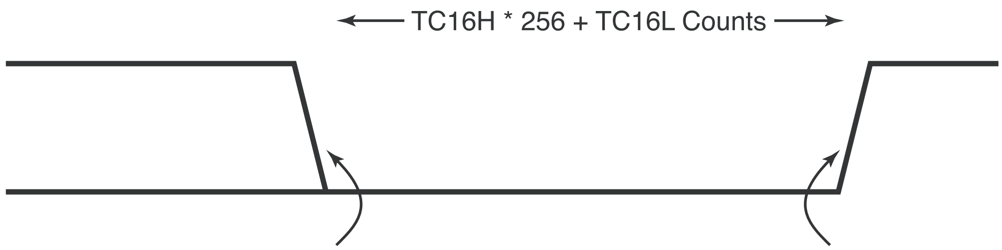


Figure 27. T16_OUT in SINGLE-PASS Mode

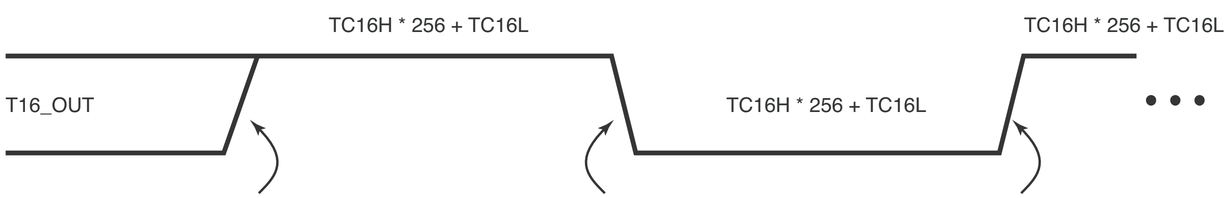


Figure 28. T16_OUT in MODULO-N Mode

T16 DEMODULATION Mode

You must program TC16L and TC16H to FFh. After T16 is enabled, and the first edge (rising, falling, or both depending on CTR1 bits [5:4]) is detected, T16 captures HI16 and LO16, reloads, and begins counting.

If Bit 6 of CTR2 Is 0—When a subsequent edge (rising, falling, or both depending on CTR1 bits [5:4]) is detected during counting, the current count in T16 is complemented and put into HI16 and LO16. When data is captured, one of the edge detect status bits

(CTR1, bit 1; bit 0) is set, and an interrupt is generated if enabled (CTR2, Bit 2). T16 is loaded with FFFFh and starts again.

This T16 mode is generally used to measure space time, the length of time between bursts of carrier signal (marks).

If Bit 6 of CTR2 Is 1—T16 ignores the subsequent edges in the input signal and continues counting down. A time-out of T8 causes T16 to capture its current value and generate an interrupt if enabled (CTR2, Bit 2). In this case, T16 does not reload and continues counting. If CTR2 bit 6 is toggled (by writing a 0 then a 1 to it), T16 captures and reloads on the next edge (rising, falling, or both depending on CTR1 bits [5:4]), continuing to ignore subsequent edges.

This T16 mode generally measures mark time, the length of an active carrier signal burst.

If T16 reaches 0, T16 continues counting from FFFFh. Meanwhile, a status bit (CTR2 bit 5) is set, and an interrupt time-out can be generated if enabled (CTR2 bit 1).

PING-PONG Mode

This operation mode is only valid in TRANSMIT mode. T8 and T16 must be programmed in SINGLE-PASS mode (CTR0, bit 6; CTR2, bit 6), and PING-PONG mode must be programmed in CTR1 bits [3:2]. You can begin the operation by enabling either T8 or T16 (CTR0, D7 or CTR2, D7). For example, if T8 is enabled, T8_OUT is set to this initial value (CTR1, bit 1). According to T8_OUT's level, TC8H or TC8L is loaded into T8. After the terminal count is reached, T8 is disabled, and T16 is enabled. T16_OUT then switches to its initial value (CTR1, bit 0), data from TC16H and TC16L is loaded, and T16 starts to count. After T16 reaches the terminal count, it stops, T8 is enabled again, repeating the entire cycle. Interrupts can be allowed when T8 or T16 reaches terminal control (CTR0, bit 1; CTR2, bit 1). To stop the Ping-Pong operation, write 00 to bits CTR1 bits [3:2]. See Figure 29 on page 71.

Note: *Enabling Ping-Pong operation while the counter/timers are running might cause intermittent counter/timer function. Disable the counter/timers and reset the status Flags before instituting this operation.*

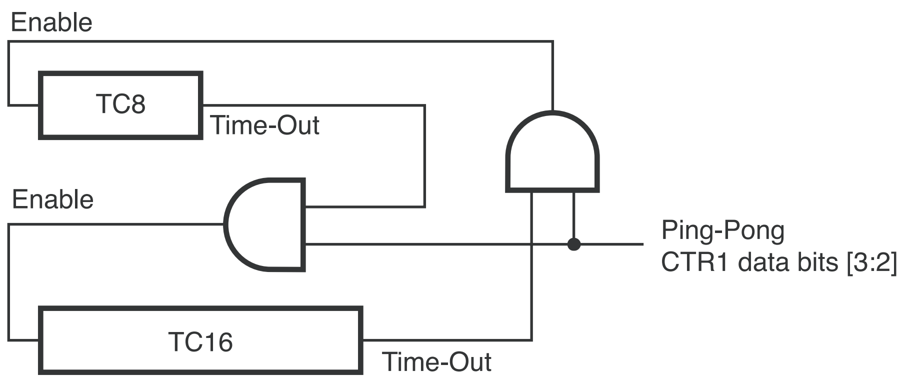


Figure 29. PING-PONG Mode Diagram

Initiating PING-PONG Mode

First, ensure that both counter/timers are not running. Set T8 into SINGLE-PASS mode (CTR0, bit 6), set T16 into SINGLE-PASS mode (CTR2, bit 6), and set the PING-PONG mode (CTR1 bits [3:2]). These instructions are not consecutive and can occur in random order. Finally, start PING-PONG mode by enabling either T8 (CTR0, D7) or T16 (CTR2, D7). The initial value of T8 or T16 must not be `1`. If you stop the timer and restart the timer, reload the initial value to avoid an unknown previous value.

During PING-PONG Mode

The enable bits of T8 and T16 (CTR0, D7; CTR2, D7) are set and cleared alternately by hardware. The time-out bits (CTR0, bit 5; CTR2, bit 5) are set every time the counter/timers reach the terminal count.

Timer Output

The output logic for the timers is displayed in Figure 30 on page 72. P34 is used to output T8_OUT when bit 0 of CTR0 is set. P35 is used to output the value of T16_OUT when bit 0 of CTR2 is set. When bit 6 of CTR1 is set, P36 outputs the logic combination of T8_OUT and T16_OUT via bits [4:5] of CTR1.

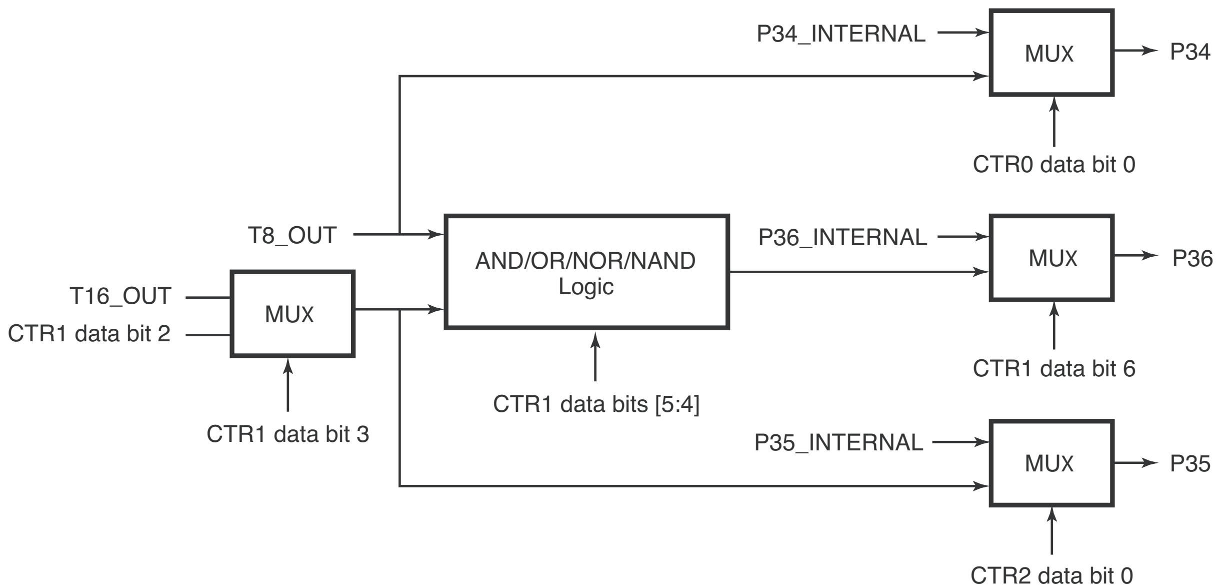


Figure 30. Output Circuit

Counter/Timer Registers

Timer 8 Capture High Register

The Timer 8 Capture High Register holds the captured data from the output of the 8-bit Counter/Timer 0. Typically, this register contains the number of counts when the input signal is 1 (Table 28).

Note: *This register is not reset after a SMR.*

Table 28. Timer 8 Capture High Register (HI8)

Bit	7	6	5	4	3	2	1	0
Field	T8_Capture_HI							
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Address	Bank D: 0Bh; Linear: D0Bh							

Bit Position	Value	Description
[7:0]	0hh–FFh	T8_Capture_HI—Reads return captured data. Writes have no effect.

Timer 8 Capture Low Register

The Timer 8 Capture Low Register holds the captured data from the output of the 8-bit Counter/Timer 0. Typically, this register contains the number of counts when the input signal is 0 (Table 29).

Note: *This register is not reset after a SMR.*

Table 29. Timer 8 Capture Low Register (L08)

Bit	7	6	5	4	3	2	1	0
Field	T8_Capture_LO							
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Address	Bank D: 0Ah; Linear: D0Ah							

Bit Position	Value	Description
[7:0]	0hh–FFh	T8_Capture_LO—Read returns captured data. Writes have no effect.

Timer 16 Capture High Register

The Timer 16 Capture High Register holds the captured data from the output of the 16-bit Counter/Timer 16. This register contains the most-significant byte (MSB) of the data (Table 30).

Note: *This register is not reset after a SMR.*

Table 30. Timer 16 Capture High Register (HI16)

Bit	7	6	5	4	3	2	1	0
Field	T16_Capture_HI							
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Address	Bank D: 09h; Linear: D09h							

Bit Position	Value	Description
[7:0]	0hh–FFh	T16_Capture_HI—Read returns captured data. Writes have no effect.

Timer 16 Capture Low Register

The Timer 16 Capture Low Register holds the captured data from the output of the 16-bit Counter/Timer 16. This register contains the LSB of the data (Table 31).

Note: *This register is not reset after a SMR.*

Table 31. Timer 16 Capture Low Register (L016)

Bit	7	6	5	4	3	2	1	0
Field	T16_Capture_LO							
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Address	Bank D: 08h; Linear: D08h							

Bit Position	Value	Description
[7:0]	0hh–FFh	T16_Capture_LO—Read returns captured data. Writes have no effect.

Counter/Timer 16 High Hold Register

The Counter/Timer 16 High Hold Register contains the high byte of the value loaded into the T16 timer (Table 32).

Note: *This register is not reset after a SMR.*

Table 32. Counter/Timer 16 High Hold Register (TC16H)

Bit	7	6	5	4	3	2	1	0
Field	T16_Data_HI							
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	Bank D: 07h; Linear: D07h							

Bit Position	Value	Description
[7:0]	0hh–FFh	T16_Data_HI—Read/Write Data.

Counter/Timer 16 Low Hold Register

The Counter/Timer 16 Low Hold Register contains the low byte of the value loaded into the T16 timer (Table 33).

Note: *This register is not reset after a SMR.*

Table 33. Counter/Timer 16 Low Hold Register (TC16L)

Bit	7	6	5	4	3	2	1	0
Field	T16_Data_LO							
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	Bank D: 06h; Linear: D06h							

Bit Position	Value	Description
[7:0]	0hh–FFh	T16_Data_LO—Read/Write Data.

Counter/Timer 8 High Hold Register

The Counter/Timer 8 High Hold Register contains the value to be counted while the T8 output is 1 (Table 34).

Note: *This register is not reset after a SMR.*

Table 34. Counter/Timer 8 High Hold Register (TC8H)

Bit	7	6	5	4	3	2	1	0
Field	T8_Level_HI							
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	Bank D: 05h; Linear: D05h							

Bit Position	Value	Description
[7:0]	0hh–FFh	T8_Level_HI—Read/Write Data.

Counter/Timer 8 Low Hold Register

The Counter/Timer 8 Low Hold Register contains the value to be counted while the T8 output is 0 (Table 35).

Note: *This register is not reset after a SMR.*

Table 35. Counter/Timer 8 Low Hold Register (TC8L)

Bit	7	6	5	4	3	2	1	0
Field	T8_Level_LO							
Reset	0	0	0	0	0	0	0	0
R/W	Bank D: 04h; Linear: D04h							
Address	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Value	Description
[7:0]	0hh–FFh	T8_Level_LO—Read/Write Data.

Counter/Timer 8 Control Register

The Counter/Timer 8 Control Register controls the timer function of the T8 timer. This Bank D register is described in Table 36.

Caution: *Writing a 1 to CTR0[5] is the only way to reset the Terminal Count status condition. Reset this bit before using/enabling the counter/timers.*

Note: *Ensure to manipulate CTR0, bit 5 and CTR1, bits 0 and 1 (DEMODULATION mode) when using the OR or AND commands. These instructions use a Read-Modify-Write sequence in which the current status from the CTR0 and CTR1 registers is ORed or ANDed with the designated value and then written back into the registers.*

Example: When the status of bit 5 is 1, a timer reset condition occurs.

Table 36. Counter/Timer 8 Control Register (CTR0)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	T8_Enable	Single/ Modulo-N	Time_Out	T8 _Clock		Capture_INT_Mask	Counter_INT_Mask	P34_Out
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	Bank D: 00h; Linear: D00h							

Bit Position	Value	Description
[7]		T8_Enable—Disable/enable the T8 counter.
	0	Disable counter.
	1	Enable counter. Configure T8 properly before enabling it.
[6]		SINGLE-PASS/MODULO-N
	0	MODULO-N mode—Counter reloads the initial value when terminal count is reached.
	1	SINGLE-PASS mode—Counter stops when the terminal count is reached.
[5]	Read	Time_Out—This bit is set when the T8 terminal count is reached.
	0	No counter time-out occurs.
	1	Counter time-out occurred.
	Write	
	0	No effect.
	1	Reset Flag to 0. Software must reset this Flag before using counter/timers.
[4:3]		T8 _Clock—Select the T8 input clock frequency. These bits are not reset upon Stop Mode Recovery.
	00	SCLK.
	01	SCLK ÷ 2.
	10	SCLK ÷ 4.
	11	SCLK ÷ 8.
[2]		Capture_INT_Mask—Disable/enable interrupt when data is captured into either LO8 or HI8 upon a positive or negative edge detection in DEMODULATION mode. This bit is not reset upon Stop Mode Recovery.
	0	Disable data capture interrupt.
	1	Enable data capture interrupt.

Bit Position	Value	Description
[1]		Counter_INT_Mask—Disable/enable T8 time-out interrupt. This bit is not reset upon Stop Mode Recovery.
	0	Disable time-out interrupt.
	1	Enable time-out interrupt.
[0]		P34_Out—Select normal I/O or T8 output function for Port 3, pin 4.
	0	P34 as port output.
	1	T8 output on P34.

