Analog Devices Inc./Maxim Integrated - ZLF645E0H2864G Datasheet

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	Z8 LXMC
Core Size	8-Bit
Speed	8MHz
Connectivity	UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, WDT
Number of I/O	24
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.9V ~ 3.6V
Data Converters	-
Oscillator Type	-
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/zlf645e0h2864g

Interrupt Sources

The ZLF645 MCU supports 23 interrupt sources with 6 interrupt vectors, as given below:

- Three external interrupts.
- Two from T8, T16 time-out and capture.
- Three from UART Tx, UART Rx, and UART BRG.
- One from LVD.
- Fourteen from SMR source P20-P27, P30-P33, P00, and P07:
  - Any change in logic from P20-P27, P30-P33 can generate an interrupt or SMR

Additional Features

The additional features of ZLF645 MCU include:

- IR learning amplifier.
- Low power consumption—11 mW (typical).
- Three standby modes:
  - STOP—1.7 μA (typical)
  - HALT—0.6 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

Table 5 lists the function and signal directions of each pin within the 20-pin PDIP, SOIC, and SSOP packages sequentially by pin number.

Table 5. 20-Pin PDIP/SOIC/SSOP Sequential Pin Identification

Pin No	Symbol	Function	Signal 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	Input
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	Input/Output
12	P36	Port 3, bit 6	Output
13	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 In	put
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

Note: When the Port 0 low-nibble pull-up option is enabled and the P30 input is Low, current flows through the pull-up to Ground.

Figure 9 displays the Port 3 configuration.

