# ENT-AN1187 Application Note
# Using the Serial GPIO/LED Controller

Microsemi
Power Matters.

**Microsemi Corporate Headquarters**
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com





## About Microsemi

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, California, and has approximately 4,800 employees globally. Learn more at www.microsemi.com.

VPPD-04342. 1.0 12/16

# Contents

# Figures

# Tables

# 1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

## 1.1 Revision 1.0

Revision 1.0 was the first publication of this document.

# 2 Using the Serial GPIO/LED Controller

The content of this document can be applied to all Microsemi Ethernet switch chips that feature a serial GPIO (SIO) controller. The SIO controller significantly extends the number of available GPIOs with a minimum number of additional pins on the device. The SIO controller can be used for serial LEDs, status, and control of SFP modules, or provide additional customized inputs and outputs.

This application note describes the SIO controller use and examples of how to implement the serial LED feature, as well as control and monitor the SFP modules.

## 2.1 SIO

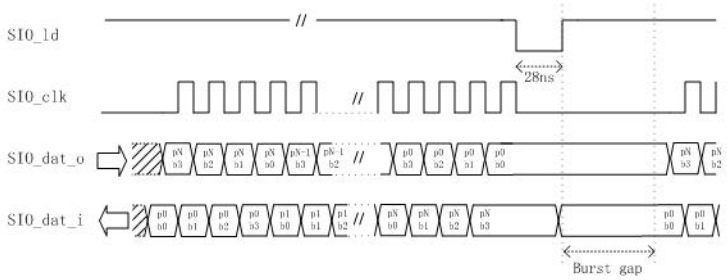
The SIO controller uses a 4-pin serial interface to significantly extend the number of available general purpose I/O pins. The SIO controller has both serial input and serial output functions.

The following illustration shows the input/output timing of the SIO controller. Serial data is output on the SG_DO pin, which is clocked by the SG_CLK pin in bursts. After each burst, there is an assertion of the SG_LD signal. Serial data can be output as a single burst or as continuous bursts separated by a configurable burst gap. The burst gap is configured to between 0 and 33 ms in steps of approximately 1 ms. At the same time, as shifting out serial data on SG_DO, the SIO controller also samples the SG_DI input. The values sampled on SG_DI are made available to software or forwarded to switch port modules if LOS operation is configured.

When the controller is used for serial LEDs, each single bit in the SG_DO stream corresponds to a different LED. A single burst of data is a frame of data for all the LEDs at a certain moment. External serial-to-parallel shift registers are needed to convert the serial data to parallel signals for driving LEDs. External parallel to serial shift registers should be used for serial input. SG_LD latches parallel input values to the shift registers while SG_CLK shifts them to the switch device bit by bit.

SG_LD can assure that outputs are stable when serial data is being shifted through the shift registers. This can be done by using the SG_LD signal to load the serial data onto the parallel output pins after the burst has completed for load-capable shift registers. If the SIO controller is only used for serial LED output, then SG_LD is optional because it is usually acceptable for outputs to toggle when serial data is updated (for instance, shifted through the chain). When the shift frequency is fast enough, the shifts through the chain are not perceptible to the human eye. It is very easy to make the shift frequency fast enough because the clock frequency of SG_CLK is programmable.