T8 and T16 Common Functions Register

The T8 and T16 Common Functions Register (CTR1) controls the functions in common with Timer 8 and Timer 16. Table 37 describes the bits for this register.

Note: *Take care to differentiate* TRANSMIT *mode from* DEMODULATION *mode, as set by CTR1[7]. The functions of CTR1[6:0] and CTR2[6] are different depending on which mode is selected. Do not change from one mode to another without first disabling the counter/timers.*

Table 37. Timer 8 and Timer 16 Common Functions Register (CTR1)

Bit	7	6	5	4	3	2	1	0
Field	Mode	P36 Out/ Demodulator Input	T8/T16 Logic/ Edge Detect		Transmit Submode/ Glitch Filter		Initial Timer 8 Out/ Rising Edge	Initial Timer 16 Out/ Falling Edge
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	Bank D: 01h; Linear: D01h							

Bit Position	Description
[7]	Mode—Selects the timer mode for signal transmission or demodulation. 0 TRANSMIT mode. 1 DEMODULATION mode.
[6]	**TRANSMIT Mode** P36 Out—Select normal I/O or timer output on Port 3, Pin 6. 0 P36 acts as normal I/O port output. 1 P36 acts as combined Timer 8/Timer 16 output.
	DEMODULATION Mode Demodulator Input—Select Port 2, Pin 0 or Port 3, Pin 1 as the counter/timer input. 0 P31 acts as the demodulator input. If IMR[2] = 1, a P31 event can also generate an IRQ1 interrupt. To prevent this, clear IMR[2] or select P20 as input instead. 1 P20 acts as the demodulator input.
[5:4]	**TRANSMIT Mode** T8/T16 Logic—Defines how the Timer 8/Timer 16 outputs are combined logically. These bits are not reset upon Stop Mode Recovery. 00 Output is T8 AND T16. 01 Output is T8 OR T16. 10 Output is T8 NOR T16. 11 Output is T8 NAND T16.
	DEMODULATION Mode Edge Detect—Define the behavior of the edge detector. 00 Falling edge detection. 01 Rising edge detection. 10 Falling and rising edge detection. 11 Reserved.

Bit Position	Description
[3:2]	**TRANSMIT Mode** Submode Selection—Select normal or PING-PONG mode operation, or force T16 output. When these bits are written to 00b (NORMAL mode) or 01b (PING-PONG mode), T16_OUT assumes the opposite state of bit CTR1[0] until the timer begins counting. 00 Normal operation. Writing 00 terminates PING-PONG mode, if it is active. 01 PING-PONG mode. 10 Force T16_OUT = 0. 11 Force T16_OUT = 1.
	DEMODULATION Mode Glitch Filter—Define the maximum glitch width to be rejected by the counter/timer. 00 No filter. 01 4 SCLK cycle filter. 10 8 SCLK cycle filter. 11 Reserved.
[1]	**TRANSMIT Mode** Initial Timer 8 Out—Select the initial T8_OUT state when Timer 8 is enabled. While the timer is disabled, the opposite state is asserted on the pin to ensure that a transition occurs when the timer is enabled. Changing this bit while the counter is enabled can cause unpredictable output on T8_OUT. 0 T8_OUT transitions from High to Low when Timer 8 is enabled. 1 T8_OUT transitions from Low to High when Timer 8 is enabled.
	DEMODULATION Mode Rising Edge—Indicates whether a rising edge was detected on the input signal. Write 1 to this Flag to reset it. Read 0 No rising edge detection. 1 Rising edge detection. Write 0 No effect. 1 Reset Flag to 0.

Bit Position	Description
[0]	**TRANSMIT Mode** Initial Timer 16 Out—In NORMAL or PING-PONG mode, this bit selects the initial T16_OUT state when Timer 16 is enabled. While the timer is disabled, the opposite state is asserted on the pin to ensure that a transition occurs when the timer is enabled. Changing this bit while the counter is enabled can cause unpredictable output on T16_OUT. 0 If CTR1[3]=0, T16_OUT transitions from High to Low when Timer 16 is enabled. 1 If CTR1[3]=0, T16_OUT transitions from Low to High when Timer 16 is enabled.
	DEMODULATION Mode Falling Edge—Indicates whether a falling edge was detected on the input signal. Write 1 to this Flag to reset it. Read 0 No falling edge detection. 1 Falling edge detection. Write 0 No effect. 1 Reset Flag to 0.

Timer 16 Control Register

Table 38 describes the bits for the Timer 16 Control Register (CTR2).

Table 38. Counter/Timer 16 Control Register (CTR2)

Bit	7	6	5	4	3	2	1	0
Field	T16_Enable	Single/ Modulo-N	Time_Out	T16 _Clock		Capture_INT _Mask	Counter_INT _Mask	P35_Out
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	Bank D: 02h; Linear: D02h							

Bit Position	Description
[7]	T16_Enable—Disable/enable the T16 counter. 0 Disable T16 counter. 1 Enable T16 counter.
[6]	**TRANSMIT Mode (CTR1[7]=0)** Single/Modulo-N—Selects Timer 16 terminal count action. 0 MODULO-N mode. T16 reloads the initial value when terminal count is reached 1 SINGLE-PASS mode. T16 stops when the terminal count is reached
	DEMODULATION Mode (CTR1[7]=1) Enable single-edge capture. See T16 DEMODULATION Mode on page 69. 0 Timer 16 captures and reloads on all edges. 1 Timer 16 captures and reloads on first edge only.
[5]	Time_Out—This bit is set when the T16 terminal count is reached. Read Time_Out—This bit is set when the T16 terminal count is reached. 0 No counter time-out occurs. 1 Counter time-out occurred. Write 0 No effect. 1 Reset Flag to 0. Software must reset this Flag before using counter/timers.
[4:3]	T16 _Clock—Select T16 input clock frequency. These bits are not reset upon Stop Mode Recovery. 00 SCLK. 01 SCLK ÷ 2. 10 SCLK ÷ 4. 11 SCLK ÷ 8.

Bit Position	Description
[2]	Capture_INT_Mask—Disable/enable interrupt when data is captured into either LO16 or HI16 upon a positive or negative edge detection in DEMODULATION mode. This bit is not reset upon Stop Mode Recovery. 0 Disable data capture interrupt. 1 Enable data capture interrupt.
[1]	Counter_INT_Mask—Disable/enable T16 time-out interrupt. 0 Disable T16 time-out interrupt. 1 Enable T16 time-out interrupt.
[0]	P35_Out—Select normal I/O or T8 output function for Port 3, pin 5. 0 P35 as port output. 1 P35 is T16 output.

Timer 8/Timer 16 Control Register

The Timer 8/Timer 16 Counter/Timer Register allows the T8 and T16 counters to be synchronized. It also can freeze the T16 output value and change T8 DEMODULATION mode to capture one cycle of a carrier. Table 39 briefly describes the bits for this Bank D register. A description of each bit follows the table.

Table 39. Timer 8/Timer 16 Control Register (CTR3)

Bit	7	6	5	4	3	2	1	0
Field	T16_Enable	T8_Enable	Sync_Mode	T16_Out Disable	T8 Demodulate	Reserved		
Reset	0	0	0	0	0	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	—	—	—
Address	Bank D: 03h; Linear: D03h							

Bit Position	Value	Description
[7]	0 1	Disable T16 counter. Enable T16 counter. Configure T16 properly before enabling it.
[6]	0 1	Disable T8 counter. Enable T8 counter.

Bit Position	Value	Description
[5]		SYNC Mode—When enabled, the first pulse of Timer 8 (the carrier) is always synchronized with Timer 16 (the demodulated signal). It can always provide a full carrier pulse. This bit is not reset upon Stop Mode Recovery.
	0	Disable SYNC mode.
	1	Enable SYNC mode.
[4]		T16_Out Disable—Set this bit to disable toggling of the Timer 16 output. Time-out interrupts are still generated. This bit is not reset upon Stop Mode Recovery.
	0	T16 toggles normally.
	1	T16 toggle is disabled.
[3]		T8 Demodulate—(Capture one cycle.) This bit is not reset upon Stop Mode Recovery.
	0	T8 captures events normally.
	1	T8 becomes active on the third edge, captures events on the fourth and fifth edges, and generates an interrupt on the fifth edge. After a T8 time-out the event count resets to 0 and the fourth and fifth edges are captured again.
[2:0]		Reserved—Always reads 111b. Writes have no effect.

Interrupts

The Crimzon® ZLP12840 MCU features six different interrupts (see Table 41 on page 87). The interrupts are maskable and prioritized (see Figure 31 on page 86). The six sources are divided as follows: three sources are claimed by Port 3 lines P33:P31, two by the counter/timers and one for low-voltage detection. P32 and the UART receiver share the same interrupt. Only one interrupt can be selected as a source. When the UART receiver is enabled P32 is no longer used as an interrupt source. The UART transmit interrupt and UART baud rate interrupt use the same interrupt as the P33 interrupt. You must select the source that triggers the interrupt. When bit 7 of UTCL is 1, the UART transmit interrupt is the source. When bit 7 of UCTL is 0 and bit 5 of UCTL is 1, the BRG interrupt is selected. The Interrupt Mask Register (globally or individually) enables or disables the six interrupt requests.

The source for IRQ1 is determined by bit 1 of the Port 3 Mode Register (P3M) and bit 4 of the SMR4 register. If P3M[1]=0 (DIGITAL mode) and SMR4[4]=0, pin P33 is the IRQ1 source. If P3M[1]=1 (ANALOG mode) or SMR4[4]=1 (SMR interrupt enabled), the output of the Stop Mode Recovery source logic is used as the source for the interrupt. See Stop Mode Recovery Interrupt on page 99.

Table 40. Interrupt Control Registers

Address (Hex)						
12-Bit	Bank	8-Bit	Register Description	Mnemonic	Reset	Page No
0F9	All	F9	Interrupt Priority Register	IPR	XXh	90
0FA	All	FA	Interrupt Request Register	IRQ	00h	92
0FB	All	FB	Interrupt Mask Register	IMR	0XXX_XXXXb	89

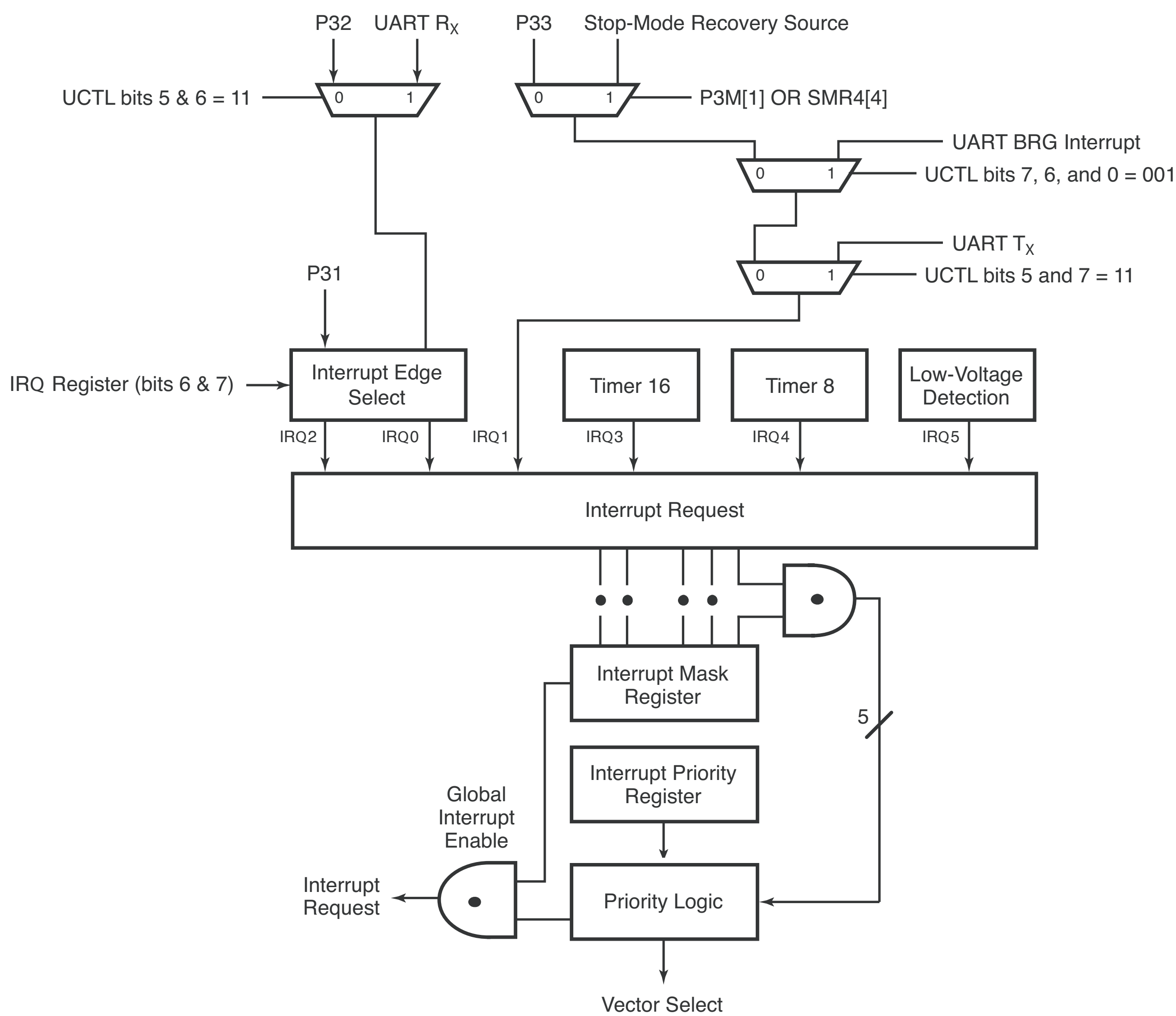


Figure 31. Interrupt Block Diagram

Table 41. Interrupt Types, Sources, and Vectors

Name	Source	Vector Location (Program Memory)	Comments
IRQ0	P32, UART Rx	0,1	External (P32), Rising, Falling Edge Triggered
IRQ1	P33, UART Tx, BRG, SMR Event	2,3	External (P33), Falling Edge Triggered
IRQ2	P31	4,5	External (P31), Rising, Falling Edge Triggered
IRQ3	Timer 16	6,7	Internal
IRQ4	Timer 8	8,9	Internal
IRQ5	Low-Voltage Detection	10,11	Internal

When more than one interrupt is pending, priorities are resolved by a programmable priority encoder controlled by the Interrupt Priority Register. An interrupt machine cycle activates when an interrupt request is granted. As a result, all subsequent interrupts are disabled, and the Program Counter and Status Flags are saved. The cycle then branches to the Program Memory vector location reserved for that interrupt. All Crimzon ZLP12840 MCU interrupts are vectored through locations in the program memory. This memory location and the next byte contain the 16-bit address of the interrupt service routine for that particular interrupt request. To accommodate polled interrupt systems, interrupt inputs are masked, and the Interrupt Request Register is polled to determine which of the interrupt requests require service.

An interrupt resulting from AN1 is mapped into IRQ2, and an interrupt from AN2 is mapped into IRQ0. Interrupts IRQ2 and IRQ0 can be rising, falling, or both edge triggered. These interrupts are programmable. The software can poll to identify the state of the pin.

Programming bits for the Interrupt Edge Select are located in the IRQ Register (R250), bits D7 and bit 6. The configuration is indicated in Table 42.

Table 42. Interrupt Request Register

IRQ Bit		Interrupt Edge	
7	**6**	**IRQ2 (P31)**	**IRQ0 (P32)**
0	0	F	F
0	1	F	R
1	0	R	F
1	1	R/F	R/F

Note: F = Falling Edge; R = Rising Edge.

Interrupt Priority Register

The Interrupt Priority Register (Table 43) defines which interrupts hold the highest priority. Interrupts are divided into three groups of two—Group A, Group B, and Group C.