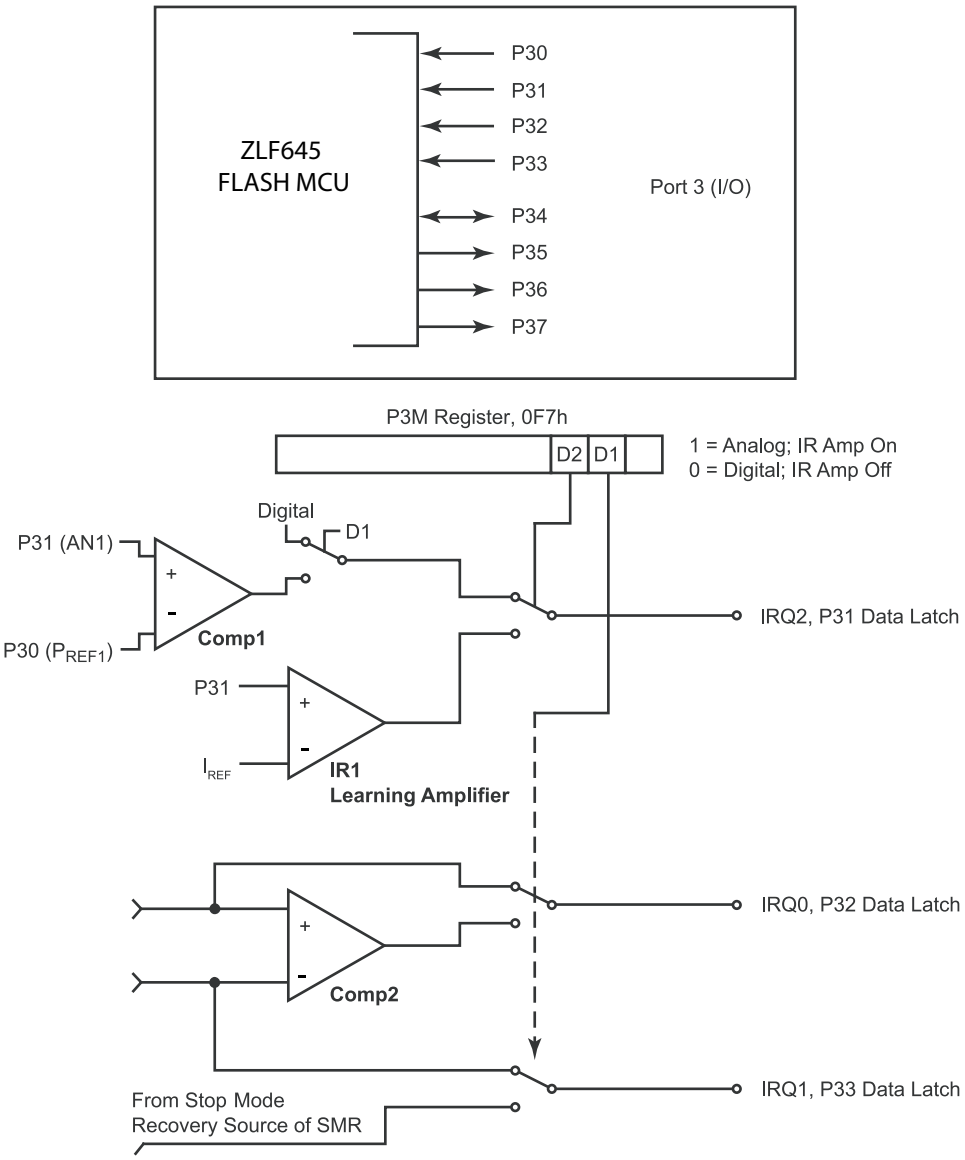

Figure 9. Port 3 Configuration

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,

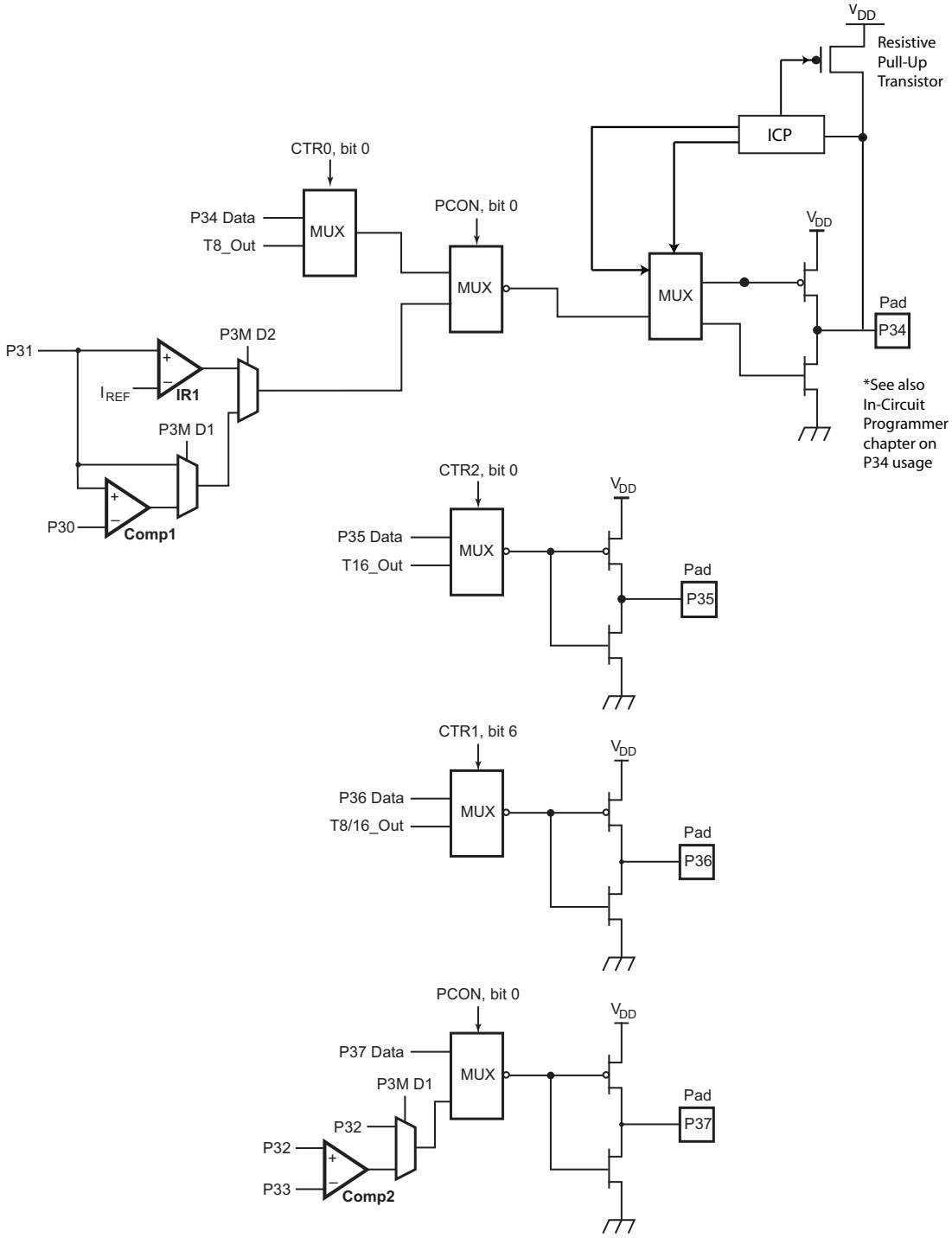

Figure 10. Port 3 Counter/Timer Output Configuration

Register File

The ZLF645 Series of Flash MCUs features up to 1024 bytes of register file space, organized in 256-byte banks. Bank 0 contains 235 or 237 bytes of RAM addressed as general purpose registers, 5 or 3 port addresses, and 16 control register addresses. For 20- or 28-pin packages, Port 1 and Port 4 registers of Bank 0 are not implemented and there locations are available as general-purpose registers. Bank 1, Bank 2, and Bank 3; each contain 256 general-purpose register bytes. Bank D and Bank 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 13 and Figure 14).

The 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 Bank F is selected (Port 01h is not implemented in this device). When Bank D or Bank F is selected, addresses 10h–EFh access the Bank 0 general-purpose registers.

The LDX and LDXI instructions or indirect addressing is 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 45.

Stack

The Stack Pointer register provides either 16-bit or 8-bit of stack pointer addressability depending upon the programming of bit 3 of User Option Byte 1 (for more details, see Flash Option Bits on page 171).

16-bit Stack Addressability

When programmed for 16-bit stack addressability, the stack address is formed as a combination of the SPL and SPH registers located at addresses FFh and FEh. For 1K and 512 B RAM products, the most significant 6 or 5 bits, respectively of the SPH register are ignored. The stack address is mapped to a particular RAM memory location by the following formula:

Bank = {2'b0, SPH[1:0]}

Group = SPL[7:4]

Register number = SPL[3:0]

With the ZLF645 MCU configured for 16-bit stack addressability, stack reads or writes to Bank 3, 2, 1, or 0 Group F Registers or to any of the Port registers actually accesses

shadow registers implemented within the RAM memory. This enables the entire 1K or 512 B, depending on the product, of the RAM memory to be used for the stack.

8-bit Stack Addressability

For 8-bit stack addressability, only the SPL register is used for stack addressing and stack operations that use the stack pointer always address Bank 0, independent of the RP[3:0] setting. For more details on the stack, refer to *Z8® LXMC CPU Core User Manual (UM0215)*.

When in 8-bit stack addressability mode, the Bank 0 register FEh can be used to store user data. See Stack Pointer Register on page 48.