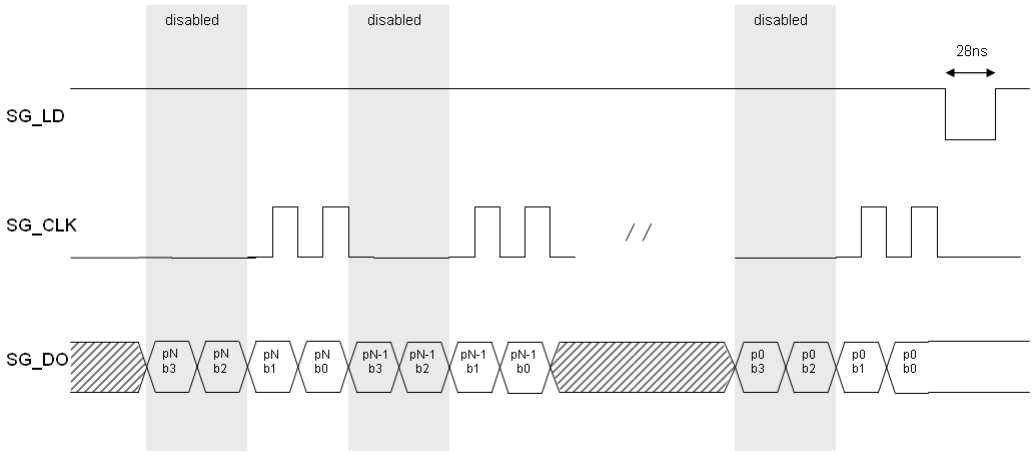
***Figure 1 •*** **SIO Timing**




The maximum burst length is 128 bits. By default, it is organized into 32 ports each with 4 bits per port. However, each port can be enabled or disabled individually. The port width (number of bits per port) is also centrally configurable, so all enabled ports will have the same width. When a port and/or a bit is disabled, the corresponding SG_CLK is omitted.

The following illustration shows the timing of the SIO output stream when the port width is set to 2.

***Figure 2 •*** **SIO Output Timing with Disabled Bits**




## 2.2 Using SIO for Serial LED

Ethernet switches generally use LEDs to indicate link status for all their ports. Normally, more than one LED per port is used because a lot of information needs to be shown (link status, speed, duplex, activity, and so on). There are a lot of LEDs on the front panel of a 24-port Ethernet switch. Traditional designs use a parallel LED scheme, which means there is a dedicated pin for each LED provided from the Ethernet chipset. The large number of LED pins increases both the chipset and printed circuit board (PCB) cost, as more layers may be required to route all the LED traces to the front panel. If the LEDs are placed on a separate LED board, such that all the parallel LED signals must be bridged over from the main board containing the Ethernet switch to the LED board through ribbon wires, then it may reduce the system reliability.

To solve the problem, a serial LED scheme has recently been adopted by customers. Serial LEDs use only two pins, a serial data line, and a serial clock line to send out a serial stream of LED status bits. This is then converted to parallel signals externally through a chain of serial-to-parallel shift registers. The shift registers can be placed at the appropriate locations of the board to ease the board layout. The number of wires between the main board and LED board are reduced significantly for the LED board usage.

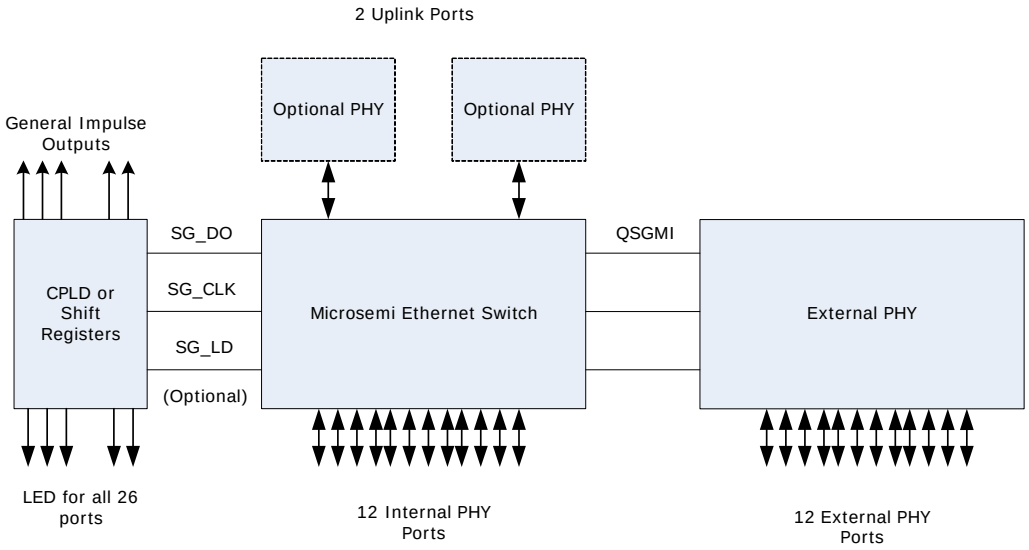
### 2.2.1 Serial LED Benefits

The use of SIO controller improves the support for the serial LED and offers the following customer benefits:

- The SIO controller supports up to 128 serial outputs from the SIO controller, which can be used for serial LEDs for both internal and external PHY ports. In addition, customized LEDs, such as a temperature alarm, fan speed indication, software alive indications, and so on, can all be supported by the 2-pin serial LED interface offered from the SIO controller.
- Using the SIO controller, the number of ports enabled and the number of bits per port in the serial LED bit stream are programmable. For example, a 24-port switch application with 2 LEDs per port only needs 48 (24 times 2) bit shift registers.