IPR bits 4, 3, and 0 determine which interrupt group has priority. For example, if interrupts IRQ5, IRQ1, and IRQ0 occur simultaneously when IPR[4:3, 0]=001b, the interrupts are serviced in the following order: IRQ1, IRQ0, and IRQ5.

IPR bits 5, 2, and 1 determine which interrupt within each group has higher priority.

Table 43. Interrupt Priority Register (IPR)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved		Group A Priority	Group Priority [2:1]		Group B Priority	Group C Priority	Group Priority [0]
Reset	X	X	X	X	X	X	X	X
R/W	—		W	W		W	W	W
Address	Bank Independent: F9h; Linear: 0F9h							

Bit Position	Value	Description
[7:6]	—	Reserved Reads are undefined; writes must be 00b.
[5]	 0 1	Group A Priority (IRQ3, IRQ5) IRQ5 > IRQ3 IRQ3 > IRQ5
{[4:3], [0]}	 000 001 010 011 100 101 110 111	Group Priority Reserved C > A > B A > B > C A > C > B B > C > A C > B > A B > A > C Reserved
[2]	 0 1	Group B Priority (IRQ0, IRQ2) IRQ2 > IRQ0 IRQ0 > IRQ2
[1]	 0 1	Group C Priority (IRQ1, IRQ4) IRQ1 > IRQ4 IRQ4 > IRQ1

Interrupt Request Register

Bits 7 and 6 of the Interrupt Request Register are used to configure the edge detection of the interrupts for Port 3, bit 1 and Port 3, bit 2. The remaining bits, 5 through 0, indicate the status of the interrupt. When an interrupt is serviced, the hardware automatically clears the bit to 0. Writing a 1 to any of these bits generates an interrupt if the appropriate bits in the Interrupt Mask Register are enabled. Writing a 0 to these bits clears the interrupts (Table 44).

Table 44. Interrupt Request Register (IRQ)

Bit	7	6	5	4	3	2	1	0
Field	Interrupt Edge		IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
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	Bank Independent: FAh; Linear: 0FAh							

Bit Position	Value	Description
[7:6]		Interrupt Edge
	00	P31↓ P32↓
	01	P31↓ P32↑
	10	P31↑ P32↓
	11	P31↑↓ P32↑↓
[5]	Read	IRQ5 (Low-Voltage Detection)
	0	Interrupt did not occur.
	1	Interrupt occurred.
	Write	
	0	Clear interrupt.
	1	Set interrupt.
[4]	Read	IRQ4 (T8 Counter)
	0	Interrupt did not occur.
	1	Interrupt occurred.
	Write	
	0	Clear interrupt.
	1	Set interrupt.
[3]	Read	IRQ3 (T16 Counter)
	0	Interrupt did not occur.
	1	Interrupt occurred.
	Write	
	0	Clear interrupt.
	1	Set interrupt.

Bit Position	Value	Description
[2]	Read	IRQ2 (Port 3 Bit 1 Input)
	0	Interrupt did not occur.
	1	Interrupt occurred.
	Write	
	0	Clear interrupt.
	1	Set interrupt.
[1]	Read	IRQ1 (Port 3 Bit 3 Input/SMR Event/UART T_X/UART BRG)
	0	Interrupt did not occur.
	1	Interrupt occurred.
	Write	
	0	Clear interrupt.
	1	Set interrupt.
[0]	Read	IRQ0 (Port 3 Bit 2 Input/UART R_X)
	0	Interrupt did not occur.
	1	Interrupt occurred.
	Write	
	0	Clear interrupt.
	1	Set interrupt.

Note: *The IRQ register is protected from change until an EI instruction is executed once.*

Interrupt Mask Register

Bits [5:0] are used to enable the interrupt. Bit 7 is the status of the master interrupt. When reset, all interrupts are disabled. When writing a 1 to bit 7, you must also execute the EI instruction to enable interrupts (Table 45).

Table 45. Interrupt Mask Register (IMR)

Bit	7	6	5	4	3	2	1	0
Field	Master Interrupt Enable	Reserved	IRQ5 Enable	IRQ4 Enable	IRQ3 Enable	IRQ2 Enable	IRQ1 Enable	IRQ0 Enable
Reset	0	X	X	X	X	X	X	X
R/W	R/W	—	R/W	R/W	R/W	R/W	R/W	R/W
Address	Bank Independent: FBh; Linear: 0FBh							

Bit Position	Value	Description
[7]		Master Interrupt Enable Use only the DI and EI instructions to alter this bit. Always disable interrupts (DI instruction) before writing this register.
	0	All interrupts are disabled.
	1	Interrupts are enabled/disabled individually in bits [5:0].
[6]	0	Reserved Reads are undefined; writes must be 0.
[5]	0 1	Disables IRQ5. Enables IRQ5.
[4]	0 1	Disables IRQ4. Enables IRQ4.
[3]	0 1	Disables IRQ3. Enables IRQ3.
[2]	0 1	Disables IRQ2. Enables IRQ2.
[1]	0 1	Disables IRQ1. Enables IRQ1.
[0]	0 1	Disables IRQ0. Enables IRQ0.

Clock

The device's on-chip oscillator has a high-gain, parallel-resonant amplifier, for connection to a crystal, ceramic resonator, or any suitable external clock source (XTAL1 = Input, XTAL2 = Output). The crystal must be AT cut, 1 MHz to 8 MHz maximum, with a series resistance (RS) less than or equal to 100 Ω. The on-chip oscillator can be driven with a suitable external clock source.

The crystal must be connected across XTAL1 and XTAL2 using the recommended capacitors from each pin to ground. The typical capacitor value is 10 pF for 8 MHz. Also check with the crystal supplier for the optimum capacitance.

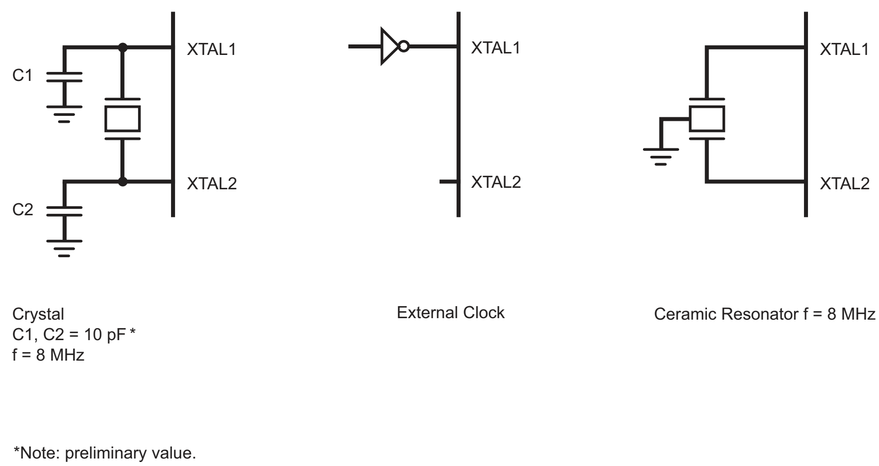


Figure 32. Oscillator Configuration

Zilog® IR MCU supports crystal, resonator, and oscillator. Most resonators have a frequency tolerance of less than ±0.5%, which is enough for remote control application. Resonator has a very fast startup time, which is around few hundred microseconds. Most crystals have a frequency tolerance of less than 50 ppm (±0.005%). However, crystal needs longer startup time than the resonator. The large loading capacitance slows down the oscillation startup time. Zilog suggests not to use more than 10 pF loading capacitor for the crystal. If the stray capacitance of the PCB or the crystal is high, the loading capacitance C1 and C2 must be reduced further to ensure stable oscillation before the T_{POR} (Power-On Reset time is typically 5–6 ms, see Table 62 on page 132).

For Stop Mode Recovery operation, bit 5 of SMR register allows you to select the Stop Mode Recovery delay, which is the T_{POR}. If Stop Mode Recovery delay is not selected, the MCU executes instruction immediately after it wakes up from the STOP mode. If resonator or crystal is used as a clock source then Stop Mode Recovery delay needs to be selected (bit 5 of SMR = 1).

For both resonator and crystal oscillator, the oscillation ground must go directly to the ground pin of the microcontroller. The oscillation ground must use the shortest distance from the microcontroller ground pin and it must be isolated from other connections.

Crystal 1 Oscillator Pin (XTAL1)

The Crystal 1 Oscillator time-based input pin connects a parallel-resonant crystal or ceramic resonator to the on-chip oscillator input. Additionally, an optional external single-phase clock can be connected to the on-chip oscillator input.

Crystal 2 Oscillator Pin (XTAL2)

The Crystal 2 Oscillator time-based output pin connects a parallel-resonant, crystal, or ceramic resonant to the on-chip oscillator output.

Internal Clock Signals (SCLK and TCLK)

The CPU and internal peripherals are driven by the internal SCLK signal during normal execution. During HALT mode, the interrupt logic is driven by the internal TCLK signal. These signals are produced by dividing the on-chip oscillator signal by a factor of two, and optionally by applying an additional divide-by-16 prescaler enabled in register bit SMR[0] (see Table 48 on page 102 and Figure 33).

Selecting the divide-by-16 prescaler reduces device power draw during normal operation and HALT mode. The prescaler is disabled by a Power-On Reset or Stop Mode Recovery.

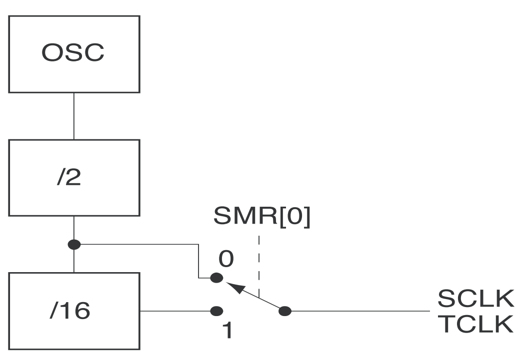


Figure 33. SCLK/TCLK Circuit

Resets and Power Management

The ZLP12840 provides the following reduced-power modes, power monitoring, and reset features:

- Power-On Reset—Starts the oscillator and internal clock and initializes the system to its power-on reset defaults.
- Voltage Brownout Standby—Stops the oscillator and internal clock if a low-voltage condition occurs. Initiates a Power-On Reset when power is restored.
- Voltage Detection—Optionally sets a Flag if a Low- or High-voltage condition occurs. The low-voltage detection Flag can generate an interrupt request, if enabled.
- HALT Mode—Stops the internal clock to the CPU until an enabled interrupt request is received.
- STOP Mode—Stops the clock and oscillator, reducing the MCU supply current to a very low level until a Power-On Reset or Stop Mode Recovery occurs.
- Stop Mode Recovery—Restarts the oscillator and internal clock and initializes most of the system to its power-on reset defaults. Some register values are not reset by a Stop Mode Recovery.
- Watchdog Timer—Optionally generates a Power-On Reset if the program fails to execute the WDT instruction within a specified time interval.

Note: *For supply current values under various conditions, see* DC Characteristics on page 129.

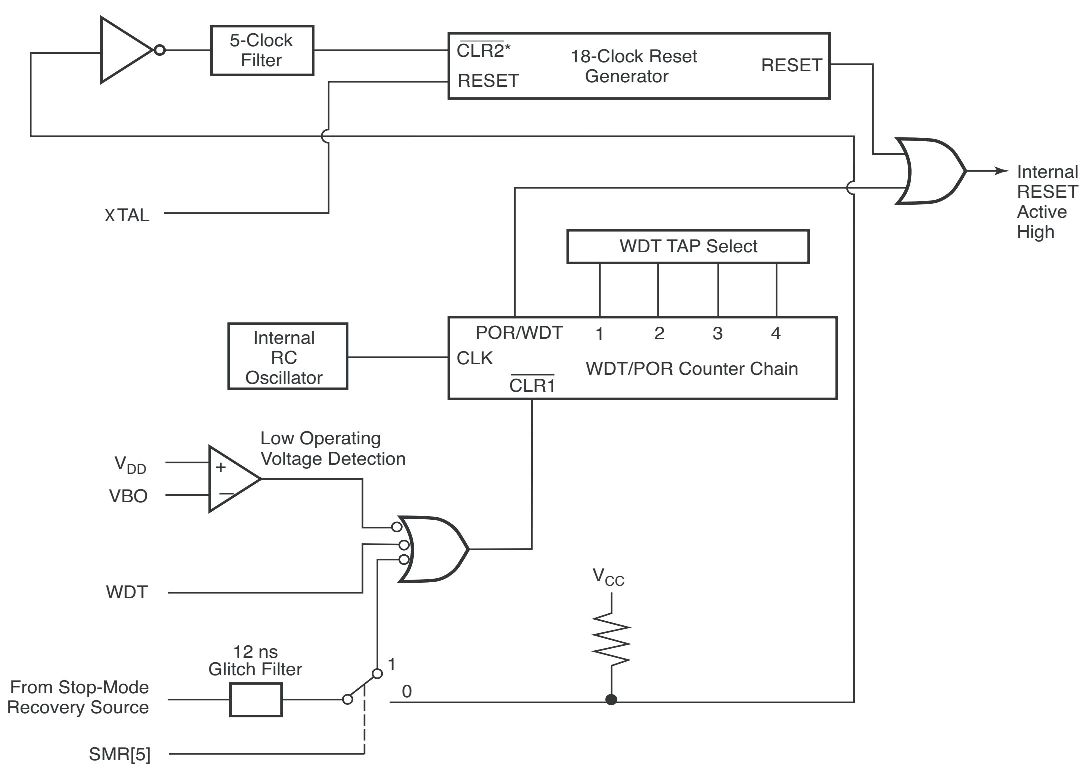
Figure 34 on page 96 displays the Power-On Reset sources. Table 46 lists control registers for reset and power management features. Some features are affected by registers described in other chapters.

Table 46. Reset and Power Management Registers

Address (Hex)						
12-Bit	**Bank**	**8-Bit**	**Register Description**	**Mnemonic**	**Reset**	**Page No**
D0C	D	0C	Low-Voltage Detection Register	LVD	1111_1000b	98
F0A	F	0A	Stop Mode Recovery Register 4	SMR4	XXX0_0000b	111
F0B	F	0B	Stop Mode Recovery Register	SMR	0010_0000b	102

Table 46. Reset and Power Management Registers (Continued)

Address (Hex)						
12-Bit	**Bank**	**8-Bit**	**Register Description**	**Mnemonic**	**Reset**	**Page No**
F0C	F	0C	Stop Mode Recovery Register 1	SMR1	00h	105
F0D	F	0D	Stop Mode Recovery Register 2	SMR2	X0X0_00XXb	107
F0E	F	0E	Stop Mode Recovery Register 3	SMR3	X0h	110
F0F	F	0F	Watchdog Timer Mode Register	WDTMR	XXXX_1101b	112



*CLR1 and CLR2 enable the WDT/POR and 18 Clock Reset timers, respectively, upon a Low-to-High input translation.

Figure 34. Resets and Watchdog Timer

Power-On Reset Timer

When power is initially applied to the device, a timer circuit clocked by a dedicated on-board RC-oscillator provides the Power-On Reset timer function.

The POR timer circuit is a one-shot timer that keeps the internal reset signal asserted long enough for V_{DD} and the oscillator circuit to stabilize before instruction execution begins.

The reset timer is triggered by one of three conditions:

- Initial power-on or recovery from a Voltage Brownout/standby condition.
- Stop Mode Recovery (if register bit SMR[5] = 1)
- Watchdog Timer time-out.

SMR[5] can be cleared to 0 to bypass the POR timer upon a Stop Mode Recovery. This should only be done when using an external clock that does not require a start-up delay.

Reset/Stop Mode Recovery Status

Read-only bit SMR[7]=0 if the previous reset was initiated by a power-on reset (including brown-out or WDT resets). SMR[7]=1 if the previous reset was initiated by a Stop Mode Recovery.