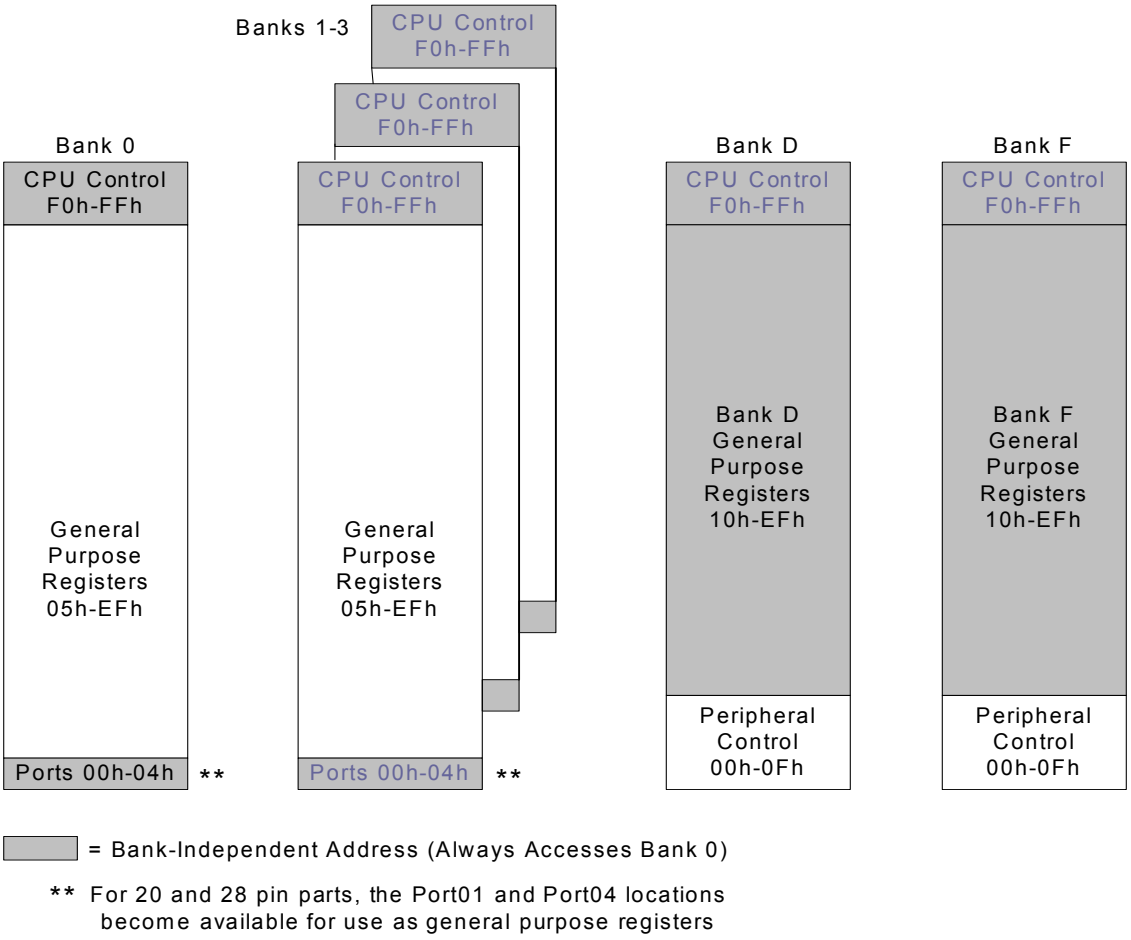

Figure 13. Register File 8-Bit Banked Address Map

Register Pointer Register

The upper nibble of the Register Pointer register (see Table 23) selects which working register group is accessed. A working register group consists of 16 bytes. The lower nibble selects the expanded register file bank; for ZLF645 MCU, 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 23. 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.

Stack Pointer Register

Through a Flash programmable option bit, the Stack Pointer register of the ZLF645 MCU is either one or two bytes providing either 8-bit or 16-bit of stack addressing. When not enabled through the option bit for 16-bit stack addressability, the SPH register can be used as a User Data register (USER). The stack pointer resides in the RAM and when the ZLF645 MCU is programmed for 8-bit addressing, this stack pointer resides in Bank 0 of the RAM only. With 16-bit addressing, the entire RAM's address space is available for use as the stack.

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 or unpredictable operations may result (see Table 24 on page 49).