The following illustration shows the serial LED deployment of a 24/26-port Microsemi switch.

***Figure 3 •*** **Serial LEDs for a 24/26 Port Switch**




## 2.2.2 Serial Output/LED Example

Serial LEDs only use the serial outputs from the SIO controller. If serial LEDs use less than the total 128 bits, then the unassigned bits can be used for other general purpose outputs, such as output signals for controlling TxDisable signals for SFP modules or driving a 7-segment display for showing the chipset die temperature.

The external circuits for the serial LED implementation are as simple as a single chain of serial-to-parallel shift registers. The number of shift registers must equal the number of the enabled ports times the port width.

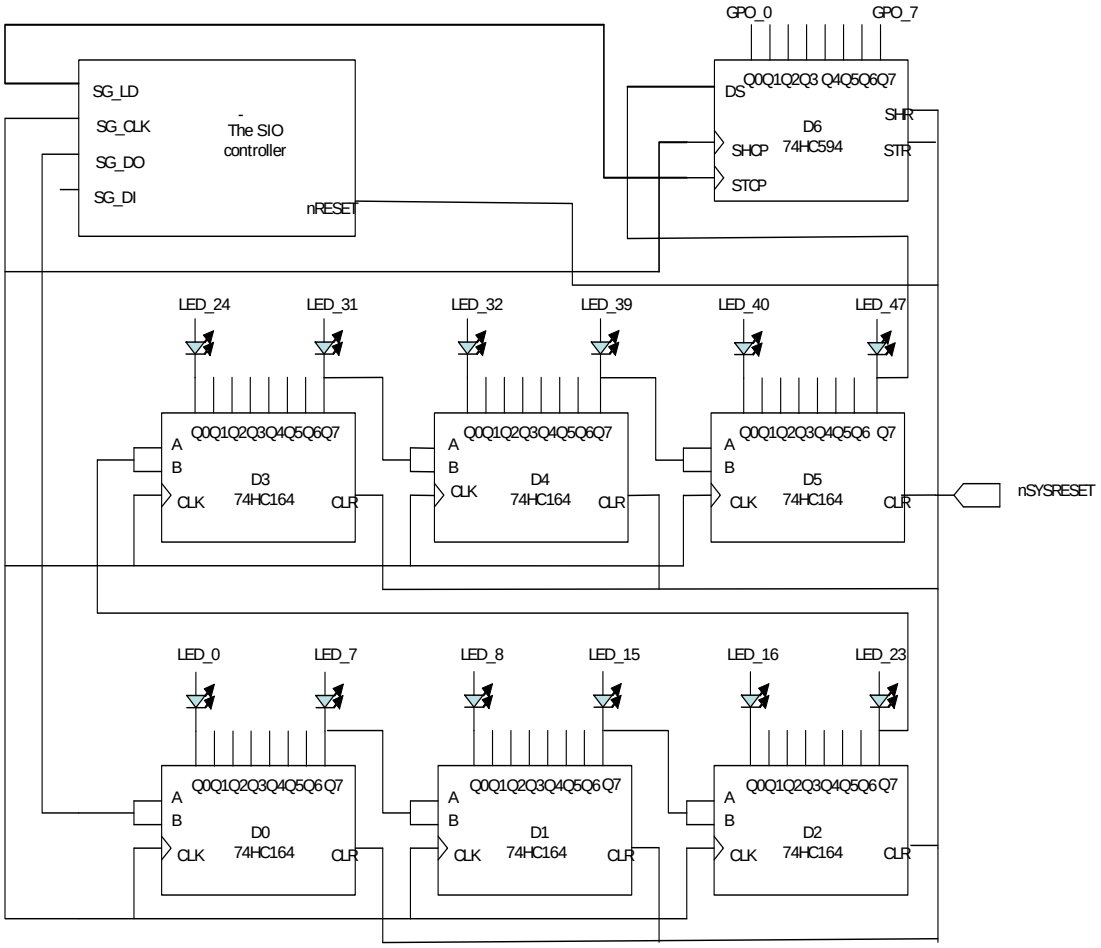
In the 74 series, different serial-to-parallel shift registers have different features to meet the customer's need. A common requirement is that the shift register should shift in data on the rising edge of the clock. The following is a list of commonly-used serial-to-parallel shift registers.