A power-on, brown-out, or WDT reset restores all registers to their Power-On Reset defaults. A Stop Mode Recovery restores most registers to their Power-On Reset defaults. Register bits not reset by a Stop Mode Recovery are highlighted in grey in the register tables. Register bit SMR[7] is set to 1 instead of reset by a Stop Mode Recovery.

Voltage Brownout/Standby

An on-chip Voltage Comparator checks that the V_{DD} is at the required level for correct operation of the device. Reset is globally driven when V_{DD} falls below V_{BO}. A small drop in V_{DD} causes the XTAL1 and XTAL2 circuitry to stop the crystal or resonator clock. If the V_{DD} is allowed to stay above V_{RAM}, the RAM content is preserved. When the power level is returned to above V_{BO}, the device performs a power-on reset and functions normally.

Voltage Detection

The Voltage Detection register (LVD, register `0Ch` at the expanded register bank `0Dh`) offers an option of monitoring the V_{CC} voltage. The Voltage Detection is enabled when bit 0 of LVD register is set. After Voltage Detection is enabled, the V_{CC} level is monitored in real time. The HVD Flag (bit 2 of the LVD register) is set only if V_{CC} is higher than $V_{HVD.}$ The LVD Flag (bit 1 of the LVD register) is set only if V_{CC} is lower than the V_{LVD}. When Voltage Detection is enabled, the LVD Flag also triggers IRQ5. The IRQ bit 5 latches the low voltage condition until it is cleared by instructions or reset. The IRQ5 interrupt is served if it is enabled in the IMR register. Otherwise, bit 5 of IRQ register is latched as a Flag only.

Note: *Do not modify register P01M while checking a low-voltage condition. Switching noise coming from Port 0 can trigger the LVD Flag.*

Voltage detection does not work in STOP mode. This register is described in Table 47.

Table 47. Low-Voltage Detection Register (LVD)

Bit	7	6	5	4	3	2	1	0
Field	Reserved					High Battery Detect	Low Battery Detect	Voltage Detect Enable
Reset	1	1	1	1	1	0	0	0
R/W	R	R	R	R	R	R	R	R/W
Address	Bank D: 0Ch; Linear: D0Ch							

Bit Position	R/W	Value	Description
[7:3]	—	—	Reserved—Reads 11111b. Writes have no effect.
[2]	R	0 1	HVD clear. High voltage detected. $V_{CC}>V_{HVD}$
[1]	R	0 1	LVD clear. Low voltage detected. $V_{CC}>V_{LVD}$
[0]	R/W	0 1	Voltage detection disabled. Voltage detection enabled.

HALT Mode

This instruction turns off the internal CPU clock, but not the XTAL oscillation. The counter/timers, UART, and interrupts IRQ0, IRQ1, IRQ2, IRQ3, IRQ4, and IRQ5 remain active. The devices are recovered by interrupts, either externally or internally generated. An interrupt request must be executed (enabled) to exit HALT mode. After the interrupt service routine, the program continues from the instruction after HALT mode.

To enter HALT mode, first flush the instruction pipeline to avoid suspending execution in mid-instruction. Execute a NOP (Op Code = `FFh`) immediately before the appropriate sleep instruction, as follows:

```
FF          NOP         ; clear the pipeline
7F          HALT        ; enter HALT mode
```

Power consumption during HALT mode can be reduced by first setting SMR[0]=1 to enable the divide-by-16 clock prescaler.

STOP Mode

This instruction turns OFF the internal clock and external crystal oscillation, reducing the MCU supply current to a very low level. For STOP mode current specifications, see DC Characteristics on page 129.

To enter STOP mode, first flush the instruction pipeline to avoid suspending execution in mid-instruction. Execute a NOP (Op Code = `FFh`) immediately before the appropriate sleep instruction, as follows:

```
FF          NOP         ; clear the pipeline
6F          STOP        ; enter STOP mode
```

STOP mode is terminated only by a reset, such as WDT time-out, POR, or one of the SMR events described in the following sections. This condition causes the processor to restart the application program at address `000Ch`.

Unlike a normal POR or WDT reset, a SMR reset does not reset the contents of some registers and bits. Register bits not reset by a SMR are highlighted in grey in the register tables. Register bit SMR[7] is set to 1 by a SMR.

Fast Stop Mode Recovery

SMR[5] can be cleared to 0 before entering STOP mode to bypass the default T_{POR} reset timer upon SMR. See Power-On Reset Timer on page 97. If SMR[5]=0, the SMR source must be kept active for at least 10 input clock periods (TpC).

Note: *SMR[5] must be set to 1 if using a crystal or resonator clock source. The T_{POR} delay allows the clock source to stabilize before executing instructions.*

Stop Mode Recovery Interrupt

Software can set register bit SMR4[4] = 1 to enable routing of SMR events to IRQ1 and to Port 3, pin 3. In this configuration, if an IRQ1 interrupt occurs, register bit P3[3] = 0 indicates that a SMR event is occurring.

Stop Mode Recovery Event Sources

Any Port 2 or 3 input pin can be configured to generate a SMR event, either individually or in a variety of logical combinations. The PartName provides the following registers for SMR source configuration and status:

- SMR Register—Selects one Port 3, pin 1–3 pin state or one of three Port 2 pin logical combinations to generate an event when a defined 0 or 1 level occurs.
- SMR1 Register—Configure one or more Port 2 input pins (0–7)to latch the latest read or write value and generate an event when the pin state changes.
- SMR2 Register—Selects one of seven Port 2 and 3 pin logical combinations to generate an event when a defined 0 or 1 level occurs.
- SMR3 Register—Configure one or more Port 3 input pins (0–3) to latch the latest read or write value and generate an event when the pin state changes.
- SMR4 Register—Enables routing of SMR events to IRQ1. Indicates whether port data has been latched for SMR1 or SMR3 event monitoring, and whether the latch was on a port read or write.

A SMR event occurs if any of the sources defined in the SMR, SMR1, SMR2, and SMR3 registers is active.

SMR Register Events

The SMR register function is similar to the standard SMR feature used in previous Z8 CPU-compatible parts. Register bits SMR[4:2] are set to select one of six event modes, as displayed in Figure 35 on page 101. The output of the corresponding logic is compared to the state of SMR[6]; when they are the same, a SMR event is generated.

If SMR[4:2]=000, no event source is selected by SMR. The state SMR[4:2]=001 is reserved and selects no event in this device.

The logic configured by the SMR register ignores any port pins that are configured as an output, or that are selected as source pins in registers SMR1 or SMR3. The SMR register is summarized in Table 48 on page 102.

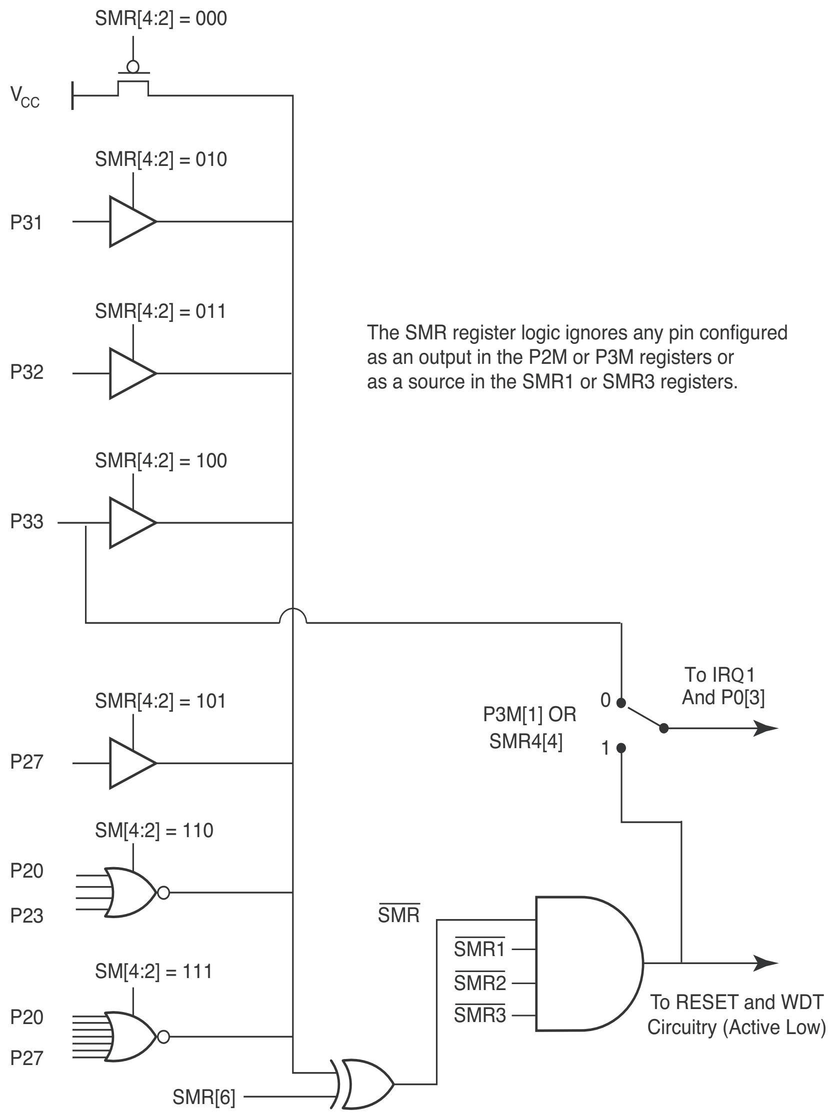


Figure 35. SMR Register-Controlled Event Sources

Table 48. Stop Mode Recovery Register (SMR)

Bit	7	6	5	4	3	2	1	0
Field	Stop Flag	Stop Mode Recovery Level	Stop Delay	Stop Mode Recovery Source			Reserved	SCLK/TCLK Divide-by-16
Reset	**0**	0	1	0	0	0	0	0
R/W	R	W	W	W	W	W	W	W
Address	Bank F: 0Bh; Linear: F0Bh							

Bit Position	Value	Description
[7]		Stop Flag—Indicates whether last startup was power-on Reset or Stop Mode Recovery. A write to this bit has no effect.
	0	Power-On Reset.
	1	Stop Mode Recovery.
[6]		Stop Mode Recovery Level—Selects whether an SMR[4:2]-selected SMR is initiated by a Low or High level at the XOR-gate input (see Figure 35 on page 101).
	0	Low.
	1	High.
[5]		Stop Delay—Controls the reset delay after recovery. Must be 1 if using a crystal or resonator clock source.
	0	Off.
	1	On.
[4:2]		Stop Mode Recovery Source—Specifies a Stop Mode Recovery wake-up source at the XOR gate input (see Figure 35 on page 101). This value is not changed by a Stop Mode Recovery. The following equations ignore any Port pin configured as output or selected in SMR1 or SMR3.
	000	No SMR register source selected.
	001	Reserved.
	010	P31.
	011	P32.
	100	P33.
	101	P27.
	110	Port 2 NOR 0–3.
	111	Port 2 NOR 0–7.
[1]	—	Reserved—Reads are undefined; must write 0.
[0]		SCLK/TCLK Divide-by-16 Select—Controls a divide-by-16 prescaler of the internal SCLK/TCLK signal (see Internal Clock Signals (SCLK and TCLK) on page 94). A Power-On Reset or Stop Mode Recovery clears this bit to 0.
	0	OFF.
	1	ON.

SMR1 Register Events

The SMR1 register can be used to configure one or more Port 2 pins to be to be compared to a written or sampled reference value and generate a SMR event when the pin state differs from the reference value.

To configure a Port 2 pin as an SMR1 event source, make sure it is configured as an input in the P2M register, then set the corresponding SMR1 register bit. By default, a SMR event occurs when the pin's state is zero.

After a Port 2 pin is configured as an SMR1 source, any subsequent read from or write to the P2 register latches the read or written value for reference. A SMR event occurs when the pin's state differs from the last reference value latched. The SMR1 source logic is displayed in Figure 36 on page 104.

The program can read register bits SMR4[1:0] to determine whether the Port 2 pins trigger a SMR on a change from the last read value (SMR4[1:0]=01), or on a change from the last written value (SMR4[1:0]=10). Software can clear SMR4[1:0] to 00 to restore the default behavior (configured pins trigger when their state is 0).

The SMR1 register is summarized in Table 49 on page 105.

After the following example code is executed, a 1 on P2 0 will wake the part from STOP mode.

```
LD P2M, #%FF     ;Set Port 2 to inputs.
SRP #%0F         ;Point to expanded bank F
LD SMR1, #%01    ;Select P20 for SMR1.
SRP #%00         ;Point to bank 0
LD P2, #%00      ;Write 00h to Port 2, so the P20 reference
                 ;value is 0, and a 1 on P20 wakes the part.
NOP
STOP
```

After the following example code is executed when the value of P2 is 00h, a 1 on P20 will wake the part from STOP mode:

```
LD P2M, #%FF     ;Set ports to inputs.
SRP #%0F         ;Point to expanded bank F
LD SMR1, #%01    ;Select P20 for SMR1.
SRP #%00         ;Point to bank 0
LD R6, P2        ;If a 0 is read from Port 2, the P20 reference
                 ;value is 0, so a 1 on P20 wakes the part.
NOP
STOP
```

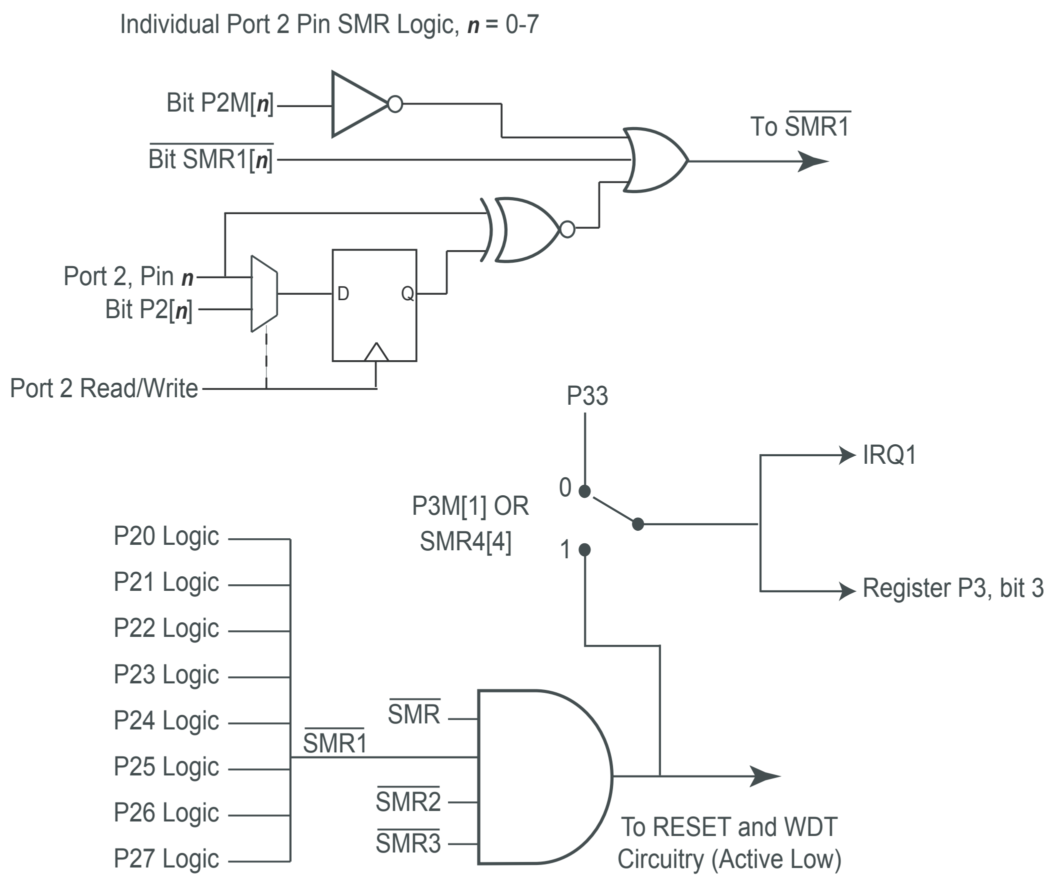


Figure 36. SMR1 Register-Controlled Event Sources

Table 49. Stop Mode Recovery Register 1 (SMR1)

Bit	7	6	5	4	3	2	1	0
Field	P27 Stop Select	P26 Stop Select	P25 Stop Select	P24 Stop Select	P23 Stop Select	P22 Stop Select	P21 Stop Select	P20 Stop Select
Reset	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W
Address	Bank F: 0Ch; Linear: F0Ch							

Bit Position	Value	Description
[7]	0 1	P27 not selected. P27 selected as an SMR source.
[6]	0 1	P26 not selected. P26 selected as an SMR source.
[5]	0 1	P25 not selected. P25 selected as an SMR source.
[4]	0 1	P24 not selected. P24 selected as an SMR source.
[3]	0 1	P23 not selected. P23 selected as an SMR source.
[2]	0 1	P22 not selected. P22 selected as an SMR source.
[1]	0 1	P21 not selected. P21 selected as an SMR source.
[0]	0 1	P20 not selected. P20 selected as an SMR source.