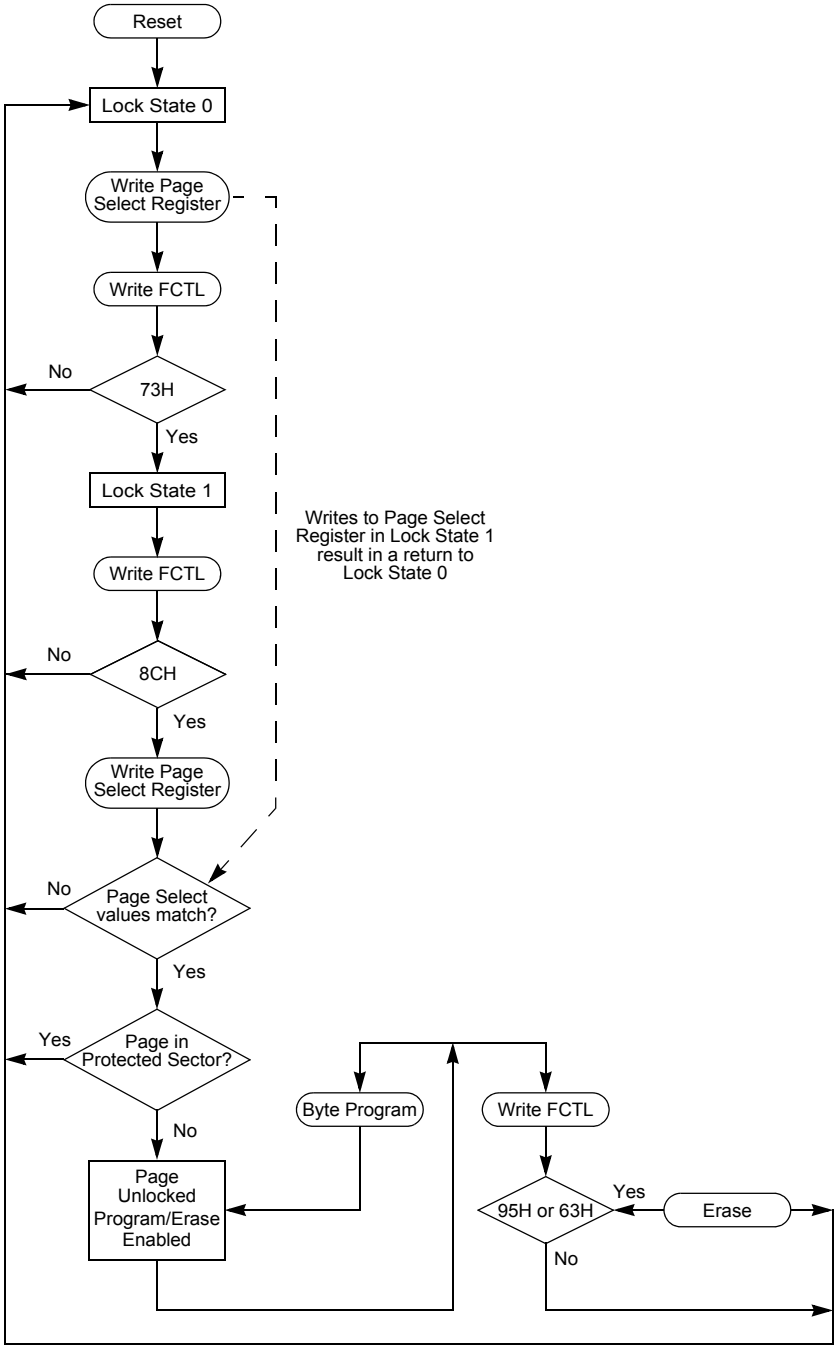

Figure 20. Flash Controller Operation Flow Chart

Flash Code Protection Against External Access

The Flash Controller limits Flash Access capabilities of the ICP and Flash Byte Programming Interfaces based upon the Flash read/write protect bits in User Option Byte 1. By programming these bits, you can configure the Flash Controller to block page 3 information area erasures, main memory reads, and main memory page erasures and programming as initiated through the ICP or Byte Programming Interfaces of the ZLF645. For more information, see Table 85 on page 174.

Flash Code Protection Against Accidental Program and Erasure

As mentioned previously, the ZLF645 products provide several levels of protection against accidental program and erasure of the Flash main memory contents by ICP and CPU accesses through the Flash Controller. Through the Flash Controller's register locking mechanism, page select redundancy, and sector level protection control, the ZLF645 products provide protection against accidental program and erasure of the Flash main memory contents by CPU and ICP accesses, except that for the ICP sector level protection is ignored. Similar levels of protection are in place for the Flash Information Area, minus the sector level protection.

Sector Based Flash Protection

For CPU initiated Flash main memory accesses, programming/erase protection is possible on a sector level basis through programming of the Flash Controller's Sector Protect (FSEC) register. For all ZLF645 products, each sector contains 16 pages (of 512 bytes each).

Part Number	Number of Sectors
ZLF645xxxxx32	4
ZLF645xxxxx64	8

The Sector Protect (FSEC) register controls the protection state of each Flash sector. This register is address-shared with the Page Select register. It can only be accessed with the Flash Controller in 'locked' state. With the Flash Controller in 'locked' state, writing the Flash Control (FCTL) register with a value `5EH` enables the Flash Controllers Sector Protect register to be written. The next write performed to Bank F, Register Address `02H` then targets the Flash Controller's Sector Protect (FSEC) register.

The Sector Protect register is initialized to 0 on reset, putting each sector into an unprotected state. When a bit in the Sector Protect register is written to 1, the corresponding sector within the Flash memory can no longer be programmed or erased if for operations initiated by the CPU. Operations through the ICP are unaffected by the

settings of the Sector Protect (FSEC) register. After a bit of the Sector Protect register has been set, it cannot be cleared except by powering down the device.