- 74164—an 8-bit serial-to-parallel shift register without load capability and tri-state output. It can be used for serial LED.
- 74594—an 8-bit serial-to-parallel shift register with load capability. It uses a separate load signal to latch shift register values to the storage register. The storage register holds the values for the parallel output pins. The 74594 can be used for outputs that are not tolerant of output toggling during shifting (for instance, TxDisable output for SFP modules) so that SG_LD can be used to load data after a complete burst. The 74594 has independent shift and storage register clear inputs. These clear inputs can be connected to the system reset signal to ensure parallel outputs are in a known state after reset. The 74594 does not support tri-state outputs—as the outputs are always enabled.
- 74595—an 8-bit serial-to-parallel shift register with load capability and tri-state outputs. The 74595, like the 74594, uses a separate load signal to latch shift register values to the storage register similar to the 74594. The difference is that tri-state outputs are controlled by an output enable (/OE) signal. When the 74595 is used for serial LEDs, it is possible to pulse gate the parallel LED outputs by using a pulse width modulation (PWM) control signal. There is a PWM output on the switch with programmable clock frequency and duty cycle. The signal is initially for the purpose of controlling fan speed. If the PWM output is not used for controlling fan speed, then it can be connected to the /OE input of the 74595 to adjust the LED brightness. By changing the duty cycle of the signal, the LED brightness will change. Please note that the clear input for the 74595 only clears the shift register, but not the storage register. So the storage register is undefined until the first load pulse is applied, and the shift register values are loaded to the storage register.

The following illustration shows a system with 48 serial LEDs and 8 general purpose outputs. Six 74HC164 (D0 through D5) are used for the 48 LEDs, and a 74HC594 (D6) is used for the 8 general purpose outputs.

28 SIO ports are enabled, and the port width is set to 2. SIO port0 through port23 are used for serial LEDs (port0 bit0 for LED_0, port0 bit1 for LED_1, port1 bit0 for LED_2, port1 bit1 for LED3, …, port 23 bit 0 for LED_46, and port 23 bit1 for LED_47). SIO port24 through port27 are used for general purpose outputs (port 24 bit0 for GPO_0, port_24 bit1 for GPO_1, …, port 27 bit0 for GPO_6, and port27 bit1 for GPO_7).