Note: *This register is not reset after a SMR.*

SMR2 Register Events

The SMR2 register function is similar to the standard SMR feature used in previous Z8 CPU-compatible parts. Register bits SMR2[4:2] are set to select one of seven event modes, as displayed in Figure 37. The output of the corresponding logic is compared to the state of SMR2[6]; when they are the same, a SMR event is generated. If SMR2[4:2]=000, no event source is selected by SMR2.

The logic configured by the SMR2 register ignores any port pins that are configured as an output, or that are selected as source pins in registers SMR1 or SMR3.

The SMR2 register is summarized in Table 50 on page 107.

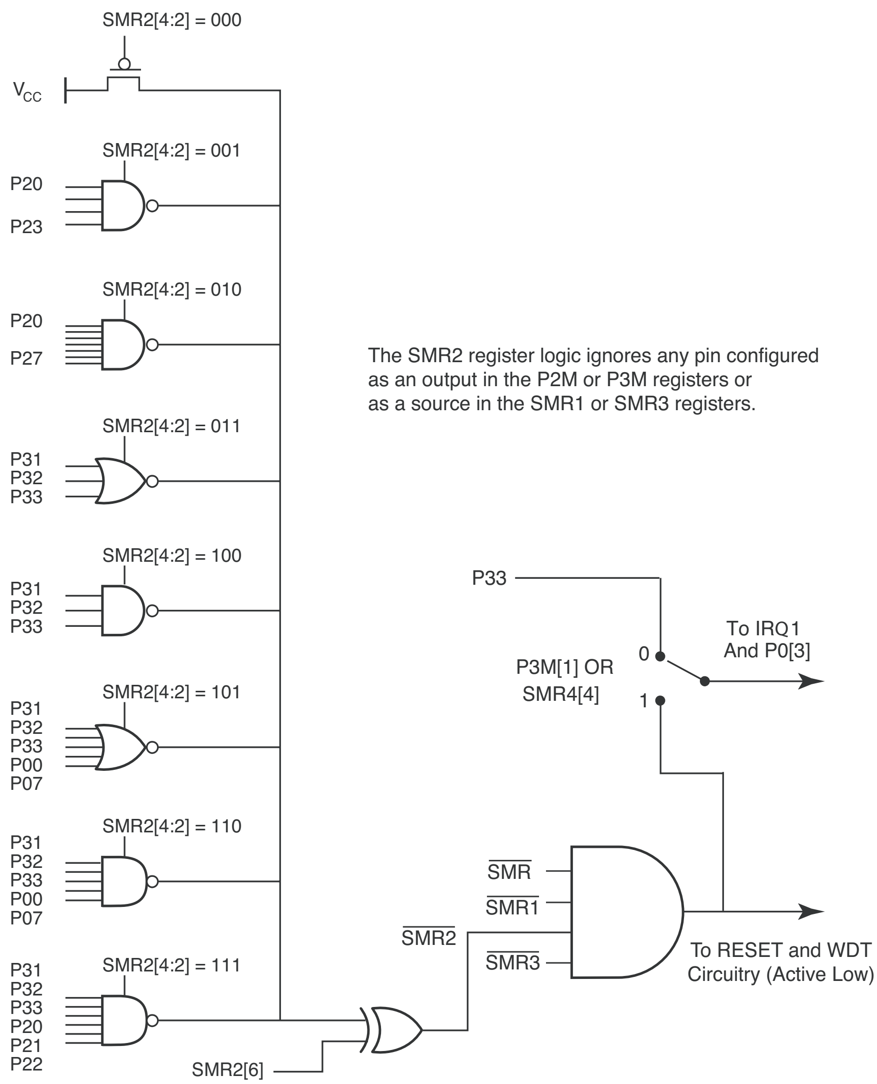


Figure 37. SMR2 Register-Controlled Event Sources

Table 50. Stop Mode Recovery Register 2 (SMR2)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Stop Mode Recovery Level 2	Reserved	Stop Mode Recovery Source			Reserved	
Reset	X	0	X	0	0	0	X	X
R/W	—	W	—	W	W	W	—	
Address	Bank F: 0Dh; Linear: F0Dh							

Bit Position	Value	Description
[7]	—	Reserved—Read is undefined; write must be 0.
[6]		Stop Mode Recovery Level 2 Selects whether an SMR2[4:2]-selected SMR is initiated by a Low or High level at the XOR-gate input (see Figure 37 on page 106).
	0	Low.
	1	High.
[5]	—	Reserved—Read is undefined; write must be 0.
[4:2]		Stop Mode Recovery Source Specifies a SMR wake-up source at the XOR gate input (see Figure 37 on page 106). Additional sources can be selected by SMR, SMR1, and SMR3 registers. If more than one source is selected, any selected source event causes a SMR. The following equations ignore any Port pin that is selected in register SMR1 or configured as an output.
	000	No SMR2 register source selected.
	001	NAND of P23:P20.
	010	NAND of P27:P20.
	011	NOR of P33:P31.
	100	NAND of P33:P31.
	101	NOR of P33:P31, P00, P07.
	110	NAND of P33:P31, P00, P07.
	111	NAND of P33:P31, P22:P20.
[1:0]	—	Reserved—Read is undefined; write must be 00b.

Note: *This register is not reset after a SMR.*

SMR3 Register Events

The SMR3 register can be used to configure one or more of Port 3, pins 0–3 to be compared to a written or sampled reference value and generate a SMR event when the pin state differs from the reference value.

To configure a Port 3 input pin as an SMR3 event source set the corresponding SMR3 register bit. By default, a SMR event occurs when the pin's state is zero.

After a Port 3 pin is configured as an SMR3 source, any subsequent read from or write to the P2 register latches the read or written value for reference. A SMR event occurs when the pin's state differs from the last reference value latched. The SMR3 source logic is displayed in Figure 38 on page 109.

The program can read register bits SMR4[3:2] to determine whether the Port 3 pins trigger a SMR on a change from the last read value (SMR4[3:2]=01), or on a change from the last written value (SMR4[3:2]=10). Software can clear SMR4[3:2] to 00 to restore the default behavior (configured pins trigger when their state is 0). The SMR3 register is summarized in Table 48 on page 102.

After the following example code is executed, a 1 on P30 will wake the part from STOP mode.

```
LD SMR3, #%01    ;Select P30 from SMR3.
LD P3, #%00      ;Write 00h to Port 3, so the P30 reference
                 ;value is 0, and a 1 on P30 wakes the part.
NOP
STOP
```

After the following example code is executed when the value of P3 is 00h, a 1 on P30 will wake the part from STOP mode.

```
LD SMR3, #%01    ;Select P30 for SMR3.
LD R6, P3        ;If a 0 is read from Port 3, the P30 reference
                 ;value is 0, so a 1 on P30 wakes the part.
NOP
STOP
```

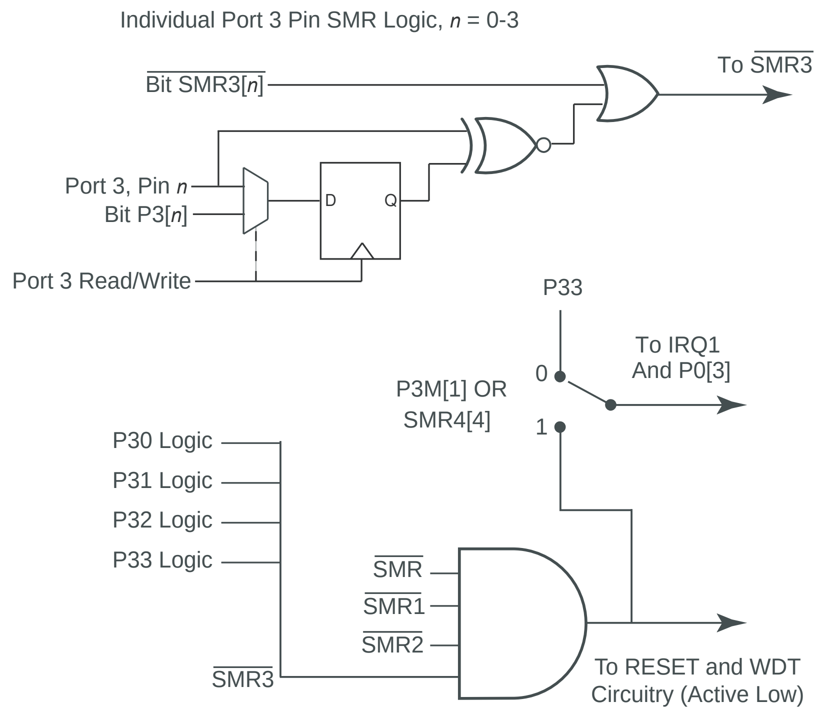


Figure 38. SMR3 Register-Controlled Event Sources

Table 51. Stop Mode Recovery Register 3 (SMR3)

Bit	7	6	5	4	3	2	1	0
Field	—				P33 SMR Select	P32 SMR Select	P31 SMR Select	P30 SMR Select
Reset	X	X	X	X	0	0	0	0
R/W	—	—	—	—	W	W	W	W
Address	Bank F: 0Eh; Linear: F0Eh							

Bit Position	Value	Description
[7:4]	—	Reserved—Reads undefined; writes have no effect.
[3]	0 1	P33 not selected. P33 SMR source selected.
[2]	0 1	P32 not selected. P32 SMR source selected.
[1]	0 1	P31 not selected. P31 SMR source selected.
[0]	0 1	P30 not selected. P30 SMR source selected.

Note: *This register is not reset after a SMR.*

Stop Mode Recovery Register 4

The Stop Mode Recovery Register 4 (SMR4) Register enables the SMR interrupt source and indicates the reference value status for registers SMR1 and SMR3.

Table 52. Stop Mode Recovery Register 4 (SMR4)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved			SMR IRQ Enable	Port 3 SMR Status		Port 2 SMR Status	
Reset	X	X	X	0	0	0	0	0
R/W	—	—	—	R/W	R/W	R/W	R/W	R/W
Address	Bank F: 0Ah; Linear: F0Ah							

Bit Position	Value	Description
[7:5]	—	Reserved—Reads are undefined; must write 000b.
[4]	 0 1	SMR IRQ Enable If P3M[1]=0, SMR events do not generate an interrupt. SMR events generate an interrupt on IRQ1.
[3:2]	 00 01 10 11	Port 3 SMR Status No Read or Write of the P3 register occurs. P3 Read occurs; used as SMR3 reference. P3 Write occurs; used as SMR3 reference. Reserved.
[1:0]	 00 01 10 11	Port 2 SMR Status No Read or Write of the P2 register occurs. P2 Read occurs; use P2 Read as SMR1 reference. P2 Write occurs; use P2 Write as SMR1 reference. Reserved.

Note: *This register is not reset after a SMR.*

Watchdog Timer

The Watchdog Timer is a retriggerable one-shot timer that resets the Z8 LXM CPU if it reaches its terminal count. The WDT must initially be enabled by executing the WDT instruction. On subsequent executions of the WDT instruction, the WDT is refreshed. The WDT circuit is driven by an on-board RC-oscillator. The WDT instruction affects the Zero (Z), Sign (S), and Overflow (V) Flags.

The POR clock source is the internal RC-oscillator. Bits 0 and 1 of the WDT register control a tap circuit that determines the minimum time-out period. Bit 2 determines whether the WDT is active during HALT, and bit 3 determines WDT activity during STOP mode. Bits 4 through 7 are reserved (see Table 53). This register is accessible only during the first 60 processor cycles (120 XTAL clocks) from the execution of the first instruction after Power-On Reset, Watchdog Timer Reset, or a SMR (see STOP Mode on page 99). After this point, the register cannot be modified by any means (intentional or otherwise). The WDTMR register cannot be read. The register is located in Bank F of the Expanded Register Group at address location `0Fh`.

Note: *This register is not reset after a SMR.*

Table 53. Watchdog Timer Mode Register (WDTMR)

Bit	7	6	5	4	3	2	1	0
Field	—				WDT During STOP Mode	WDT During HALT Mode	Time-Out Select	
Reset	X	X	X	X	1	1	0	1
R/W	X	X	X	X	W	W	W	W
Address	Bank F: 0Fh; Linear: F0Fh							

Bit Position	Value	Description
[7:4]	—	Reserved—Reads are undefined; must write 0000.
[3]		WDT During STOP Mode—Determines whether or not the WDT is active during STOP mode.
	0	Off.
	1	WDT active during STOP mode.
[2]		WDT During HALT Mode—Determines whether the WDT is active or not during HALT mode. See Figure 34 on page 96.
	0	Off.
	1	WDT active during HALT mode.

Bit Position	Value	Description
[1:0]		Time-Out Select—Selects the WDT time period.
	00	5 ms minimum.
	01	10 ms minimum.
	10	20 ms minimum.
	11	80 ms minimum.

Z8 LXM CPU Programming Summary

This chapter provides information for programming the Z8 LXM CPU included in this device. For details on the CPU and its instruction set, refer to *Z8 LXM CPU Core User Manual (UM0183)*.

Addressing Notation

Table 54 summarizes Z8 LXM CPU addressing modes and symbolic notation. The text variable *n* represents a decimal number; *aa* represents a hexadecimal address; and *LABEL* represents a label defined elsewhere in the assembly source.

In reference notation *only*, lowercase is used to distinguish 4-bit addressed working registers (r1, r2) from 8-bit addressed registers (R1, R2). The numerals 1 and 2, respectively, indicate whether the register is used for destination or source addressing.

Table 54. Symbolic Notation for Operands

Symbol	Assembly Operand	Description
cc	–	**Condition Code** cc represents a condition code mnemonic. See Condition Codes on page 119.
IM	#*n*	**Immediate Data** IM represents an Immediate Data value, prefixed by # in assembly language. The immediate value follows the instruction opcode in program memory. *n* = 0 to 255.
r1 r2	R*n*	**Working Register** r1 or r2 represents the name, R*n*, of a working register, where *n* = 0, 1, 2,..., 15. The equivalent 12-bit address is {RP[3:0], RP[7:4], *n*}.
rr1 rr2	RR*n*	**Working Register Pair** rr1 or rr2 represents the name, R*n*, of a working register pair, where *n* = 0, 2, 4,..., 14. The equivalent 12-bit address is {RP[3:0], RP[7:4], *n*}.
R1 R2	%*aa*	**Register** R1 or R2 represents an 8-bit register address. For addresses 00h–DFh or F0h–FFh, the equivalent 12-bit address is {RP[3:0], %*aa*}. For addresses E0h–EFh (escaped mode), the equivalent 12-bit address is {RP[3:0], RP[7:4], %*aa*[3:0]}.
RR1 RR2	%*aa*	**Register Pair (8-bit Address)** RR1 or RR2 represents the 8-bit address of a register pair. For addresses 00h–DFh or F0h–FFh, the equivalent 12-bit address is {RP[3:0], %*aa*}. For addresses E0h–EFh (escaped mode), the equivalent 12-bit address is {RP[3:0], RP[7:4], %*aa*[3:0]}.