Byte Programming

All Flash accesses either through CPU code execution or through the ICP interface occur using the Flash memory byte mode of operation. The Flash Controller allows CPU programming access to the Flash's main memory area only whereas the ICP has access to both the main memory and the page 3 information area for programming. The Flash memory is enabled for byte programming by either the CPU or the ICP after unlocking the Flash Controller and executing either a Mass Erase or Page Erase operation. When the Flash Controller is unlocked and a main memory Mass Erase is executed, all Flash Main Memory locations are available for byte programming by the CPU. In contrast, when the Flash Controller is unlocked and a main memroy Page Erase is executed, only the locations of the selected page as per the Page Select (PGS) register are available for byte programming by the CPU. An erased Flash byte contains all 1's (FFH).

The programming operation can only be used to change bits from 1 to 0. To change a Flash bit (or multiple bits) from 0 to 1 requires an erase operation through execution of either a Page Erase or Mass Erase command to the Flash Controller.

Byte Programming can be accomplished through the ICP by using the Write Memory command or by the Z8 LXMC CPU through execution of the LDC or LDCI instructions. For a description of the LDC and LDCI instructions, refer to *Z8® LXMC CPU User Manual (UM0215)*. During execution of a CPU initiated programming operation the system clock to the CPU is halted preventing further code execution, however the system clock and the on-chip peripherals continue to operate. Once the programming operation is complete, the CPU resumes code execution. To exit programming mode and lock the Flash the CPU can perform a write of any value to the Flash Control (FCTL) register, except the Mass Erase or Page Erase commands.

Page Erase

The Flash main memory can be erased one page (512 bytes) at a time. Page Erasing the Flash memory sets all bytes in that page to the value FFH. The Flash Page Select (PGS) register identifies the page to be erased. For CPU initiated page erase operations, only a page residing in an unprotected sector can be erased. With the Flash Controller unlocked and the active page set, writing the value 95H to the Flash Control (FCTL) register initiates the Page Erase operation. As with programming, during execution of a CPU initiated page erase operation the system clock to the CPU is halted preventing further code execution, however the system clock and the on-chip peripherals continue to operate. Once the page erase operation is complete, the CPU resumes code execution.

If a Page Erase operation to the Flash's main memory is performed using the ICP, bit 3 of the ICP Status register can be polled to determine when the operation is complete. When the Page Erase is complete, the Flash Controller returns to its locked state. Although the

Flash Controller prevents CPU accesses to the Flash's Information block, the ICP can initiate a Page erase to page 3 of Information Area by a similar process as used for the main memory. The only difference is that the ICP must first write bit 7 of the Flash Page Select (PGS) register to a 1 before writing the page erase command to the Flash Control (FCTL) register. For more details, see Table 34 on page 77.

Mass Erase

The Flash main memory can also be Mass Erased using the Flash Controller, but only through the ICP interface and not by the CPU. Mass Erasing the Flash memory sets all bytes to the value FFH. With the Flash Controller unlocked, writing the value 63H to the Flash Control register initiates the Mass Erase operation. If a Mass Erase operation is performed using the ICP, bit 3 of the ICP Status register is polled to determine when the operation is complete. When the Mass Erase is complete, the Flash Controller returns to its locked state. You cannot mass erase the Information Area.

Caution: *If either of the Flash Memory Protect Option Bits are set as defined in the Flash Option Bits section, a mass erase of the Flash's main memory must be performed before Page 3 of the Flash's Information Area can be erased. These two operations must be done when the device is at operating voltage. That is, if a mass erase is followed with a power-down then power-up sequence, performing an Information Area Page 3 erase will not erase its contents.*

Flash Control Register Definitions

Flash Control Register

The Flash Controller must be unlocked using the Flash Control (FCTL) register (see Table 33) before the Flash Controller is enabled for programming or erasing the Flash memory. Writing values of 73H and then 8CH sequentially to the Flash Control register unlocks the Flash Controller, as long as the other conditions described in Enabling the Flash Controller For Flash Memory Accesses on page 70 have been met. When the Flash Controller is unlocked, a Mass Erase initiated by the ICP, or Page Erase initiated by the ICP or CPU can be executed by the Flash Controller by writing the appropriate command value to this register. Execution of a Page Erase applies only to the active page selected in Flash Page Select (FPS) register. Writing an invalid value or an invalid sequence returns the Flash Controller to its locked state. The Write-only Flash Control register shares its Register File address with the Read-only Flash Status register.