***Figure 4 •*** **48 LEDs plus 8 Serial Outputs**

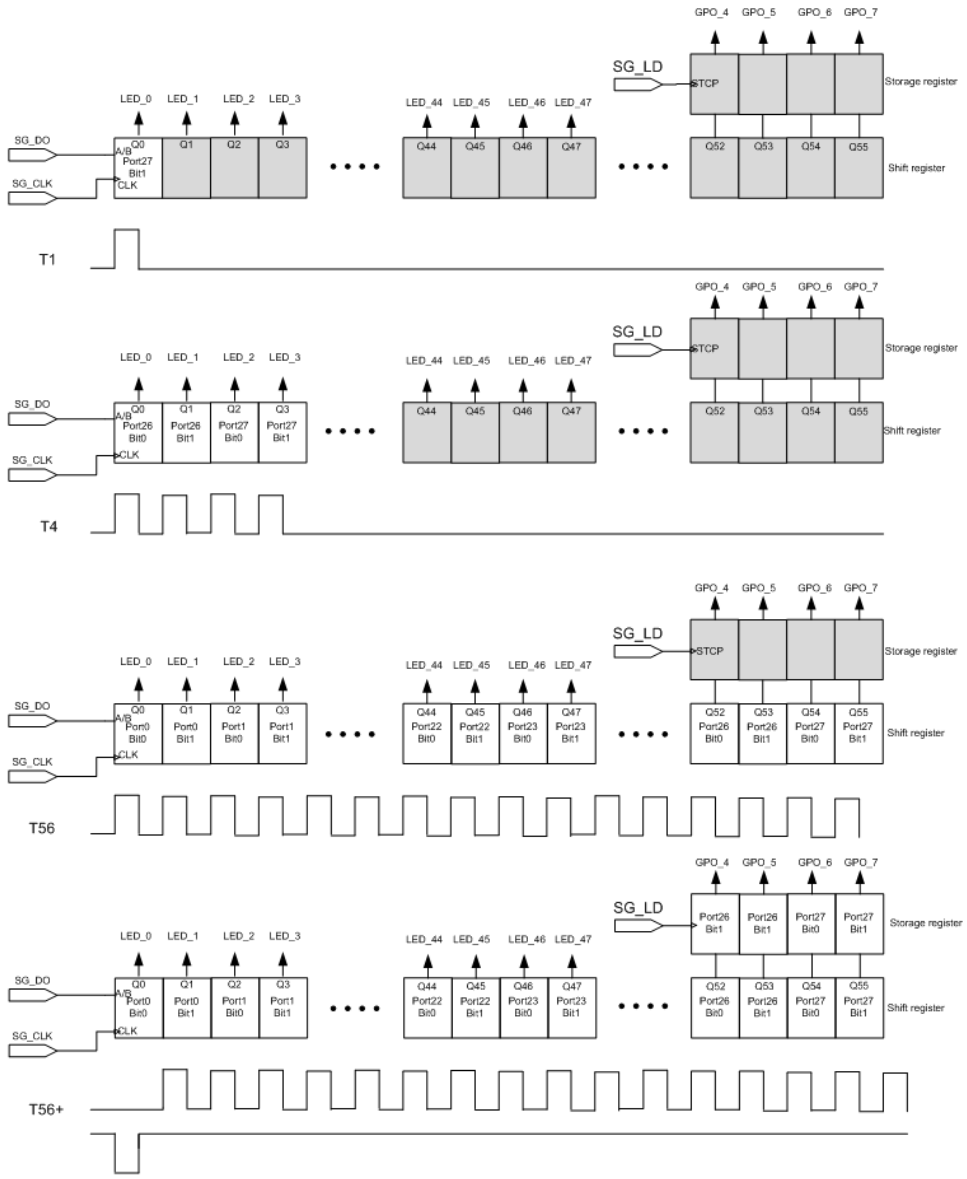



The following illustration shows a burst of serial output data being shifted through the shift register chain. The bit order within the serial stream is in default order—the highest bit (port27 bit1) is shifted out at the first clock, followed by port27 bit0, port26 bit1, …, port0 bit1, and port0 bit0. The register bits in grey hold values from the last burst of data, or default values following reset.

T1 represents the time when only one clock has been sent out. Port27 bit1 is sent out on SG_DO along with the first clock. It will be latched into LED_0, but the destination of port27 bit1 is GPO_7 (the last bit in the shift register chain). It is only at T56 that port27 bit1, together with all other bits, finally reaches the destination through the whole chain. After the assertion of SG_LD, port27 bit1 can then be latched from the shift register to the storage register and output on pin GPO_7. All shift registers hold their values during the burst gap until the next burst of clock and data.

LED_0 through LED47 are not stable from T1 to T56. However, the human eye cannot detect the toggling if the serial clock is fast enough and the burst gap is long enough.

***Figure 5 •*** **Serial Output Bits Shifting Through the Serial Register Chain**




## 2.2.3 Configuring for Serial Output/LED Timing

Several timing parameters must be configured to customize the SIO controller to meet the customer's requirements and match the customer's hardware. The parameters include the number of ports (the enabled/disabled state for each of the 32 ports), port width, serial clock frequency, burst gap, signal polarity, and bit ordering. The following registers are relevant for SIO output timing configuration.

- SIO_PORT_CONFIG
- SIO_PORT_ENABLE
- SIO_CONFIG
- SIO_CLOCK

Refer to the Microsemi Ethernet switch datasheet for the detailed descriptions of each register. Only the serial LED timing-related register fields and their usages are as follows:

- Clock frequency of the SIO output clock (signal SG_CLK)—SG_CLK is used for both serial input and output. The clock frequency of that signal is configured through register SIO_CLOCK::SIO_CLK_FREQ (bits 11:0). The resulting SG_CLK frequency is the system clock

divided by the register value. For example, if the system clock of the switch is 250 MHZ and SIO_CLK_FREQ is set to 0xA, the output frequency will be 25 MHz. A faster SIO_CLK makes LED toggling along the shift register chain less noticeable. The customer must ensure that the configured SG_CLK frequency is within the specification of the selected shift register devices. Setting SIO_CLK_FREQ to 0 disables clock division, resulting in SIO_CLK being 250 MHz. Setting SIO_CLK_FREQ to 1 is illegal.