Table 54. Symbolic Notation for Operands (Continued)

Symbol	Assembly Operand	Description
Irr1 Irr2	@R*n*	**Indirect Working Register** Ir1 or Ir2 represents the name a working register, R*n,* where *n* = 0, 1, 2,..., 15. @ indicates Indirect Working Register addressing using an 8-bit effective address contained in the specified working register. The accessed register's equivalent 12-bit address is {RP[3:0], *8-bit effective address*}.
Irr1 Irr2	@RR*n*	**Indirect Working Register Pair** Irr1 or Irr2 represents the name a working register pair, RR*n,* where *n* = 0, 2, 4,..., 14. @ indicates Indirect Working Register addressing using an effective address in the specified working register pair. Depending on the instruction, the effective address is in the register file (12-bit address) or program/constant memory (16-bit address).
IR1 IR2	@%*aa*	**Indirect Register** IR1 or IR2 represents the 8-bit address of a register. @ indicates Indirect Register addressing using an 8-bit effective address contained in the specified register. The accessed register's equivalent 12-bit address is {RP[3:0], *8-bit effective address*}.
IRR1	@%*aa*	**Indirect Register Pair** IRR1 represents the 8-bit address of a register. @ indicates Indirect Register addressing with a 16-bit effective address (in program memory) contained in the specified register pair.
X(r1) X(r2)	%*aa*(R*n*)	**Indexed (X) Addressing** X represents the 8-bit base address to which the offset is added. r1 or r2 represents the name, R*n*, of a working register containing the 8-bit signed offset. The 8-bit effective address is the sum of X and the contents of working register R*n*. The accessed register's equivalent 12-bit address is {RP[3:0], *8-bit effective address*}.
DA	*LABEL*	**Direct Address (JP, CALL)** In a JP or CALL operand, DA is a 16-bit program memory address in the range of `0000H` to `FFFFH`. DA replaces the contents of the Program Counter to cause execution to continue at a new location in Program Memory. In assembly source, the address is typically represented as a label.
RA	*LABEL*	**Relative Address (JR, DJNZ)** RA is a signed 8-bit program memory offset in the range +127 to -128, relative to the address of the next instruction in Program Memory. In a JR or DJNZ operation, RA is added to the Program Counter to cause execution to continue at a new location in Program Memory. In assembly source, the jump address is typically represented as an absolute label, and the assembler calculates RA.

Table 55 consists of additional symbols that are used throughout the instruction set summary.

Table 55. Additional Symbols

Symbol	Definition
dst	Destination Operand
src	Source Operand
@	Indirect Address Prefix
C	Carry Flag
SP	Stack Pointer Value
PC	Program Counter
FLAGS	Flags Register
RP	Register Pointer
#	Immediate Operand Prefix
b	Binary Number Suffix
%	Hexadecimal Number Prefix
h	Hexadecimal Number Suffix
←	Assignment of a value. For example, dst ← dst + src indicates the result is stored in the destination.
↔	Exchange of two values
~	One’s complement unary operator

Flags Register

The Flags Register provides information on the current status of the Z8 CPU. It consists of six bits of status information (Table 56).

Table 56. Flags Register (FLAGS)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	C	Z	S	O	D	H	F1	F2
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	Bank Independent: FCh; Linear 0FCh							

Bit Position	**Value**	**Description**
[7]		Carry Flag (C) Set when the result of an arithmetic operation generates a *carry out of* or a *borrow into* the high-order bit (bit 7) of the result. Also used in rotate and shift instructions.
	0	Flag Clear
	1	Flag Set
[6]		Zero Flag (Z) Set when the result of an arithmetic operation is 0.
	0	Flag Clear
	1	Flag Set
[5]		Sign Flag (S) Stores the value of the most significant bit following an arithmetic, logical, rotate, or shift instruction.
	0	Flag Clear
	1	Flag Set
[4]		Overflow Flag (O) Set when the result of an arithmetic operation is greater than 127.
	0	Flag Clear
	1	Flag Set
[3]		Decimal Adjust Flag (D) Used for binary-coded decimal (BCD) arithmetic.
	0	Flag Clear
	1	Flag Set
[2]		Half Carry Flag (H) Set when a *carry out of* or *borrow into* bit 3 of an arithmetic operation occurs.
	0	Flag Clear
	1	Flag Set

Bit Position	Value	Description
[1]		User Flag 1 (F1) Available to software for use as a general-purpose bit.
	0	Bit Clear
	1	Bit Set
[0]		User Flag 2 (F2) Available to software for use as a general-purpose bit.
	0	Bit Clear
	1	Bit Set

Condition Codes

The C, Z, S and V Flags control the operation of the conditional jump (JP cc and JR cc) instructions. Sixteen frequently useful functions of the Flag settings are encoded in a 4-bit field called the condition code (cc). Table 57 summarizes the condition codes. Some binary condition codes can be created using more than one assembly code mnemonic. The result of the Flag test operation determines if the conditional jump executes.

Table 57. Condition Codes

Binary	Hex	Assembly Mnemonic	Definition	Flag Test Operation
0000	0	F	Always False	–
0001	1	LT	Less Than	(S XOR V) = 1
0010	2	LE	Less Than or Equal	(Z OR (S XOR V)) = 1
0011	3	ULE	Unsigned Less Than or Equal	(C OR Z) = 1
0100	4	OV	Overflow	V = 1
0101	5	MI	Minus	S = 1
0110	6	Z	Zero	Z = 1
0110	6	EQ	Equal	Z = 1
0111	7	C	Carry	C = 1
0111	7	ULT	Unsigned Less Than	C = 1
1000	8	T (or blank)	Always True	–
1001	9	GE	Greater Than or Equal	(S XOR V) = 0
1010	A	GT	Greater Than	(Z OR (S XOR V)) = 0

Table 57. Condition Codes (Continued)

Binary	Hex	Assembly Mnemonic	Definition	Flag Test Operation
1011	B	UGT	Unsigned Greater Than	(C = 0 AND Z = 0)
1100	C	NOV	No Overflow	V = 0
1101	D	PL	Plus	S = 0
1110	E	NZ	Non-Zero	Z = 0
1110	E	NE	Not Equal	Z = 0
1111	F	NC	No Carry	C = 0
1111	F	UGE	Unsigned Greater Than or Equal	C = 0

Z8 LXM CPU Instruction Summary

Table 58 summarizes the Z8 LXM CPU instructions. The table identifies the addressing modes employed by the instruction, the effect upon the Flags register, the number of CPU clock cycles required for the instruction fetch, and the number of CPU clock cycles required for the instruction execution.

Table 58. Z8 LXM CPU Instruction Summary

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Op-code(s) (Hex)	Flags C	Z	S	V	D	H	Cycles Fetch	Cycles Execute
ADC dst, src	dst ← dst + src + C	r	r	12	*	*	*	*	0	*	6	5
		r	Ir	13							6	5
		R	R	14							10	5
		R	IR	15							10	5
		R	IM	16							10	5
		IR	IM	17							10	5

Table 58. Z8 LXM CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Op-code(s) (Hex)	Flags C Z S V D H	Cycles Fetch	Cycles Execute
ADD dst, src	dst ← dst + src	r	r	02	* * * * 0 *	6	5
		r	Ir	03		6	5
		R	R	04		10	5
		R	IR	05		10	5
		R	IM	06		10	5
		IR	IM	07		10	5
AND dst, src	dst ← dst AND src	r	r	52	– * * 0 – –	6	5
		r	Ir	53		6	5
		R	R	54		10	5
		R	IR	55		10	5
		R	IM	56		10	5
		IR	IM	57		10	5
CALL dst	SP ← SP -2 @SP ← PC PC ← dst	IRR		D4	– – – – – –	20	0
		DA		D6		20	0
CCF	C ← ~C			EF	* – – – – –	6	5
CLR dst	dst ← 00h	R		B0	– – – – – –	6	5
		IR		B1		6	5
COM dst	dst ← ~dst	R		60	– * * 0 – –	6	5
		IR		61		6	5
CP dst, src	dst – src – C	r	r	A2	* * * * – –	6	5
		r	Ir	A3		6	5
		R	R	A4		10	5
		R	IR	A5		10	5
		R	IM	A6		10	5
		IR	IM	A7		10	5
DA dst	dst ← DA(dst)	R		40	* * * X – –	8	5
		IR		41		8	5

Table 58. Z8 LXM CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Op-code(s) (Hex)	Flags C Z S V D H	Cycles Fetch	Cycles Execute
DEC dst	dst ← dst – 1	R		00	– * * * – –	6	5
		IR		01		6	5
DECW dst	dst ← dst – 1	RR		80	– * * * – –	10	5
		IR		81		10	5
DI	Disable Interrupts IRQCTL[7] ← 0			8F	– – – – – –	6	1
DJNZ dst, RA	dst ← dst – 1 if dst ≠ 0 PC ← PC + X	r		0A–FA	– – – – – –	NZ/Z 12/10	5
EI	Enable Interrupts IRQCTL[7] ← 1			9F	– – – – – –	6	1
HALT	HALT Mode			7F	– – – – – –	7	0
INC dst	dst ← dst + 1	R		20	– * * * – –	6	5
		IR		21		6	5
		r		0E–FE		6	5
INCW dst	dst ← dst + 1	RR		A0	– * * * – –	10	5
		IR		A1		10	5
IRET	FLAGS ← @SP SP ← SP + 1 PC ← @SP SP ← SP + 2 IRQCTL[7] ← 1			BF	* * * * * *	16	0
JP dst	PC ← dst	DA		8D	– – – – – –	12	0
		IRR		30		8	0
JP cc, dst	if cc is true PC ← dst	DA		0D–FD	– – – – – –	T/F 12/10	0
JR dst	PC ← PC + X	RA		8B	– – – – – –	12	0
JR cc, dst	if cc is true PC ← PC + X	RA		0B–FB	– – – – – –	T/F 12/10	0

Table 58. Z8 LXM CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Op-code(s) (Hex)	Flags C Z S V D H	Cycles Fetch	Cycles Execute
LD dst, src	dst ← src	r	IM	0C–FC	– – – – – –	6	5
		r	R	08–F8		6	5
		R	r	09–F9		6	5
		r	X(r)	C7		10	5
		X(r)	r	D7		10	5
		r	Ir	E3		6	5
		R	R	E4		10	5
		R	IR	E5		10	5
		R	IM	E6		10	5
		IR	IM	E7		10	5
		Ir	r	F3		6	5
		IR	R	F5		10	5
LDC dst, src	dst ← src	r	Irr	C2	– – – – – –	12	0
		Irr	r	D2		12	0
LDCI dst, src	dst ← src r ← r + 1 rr ← rr + 1	Ir	Irr	C3	– – – – – –	18	0
		Irr	Ir	D3		18	0
LDX dst, src	dst ← src	r	Irr	82	– – – – – –	12	0
		Irr	r	92		12	0
LDXI dst, src	dst ← src r ← r + 1 rr ← rr + 1	Ir	Irr	83	– – – – – –	18	0
		Irr	Ir	93		18	0
NOP	No operation			FF	– – – – – –	6	0
OR dst, src	dst ← dst OR src	r	r	42	– * * 0 – –	6	5
		r	Ir	43		6	5
		R	R	44		10	5
		R	IR	45		10	5
		R	IM	46		10	5
		IR	IM	47		10	5

Table 58. Z8 LXM CPU Instruction Summary (Continued)

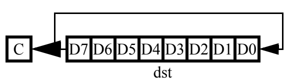
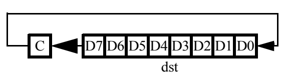
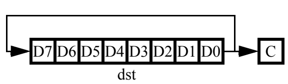
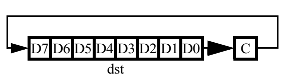
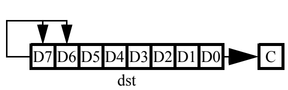
Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Op-code(s) (Hex)	Flags C Z S V D H	Cycles Fetch	Cycles Execute
POP dst	dst ← @SP SP ← SP + 1	R		50	– – – – – –	10	5
		IR		51		10	5
PUSH src	SP ← SP – 1 @SP ← src	R		70	– – – – – –	10	1
		IR		71		12	1
RCF	C ← 0			CF	0 – – – – –	6	5
RET	PC ← @SP SP ← SP + 2			AF	– – – – – –	14	0
RL dst		R		90	* * * * – –	6	5
		IR		91		6	5
RLC dst		R		10	* * * * – –	6	5
		IR		11		6	5
RR dst		R		E0	* * * * – –	6	5
		IR		E1		6	5
RRC dst		R		C0	* * * * – –	6	5
		IR		C1		6	5
SBC dst, src	dst ← dst – src – C	r	r	32	* * * * 1 *	6	5
		r	Ir	33		6	5
		R	R	34		10	5
		R	IR	35		10	5
		R	IM	36		10	5
		IR	IM	37		10	5
SCF	C ← 1			DF	1 – – – – –	6	5
SRA dst		R		D0	* * * 0 – –	6	5
		IR		D1		6	5

Table 58. Z8 LXM CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Op-code(s) (Hex)	Flags C Z S V D H	Cycles Fetch	Cycles Execute
SRP src	RP ← src		IM	31	– – – – – –	6	1
STOP	STOP Mode			6F	– – – – – –	6	0
SUB dst, src	dst ← dst – src	r	r	22	* * * * 1 *	6	5
		r	Ir	23		6	5
		R	R	24		10	5
		R	IR	25		10	5
		R	IM	26		10	5
		IR	IM	27		10	5
SWAP dst	dst[7:4] ↔ dst[3:0]	R		F0	– * * X – –	8	5
		IR		F1		8	5
TCM dst, src	(NOT dst) AND src	r	r	62	– * * 0 – –	6	5
		r	Ir	63		6	5
		R	R	64		10	5
		R	IR	65		10	5
		R	IM	66		10	5
		IR	IM	67		10	5
TM dst, src	dst AND src	r	r	72	– * * 0 – –	6	5
		r	Ir	73		6	5
		R	R	74		10	5
		R	IR	75		10	5
		R	IM	76		10	5
		IR	IM	77		10	5
WDT				5F	– – – – – –	6	0

Table 58. Z8 LXM CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Op-code(s) (Hex)	Flags C Z S V D H	Cycles Fetch	Cycles Execute
XOR dst, src	dst ← dst XOR src	r	r	B2	– * * 0 – –	6	5
		r	Ir	B3		6	5
		R	R	B4		10	5
		R	IR	B5		10	5
		R	IM	B6		10	5
		IR	IM	B7		10	5

Flag States: * = State Depends on Result; – = No Change; X = Undefined; 0 = Cleared; 1 = Set

Electrical Characteristics

Absolute Maximum Ratings

Stresses greater than those listed in Table 59 may cause permanent damage to the device. These ratings are stress ratings only. Functional operation of the device at any condition outside those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. For improved reliability, unused inputs should be tied to one of the supply voltages (V_{DD} or V_{SS}).

Table 59. Absolute Maximum Ratings

Parameter	Minimum	Maximum	Units
Ambient temperature under bias	0	+70	C
Storage temperature	–65	+150	C
Voltage on any pin with respect to V_{SS}*	–0.3	+5.5	V
Voltage on V_{DD} pin with respect to V_{SS}	–0.3	+3.6	V
Maximum current on input and/or inactive output pin	–5	+5	µA
Maximum output current from active output pin	–25	+25	mA
Maximum current into V_{DD} or out of V_{SS}		75	mA

*This voltage applies to all pins except V_{DD}, P32, and P33.