Baud Rate Generator Interrupts

If the BRG interrupt enable is set, the UART Receiver interrupt asserts when the UART Baud Rate Generator reloads. This action allows the BRG 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 below equation:

$$\text{UART Data Rate (bps)} = \frac{\text{System Clock Frequency (Hz)}}{\text{16 x UART Baud Rate Divisor Value (BCNST)}}$$

When the UART Baud Rate Low register is programmed to 00h, the UART data rate is calculated as follows:

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

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:

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

Note: *The relationship between the XTAL1 clock frequency and the system clock frequency must be considered before making this computation and is dependent upon the programming of bit 2 of User Option Byte 1 as well as the programming of bit 0 of the SMR register. Depending on the programmed values, the system clock frequency can be a divide by 1, a divide by 2, or a divide by 16 of the XTAL1 clock.*

When the UART is disabled, the BRG can function as a basic 8-bit timer with interrupt on time-out.

Counter/Timer 8 Control Register

The Counter/Timer 8 Control register (see Table 56) controls the timer function of the T8 timer.

Caution: *Writing 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: *You must be careful when using the OR or AND commands to manipulate CTR0, bit 5 and CTR1, bits 0 and 1 (DEMODULATION Mode). 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 56. Counter/Timer 8 Control Register (CTR0)

Bit	7	6	5	4	3	2	1	0
Field	T8_Enable	SINGLE-PASS/ 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							

Programming bits for the Interrupt Edge Select are located in the IRQ register (R250), bit 6 and bit 7. Table 62 provides the configuration.

Table 62. 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 (see Table 63) defines which interrupt holds 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, IRQ5.

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

Table 63. 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]		**Group A Priority (IRQ3, IRQ5)**
	0	IRQ5 > IRQ3
	1	IRQ3 > IRQ5

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 71 on page 152.