- SIO burst gap—the burst gap length is programmable through register SIO_CONFIG::SIO_BURST_GAP (bits 11:7) in approximately 1 ms steps. Setting the register field to 0 results in a 1 ms burst gap, and 0x1F results in a 33 ms burst gap. Burst gap can also be disabled by setting register bit SIO_CONFIG::SIO_BURST_GAP_DIS. In that case, frames of serial output data follow one by one without a gap between them. A shorter burst gap results in more frequent serial output updates and, thus a better visual effect for a blinking LED, especially at a fast blinking frequency (such as, 20 Hz). The cost is a little more power consumption.
- SIO burst mode—SIO supports both single and continuous bursts. In single burst mode, a frame of serial data is only sent out once on demand (after register bit SIO_CONFIG::SIO_SINGLE_SHOT is set), which is not very useful for dynamic serial LED applications with blinks. Continuous burst mode is usually used for serial LED applications, so register SIO_CONFIG::SIO_AUTO_REPEAT should be set to enable that mode.
- Port number and port width—these parameters decide how many bits will be present in every frame of serial data. Port number is configured through a 32-bit register, SIO_PORT_ENABLE, with each bit corresponding to a port. Ports can be individually enabled/disabled by setting/clearing the corresponding register bit. Port width is configured centrally for all ports through register SIO_CONFIG::SIO_PORT_WIDTH (bits 3:2). When set to 0, only bit0 for each enabled port is enabled. When SIO_CONFIG::SIO_PORT_WIDTH is set to 2, all 4 bits (bit3, bit2, bit1, and bit0) are enabled.
- Serial LED output bit order—the default order of the output bit stream is in the order of portN bit3, portN bit2, …, port0 bit1, port0 bit0 (see Table 1, page 2). This can be reversed by setting register SIO_CONFIG::SIO_REVERSE_OUTPUT, such that the bit order changes to port0 bit0, port0 bit1, …, portN bit2, portN bit3.
- SG_LD polarity—the polarity of the serial data load signal SG_LD, can be programmed to be active high or low through register bit SIO_CONFIG::SIO_LD_POLARITY. The purpose is to adapt different types of load-capable shift registers.

### 2.2.4 Serial Output Modes

Serial output modes are the supported behaviors for each individual LED. Serial output mode is programmed for each serial output bit through a group of registers, SIO_PORT_CONFIG[n]. There are 32 registers in total. SIO_PORT_CONFIG[n] corresponds to a serial output portn. Only bits 11:0 of SIO_PORT_CONFIG are used for programming serial output mode for 4 bits. For example, SIO_PORT_CONFIG[0]::BIT_SOURCE_3(bits 11:9) configures the serial output mode for port0 bit3, SIO_PORT_CONFIG[0]::BIT_SOURCE_2(bits 8:6) configures the serial output mode for port0 bit2, SIO_PORT_CONFIG[0]::BIT_SOURCE_1(bits 5:3) configures the serial output mode for port0 bit1, and SIO_PORT_CONFIG[0]::BIT_SOURCE_1(bits 2:0) configures the serial output mode for port0 bit0.

The three bits for each serial output bit are coded for the following 8 serial output modes:

- Force “0”
- Force “1”
- Blinking mode 0
- Blinking mode 1
- Link activity blinking mode 0
- Link activity blinking mode 1
- Link activity blinking mode 0 inversed polarity
- Link activity blinking mode 1 inversed polarity

Mode 1 and mode 2 are forced modes that assign a fixed value to the corresponding output bit in the serial output stream. The fixed value, when converted to parallel by the external shift registers, results in a fixed voltage level for turning on/off the connected LED according to the polarity of the external LED. The forced modes display static events like link status (linkup or linkdown), speed (10M, 100M, 1000M), and duplex of an Ethernet port.