Standard Test Conditions

The characteristics listed in this product specification apply for standard test conditions as noted. All voltages are referenced to Ground. Positive current flows into the referenced pin (see Figure 39).

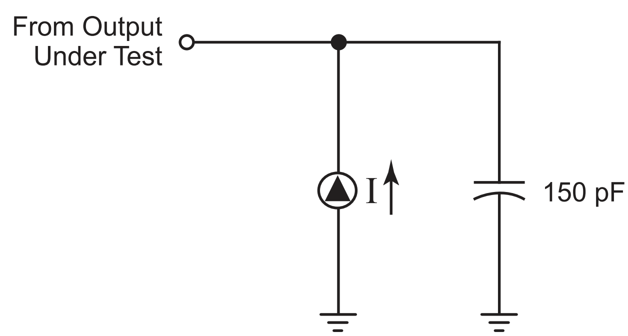


Figure 39. Test Load Diagram

Capacitance

Table 60 lists the capacitances.

Table 60. Capacitance

Parameter	Maximum
Input capacitance	12 pF
Output capacitance	12 pF
I/O capacitance	12 pF
Note: T_A = 25 °C, V_{CC} = GND = 0 V, f = 1.0 MHz, unmeasured pins returned to GND.	

DC Characteristics

Table 61 describes the direct current characteristics of the ZLP12840 OTP MCU.

Table 61. DC Characteristics

			T_A = 0 °C to +70 °C				
Symbol	**Parameter**	**V_{CC}**	**Minimum**	**Typ**	**Maximum**	**Units**	**Conditions**
V_{CC}	Supply Voltage[1]		2.0		3.6	V	See notes 5
V_{CH}	Clock Input High Voltage	2.0–3.6	0.8 V_{CC}		V_{CC}+0.3	V	Driven by External Clock Generator
V_{CL}	Clock Input Low Voltage	2.0–3.6	V_{SS}–0.3		0.4	V	Driven by External Clock Generator
V_{IH}	Input High Voltage	2.0–3.6	0.7 V_{CC}		V_{CC}+0.3	V	
V_{IL}	Input Low Voltage	2.0–3.6	V_{SS}–0.3		0.2 V_{CC}	V	
V_{OH1}	Output High Voltage	2.0–3.6	V_{CC}–0.4			V	I_{OH} = –0.5 mA
V_{OH2}	Output High Voltage (P36, P37, P00, P01)	2.0–3.6	V_{CC}–0.8			V	I_{OH} = –7 mA
V_{OL1}	Output Low Voltage	2.0–3.6			0.4	V	I_{OL} = 4.0 mA
V_{OL2}	Output Low Voltage (P00, P01, P36, P37)	2.0–3.6			0.8	V	I_{OL} = 10 mA
V_{OFFSET}	Comparator Input Offset Voltage	2.0–3.6			25	mV	
V_{REF}	Comparator Reference Voltage	2.0–3.6	0		V_{DD} –1.75	V	
I_{IL}	Input Leakage	2.0–3.6	–1		1	μA	V_{IN} = 0 V, V_{CC}; pull-ups disabled.
I_{IL1}	Input Leakage IR Amp (P31)	2.0–3.6	–2.5		–12	μA	V_{IN} = 0 V, IR amp enabled.
I_{OL}	Output Leakage	2.0–3.6	–1		1	mA	V_{IN} = 0 V, V_{CC}
I_{CC}	Supply Current[2,3]	2.0		1	3	mA	at 8.0 MHz
		3.6		5	10	mA	at 8.0 MHz
I_{CC1}	Standby Current[2,3] (HALT mode)	2.0		0.5	1.6	mA	V_{IN} = 0 V, V_{CC} at 8.0 MHz
		3.6		0.8	2.0	mA	

Table 61. DC Characteristics (Continued)

Symbol	Parameter	V_{CC}	T_A = 0 °C to +70 °C Minimum	Typ	Maximum	Units	Conditions
I_{CC2}	Standby Current[4] (STOP mode)	2.0		1.6	8	µA	V_{IN} = 0 V, V_{CC} WDT is not running
		3.6		1.8	10	µA	
		2.0		5	20	µA	V_{IN} = 0 V, V_{CC} WDT is running
		3.6		8	30	µA	
I_{LV}	Standby Current[5] (Low Voltage)			1.2	6	µA	Measured at 1.3 V
V_{BO}	V_{CC} Low-Voltage Protection			1.9	2.0	V	8 MHz maximum external clock frequency
V_{LVD}	V_{CC} Low-Voltage Detection			2.4		V	
V_{HVD}	V_{CC} High-Voltage Detection			2.7		V	
$T_{ONIRAMP}$	Wake-up time from disabled mode	2.0–3.6			20	µs	
I_{DET}	IR amp current for signal detection	2.0–3.6	10		100	µA	IR amp enabled

Notes
1. Zilog® recommends adding a filter capacitor (minimum 0.1 µF), physically close to V_{DD} and V_{SS} if operating voltage fluctuations are anticipated, such as those resulting from driving an infrared LED.
2. All outputs unloaded, inputs at rail.
3. CL1 = CL2 = 100 pF.
4. Oscillator stopped.
5. Oscillator stops when V_{CC} falls below V_{BO} limit.

AC Characteristics

Figure 40 and Table 62 on page 132 describe the alternating current (AC) characteristics.

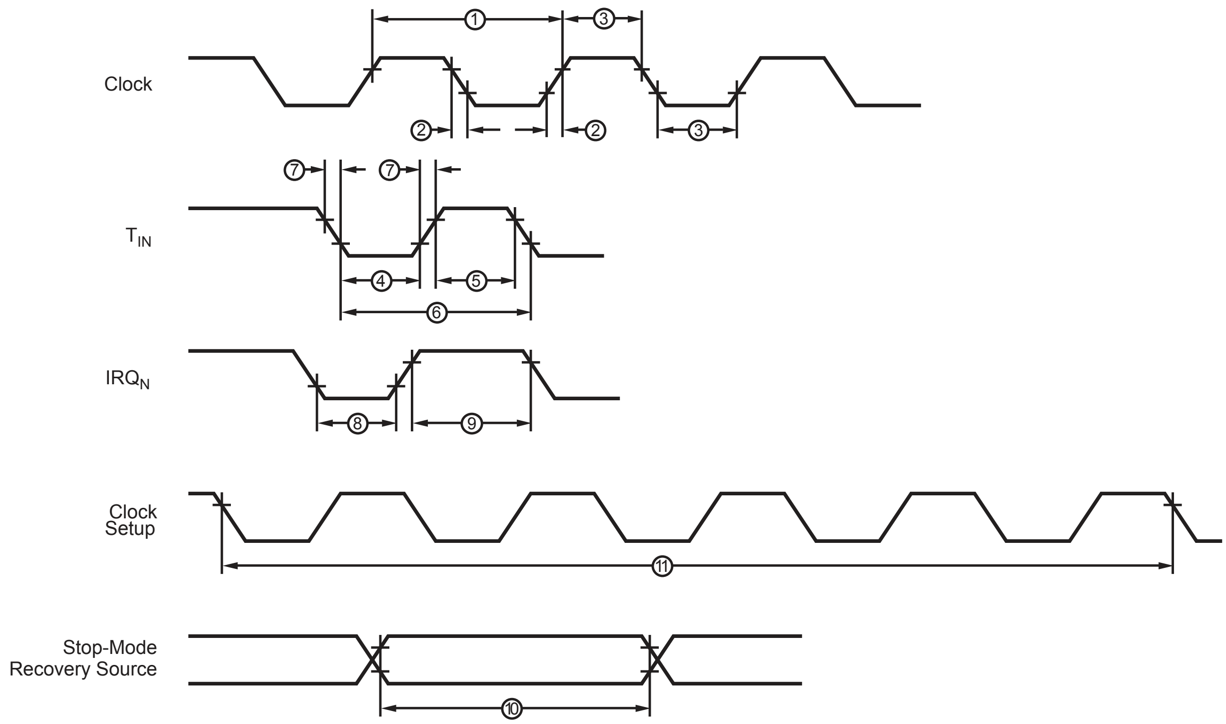


Figure 40. AC Timing Diagram

Table 62. AC Characteristics

SI No	Symbol	Parameter	V_{CC}	T_A = 0 °C to +70 °C 8.0 MHz Minimum	Maximum	Units	WDTMR (Bits 1:0)
1	T_PC	Input Clock Period[1]	2.0–3.6	121	DC	ns	
2	T_RC,T_FC	Clock Input Rise and Fall Times[1]	2.0–3.6		25	ns	
3	T_WC	Input Clock Width[1]	2.0–3.6	37		ns	
4	$T_WT_{IN}L$	Timer Input Low Width[1]	2.0	100		ns	
			3.6	70		ns	
5	$T_WT_{IN}H$	Timer Input High Width[1]	2.0–3.6	$3T_PC$			
6	T_PT_{IN}	Timer Input Period[1]	2.0–3.6	$8T_PC$			
7	T_RT_{IN},T_FT_{IN}	Timer Input Rise and Fall Timers[1]	2.0–3.6		100	ns	
8	T_WIL	Interrupt Request Low Time[1,2]	2.0	100		ns	
			3.6	70		ns	
9	T_WIH	Interrupt Request Input High Time[1,2]	2.0–3.6	$5T_PC$			
10	T_{WSM}	Stop Mode Recovery Width Spec	2.0–3.6	12[3] $10T_PC$[4]		ns	
11	T_{OST}	Oscillator Start-Up Time[4]	2.0–3.6		$5T_PC$		
12	T_{WDT}	Watchdog Timer Delay Time	2.0–3.6	5		ms	0, 0
			2.0–3.6	10		ms	0, 1
			2.0–3.6	20		ms	1, 0
			2.0–3.6	80		ms	1, 1
13	T_{POR}	Power-On Reset	2.0–3.6	2.5	10	ms	

Table 62. AC Characteristics (Continued)

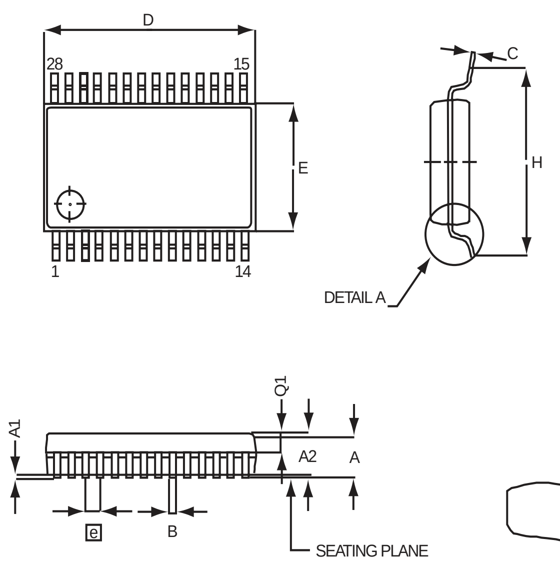
SI No	Symbol	Parameter	V_{CC}	T_A = 0 °C to +70 °C 8.0 MHz Minimum	Maximum	Units	WDTMR (Bits 1:0)
14	f_{MAX}	Maximum frequency of input signal for IR amplifier			500	kHz	
15	f_{MIN}	Minimum frequency of input signal for IR amplifier			0	kHz	

Notes

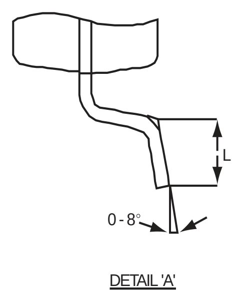
1. Timing Reference uses 0.9 V_{CC} for a logic 1 and 0.1 V_{CC} for a logic 0.
2. Interrupt request through Port 3 (P33:P31).
3. SMR – bit 5 = 1.
4. SMR – bit 5 = 0.

Packaging

Figure 41 displays the 28-pin shrink small outline package (SSOP) for the ZLP12840 device.



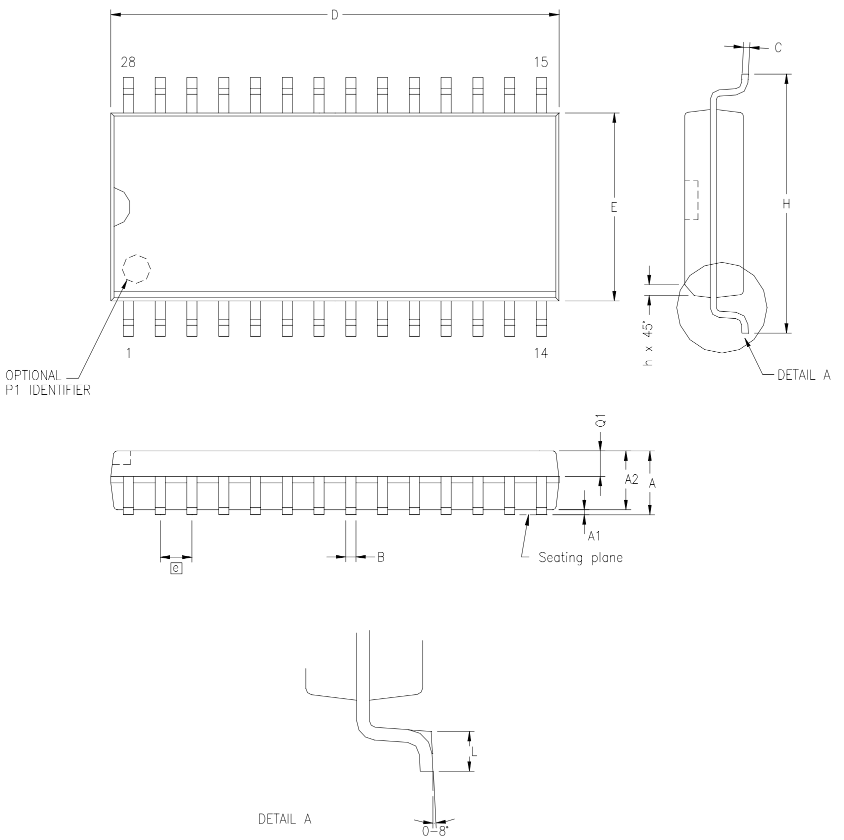
SYMBOL	MILLIMETER			INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.73	1.86	1.99	0.068	0.073	0.078
A1	0.05	0.13	0.21	0.002	0.005	0.008
A2	1.68	1.73	1.78	0.066	0.068	0.070
B	0.25		0.38	0.010		0.015
C	0.09	–	0.20	0.004	0.006	0.008
D	10.07	10.20	10.33	0.397	0.402	0.407
E	5.20	5.30	5.38	0.205	0.209	0.212
e	0.65 TYP			0.0256 TYP		
H	7.65	7.80	7.90	0.301	0.307	0.311
L	0.63	0.75	0.95	0.025	0.030	0.037



CONTROLLING DIMENSIONS: MM
LEADS ARE COPLANAR WITHIN .004 INCHES.

Figure 41. 28-Pin SSOP Package Diagram

Figure 42 displays the 28-pin small outline integrated circuit (SOIC) package for the ZLP12840 device.