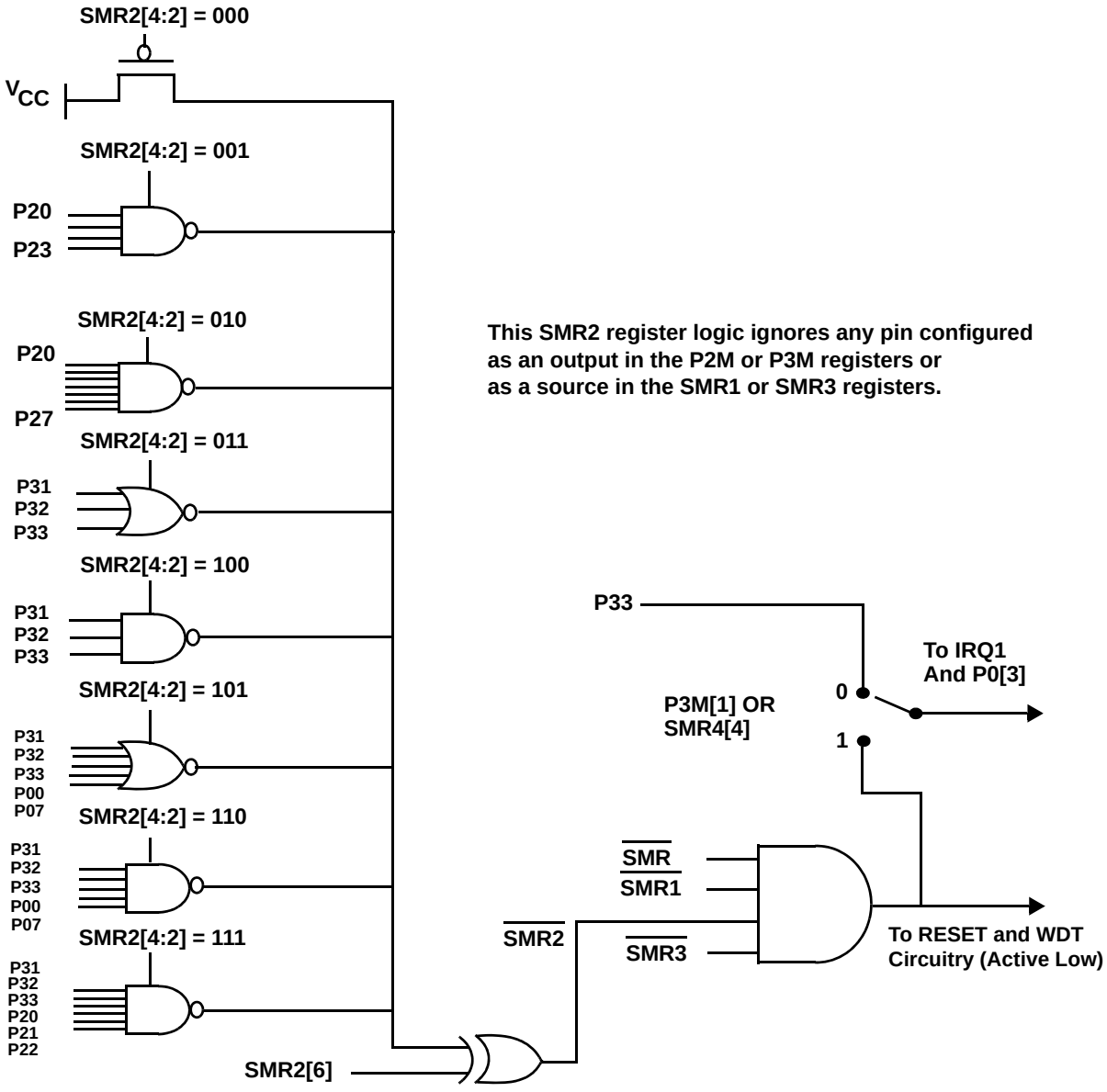

Figure 46. SMR2 Register-Controlled Event Sources

Table 72. 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; Must be written to 1.
[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 Stop Mode Recovery.*

Flash Option Bits

Programmable Flash Option Bits allow user configuration of certain aspects of ZLF645 MCU functionality. This configuration data is stored in the Flash memory Information Block and then read into option byte shadow registers during the last portion of the ZLF645 MCUs reset period.

Features available for control through the Flash Option Bits include:

- Port 0 low nibble pull-ups
- Port 0 high nibble pull-ups
- Port 1 low nibble pull-ups
- Port 1 high nibble pull-ups
- Port 2 pull-ups
- Port 3 low nibble pull-ups
- Port 4 pull-ups
- WDT always enabled
- Flash protect entire main memory
- Flash protect lower half main memory
- XTAL1 to System Clock (no division enable)
- 16-bit Stack addressiblity enable

Operation

Option Bit Shadow Register Loading By Reset

For each Flash memory option bit, there is an associated option bit shadow register that is used to register the value of the option bit. The output of the option bit shadow registers are used by the ZLF645 MCU to enable various features and functions for the ZLF645 MCU. Each time the Flash Memory Information Block Option Bits are programmed or erased, the device must be reset for the change in ZLF645 configuration to take effect.

A POR or Stop Mode Recovery Reset with SMR bit 5 set to 1, loads the option bits from the Flash memory to the Option Bit Shadow registers during the last few clock cycles of the reset period. In some cases, in order to provide a required value before being loaded, the Option Bit Shadow registers are reset to a predefined value on the start of the reset period.

[0]		**FLRWP**—Flash Main Memory Protect
	1	Flash Main Memory and Information Area Page 3 can be read, programmed, and erased by both the Flash Byte Programming interface or through the ICP interface.
	0	Reads and writes to the Flash main memory and writes and erasures to Information Area Page 3 by the ICP or Flash Byte Programming interfaces is disabled unless, with this bit 0, a main memory mass erase is completed first. A main memory mass erase causes resetting of this bit value in the Option Byte 1 shadow register to a 1 but does not effect the corresponding Flash memory bit. Once the Option Byte 1 shadow register bit is reset, the ICP or Flash Byte Programming interface is allowed full read, write, and erase access to the Flash's main memory and to Page 3 of the Information Area and can reset the corresponding Flash memory bit.

Table 86. User Option Byte 1 Shadow Register (OPT1SR)

Bit	7	6	5	4	3	2	1	0
Field	RESERVED				16BITSTK	DIVBY1	FLPROT1	FLRWP
Reset State	X	X	X	X	X	1	X	X
CPU Access (R/W)					R/W	R/W	R	R
Register Address (R/W)	Bank D: 0Fh; Linear: D0Fh							

Note: *RESERVED bits when read by the CPU will return 0 and when written have no effect.*

U

V

W

X

Z

Customer Support

For any comments, detail technical questions, or reporting problems, please visit Maxim's Technical Support at https://support.maxim-ic.com/micro.