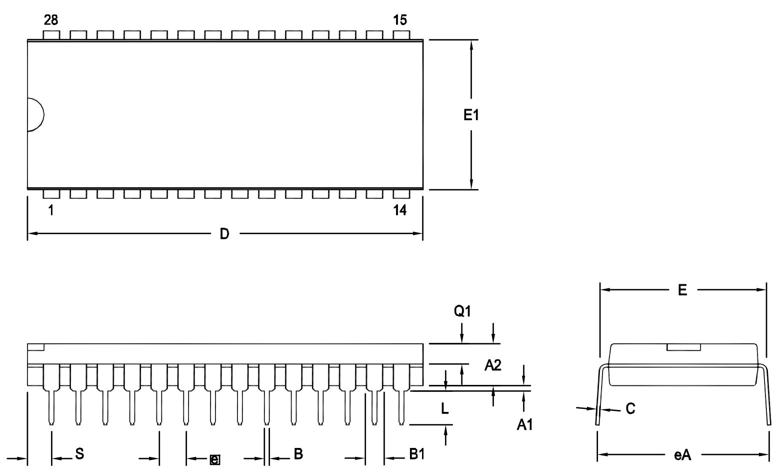


SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A	2.40	2.64	.094	.104
A1	0.10	0.30	.004	.012
A2	2.24	2.44	.088	.096
B	0.36	0.46	.014	.018
C	0.23	0.30	.009	.012
D	17.78	18.00	.700	.710
E	7.40	7.60	.291	.299
e	1.27 BSC		.050 BSC	
H	10.00	10.65	.394	.419
h	0.30	0.71	.012	.028
L	0.61	1.00	.024	.039
Q1	0.97	1.09	.038	.043

CONTROLLING DIMENSIONS : MM
LEADS ARE COPLANAR WITHIN .004 INCH.

Figure 42. 28-Pin SOIC Package Diagram

Figure 43 displays the 28-pin plastic dual inline package (PDIP) for the ZLP12840 device.



SYMBOL	OPT #	MILLIMETER		INCH	
		MIN	MAX	MIN	MAX
A1		0.38	1.02	.015	.040
A2		3.18	4.19	.125	.165
B		0.38	0.53	.015	.021
B1	01	1.40	1.65	.055	.065
	02	1.14	1.40	.045	.055
C		0.23	0.38	.009	.015
D	01	36.58	37.34	1.440	1.470
	02	35.31	35.94	1.390	1.415
E		15.24	15.75	.600	.620
E1	01	13.59	14.10	.535	.555
	02	12.83	13.08	.505	.515
e		2.54 TYP		.100 BSC	
eA		15.49	16.76	.610	.660
L		3.05	3.81	.120	.150
Q1	01	1.40	1.91	.055	.075
	02	1.40	1.78	.055	.070
S	01	1.52	2.29	.060	.090
	02	1.02	1.52	.040	.060

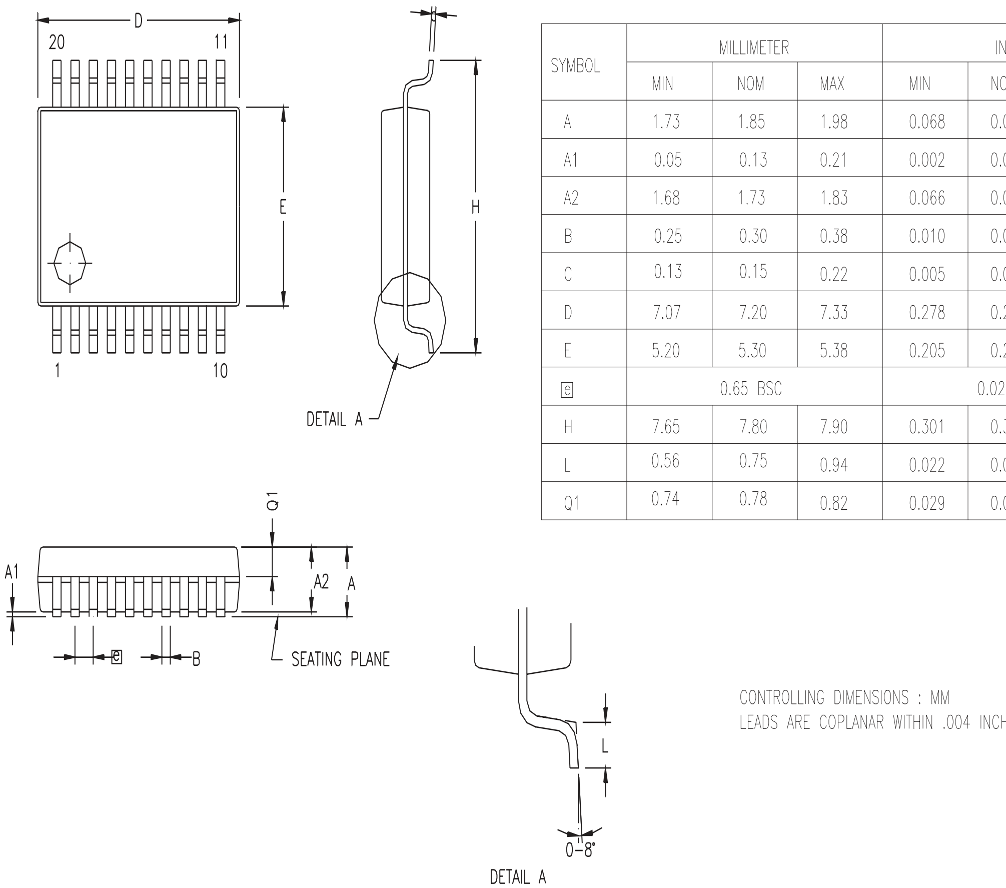
CONTROLLING DIMENSIONS : INCH

OPTION TABLE	
OPTION #	PACKAGE
01	STANDARD
02	IDF

Note: ZiLOG supplies both options for production. Component layout PCB design should cover bigger option 01.

Figure 43. 28-Pin PDIP Package Diagram

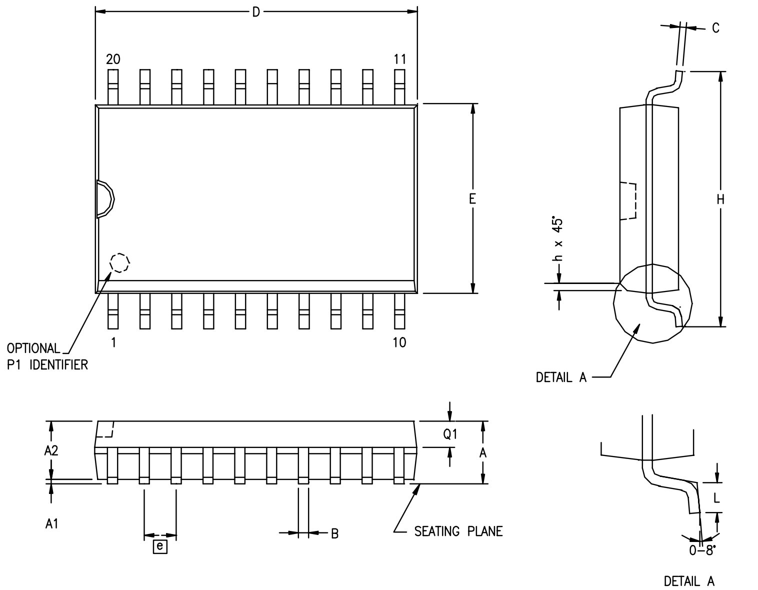
Figure 44 displays the 20-pin shrink small outline package (SSOP) for the ZLP12840 device.



SYMBOL	MILLIMETER			INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.73	1.85	1.98	0.068	0.073	0.078
A1	0.05	0.13	0.21	0.002	0.005	0.008
A2	1.68	1.73	1.83	0.066	0.068	0.072
B	0.25	0.30	0.38	0.010	0.012	0.015
C	0.13	0.15	0.22	0.005	0.006	0.009
D	7.07	7.20	7.33	0.278	0.283	0.289
E	5.20	5.30	5.38	0.205	0.209	0.212
e	0.65 BSC			0.0256 BSC		
H	7.65	7.80	7.90	0.301	0.307	0.311
L	0.56	0.75	0.94	0.022	0.030	0.037
Q1	0.74	0.78	0.82	0.029	0.031	0.032

Figure 44. 20-Pin SSOP Package Diagram

Figure 45 displays the 20-pin small outline integrated circuit (SOIC) package for the ZLP12840 device.

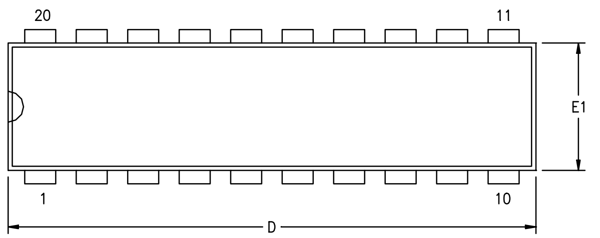


SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A	2.40	2.65	.094	.104
A1	0.10	0.30	.004	.012
A2	2.24	2.44	.088	.096
B	0.36	0.46	.014	.018
C	0.23	0.30	.009	.012
D	12.60	12.95	.496	.510
E	7.40	7.60	.291	.299
e	1.27 BSC		.050 BSC	
H	10.00	10.65	.394	.419
h	0.30	0.40	.012	.016
L	0.60	1.00	.024	.039
Q1	0.97	1.07	.038	.042

CONTROLLING DIMENSIONS : MM
LEADS ARE COPLANAR WITHIN .004 INCH.

Figure 45. 20-Pin SOIC Package Diagram

Figure 46 displays the 20-pin plastic dual inline package (PDIP) for the ZLP12840 device.



SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A1	0.38	0.81	.015	.032
A2	3.25	3.68	.128	.145
B	0.41	0.51	.016	.020
B1	1.47	1.57	.058	.062
C	0.20	0.30	.008	.012
D	25.65	26.16	1.010	1.030
E	7.49	8.26	.295	.325
E1	6.10	6.65	.240	.262
e	2.54 BSC		.100 BSC	
eA	7.87	9.14	.310	.360
L	3.18	3.43	.125	.135
Q1	1.42	1.65	.056	.065
S	1.52	1.65	.060	.065

CONTROLLING DIMENSIONS : INCH

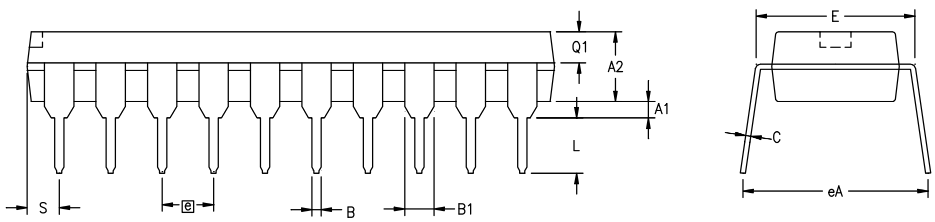


Figure 46. 20-Pin PDIP Package Diagram

Ordering Information

Table 63 provides a product specification index code and a brief description of each part. Each of the parts listed in Table 63 is shown in a lead-free package.

The use of lead-free packaging adheres to a socially responsible environmental standard. For a description of a part number's unique identifying attributes, see the Part Number Description on page 142.

Table 63. ZLP12840 OTP MCU Part Numbers Description

PSI No	Description	PSI No	Description
Lead-Free Environmental Flow			
ZLP12840H2828G	28-pin SSOP 128K OTP	ZLP12840H2896G	28-pin SSOP 96K OTP
ZLP12840S2828G	28-Pin SOIC 128K OTP	ZLP12840S2896G	28-Pin SOIC 96K OTP
ZLP12840P2828G	28-Pin PDIP 128K OTP	ZLP12840P2896G	28-Pin PDIP 96K OTP
ZLP12840H2028G	20-Pin SSOP 128K OTP	ZLP12840H2096G	20-Pin SSOP 96K OTP
ZLP12840S2028G	20-pin SOIC 128K OTP	ZLP12840S2096G	20-pin SOIC 96K OTP
ZLP12840P2028G	20-pin PDIP 128K OTP	ZLP12840P2096G	20-pin PDIP 96K OTP
ZLP12840H2864G	28-pin SSOP 64K OTP	ZLP12840H2832G	28-pin SSOP 32K OTP
ZLP12840S2864G	28-Pin SOIC 64K OTP	ZLP12840S2832G	28-Pin SOIC 32K OTP
ZLP12840P2864G	28-Pin PDIP 64K OTP	ZLP12840P2832G	28-Pin PDIP 32K OTP
ZLP12840H2064G	20-Pin SSOP 64K OTP	ZLP12840H2032G	20-Pin SSOP 32K OTP
ZLP12840S2064G	20-pin SOIC 64K OTP	ZLP12840S2032G	20-pin SOIC 32K OTP
ZLP12840P2064G	20-pin PDIP 64K OTP	ZLP12840P2032G	20-pin PDIP 32K OTP

Applications and Support Tools

The following development tools are available for programming and debugging this device:

- ZCRMZNICE01ZEMG—Crimzon In-Circuit Emulator
- ZCRMZNICE01ZACG—20-pin Accessory Kit to the ZCRMZNICE01ZEMG
- ZCRMZNICE02ZACG—40-/48-pin Accessory Kit to the ZCRMZNICE01ZEMG

- ZCRMZN00100KITG—Crimzon IR Development Kit
- Zilog Developer Studio II (ZDSII), available for download at www.zilog.com

For valuable information about customer and technical support as well as hardware and software development tools, visit the Zilog web site at www.zilog.com. The latest released version of ZDS can be downloaded from this web site.

Part Number Description

Zilog® part numbers consist of a number of components, as displayed in Figure 47. The example part number ZLP12840H2828G is a Crimzon One-Time Programmable (OTP) product in a 28-pin SSOP package, with 128 KB of OTP and built with lead-free solder.

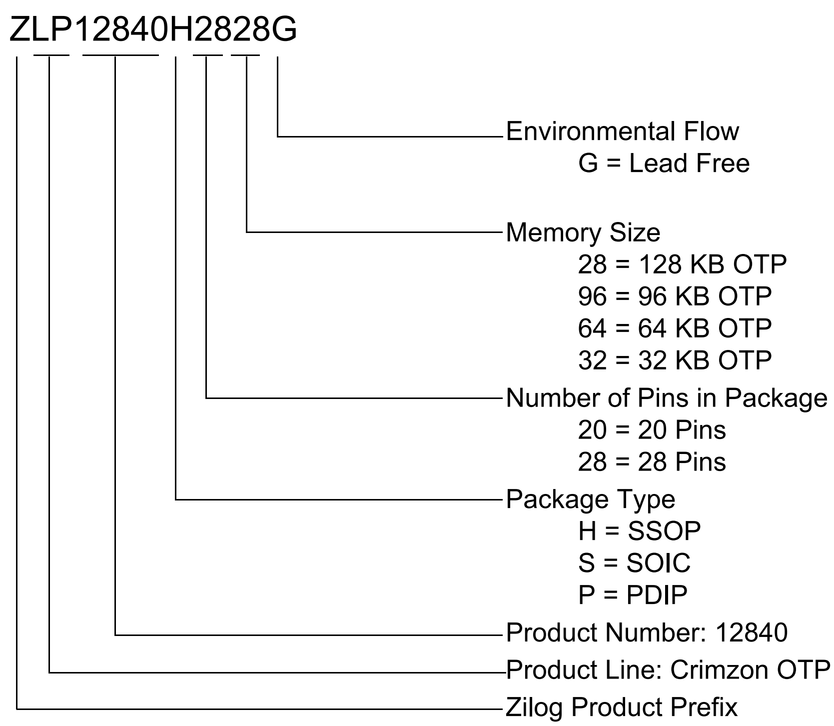


Figure 47. Part Number Example

Precharacterization Product

The product represented by this document is newly introduced and Zilog® has not completed the full characterization of the product. The document states what Zilog knows about this product at this time, but additional features or nonconformance with some aspects of the document might be found, either by Zilog or its customers in the course of further application and characterization work. In addition, Zilog cautions that delivery might be uncertain at times due to start-up yield issues. For more information, visit www.zilog.com.

Index

M

N

O

P

R

S

W

X

Z

Customer Support

For answers to technical questions about the product, documentation, or any other issues with Zilog's offerings, please visit Zilog's Knowledge Base at http://www.zilog.com/kb.

For any comments, detail technical questions, or reporting problems, please visit Zilog's Technical Support at http://support.zilog.com.