**Note:** The hardware does not collect link status, speed, and duplex information from any Ethernet port and map the serial output bits automatically. Software must get that information and program the output mode field. For example, when serial output port0 bit1 is intended to be used as link 1000M indication for Ethernet port module0, software must check link status for Ethernet port module0 periodically. Only when software detects Ethernet port module0 has established a 1000M link, it then updates SIO_PORT_CONFIG[0]::BIT_SOURCE_1 to 0 to turn on the LED, assuming the external LED circuit is active low.

Mode 3 and mode 4 are two blink modes. Blinking modes are used to make the serial output bits to blink at a fixed rate. The SIO controller features two blink modes that can be set independently. The following table shows the supported blinking frequencies. Two blink modes are able to convey different information and can be programmed at different blinking frequencies so that some LEDs can blink faster than the others, which is useful to give different levels of alarm by using different blinking frequency.

***Table 1 •*** **Blinking Modes**

| Mode | Description |
|---|---|
| Blink mode 0 | 0: 20 Hz blink frequency<br>1: 10 Hz blink frequency<br>2: 5 Hz blink frequency<br>3: 2.5 Hz blink frequency |
| Blink mode 1 | 0: 20 Hz blink frequency<br>1: 10 Hz blink frequency<br>2: 5 Hz blink frequency<br>3: Burst toggle |

**Note:** The blink frequency will vary by chips.

Blinking frequencies for blink mode 0 and 1 can be programmed through register SIO_CONFIG::SIO_BMODE_0 (bits 19:18) and SIO_CONFIG::SIO_BMODE_1 (bits 21:20). The bit fields of the SIO_BMODE_0 or SIO_BMODE_1 register varies across Microsemi's switch chips.

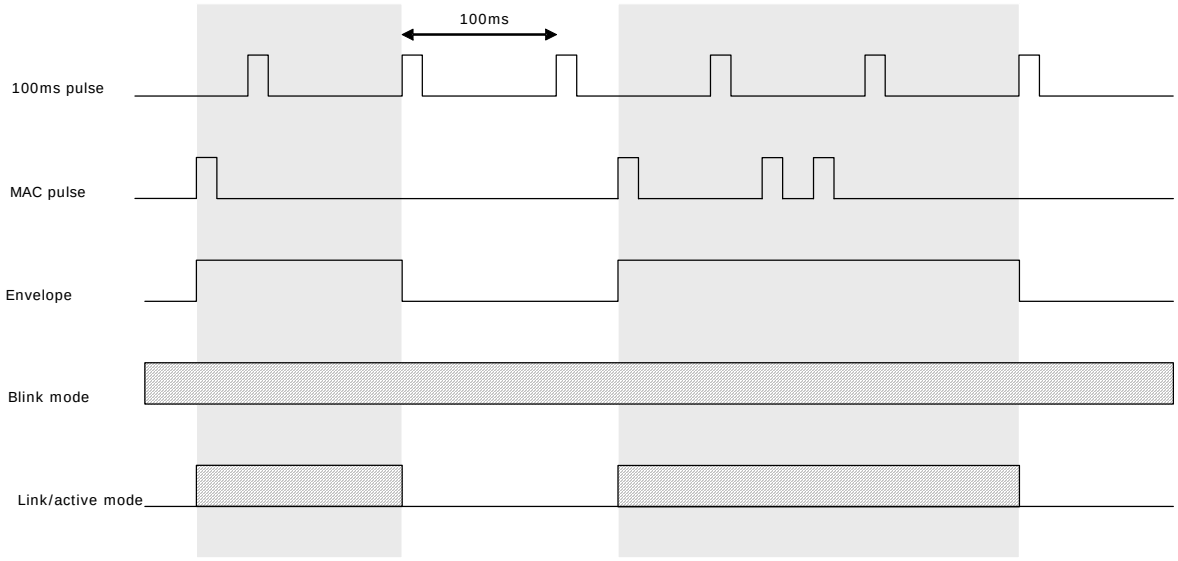
If more than one Ethernet switch/PHY is used in a system, the LED blinking pattern among these devices can go out of sync. To solve the problem the SIO provides a blink counter reset bit (register SIO_CONFIG::SIO_BLINK_RESET) to synchronize the blink patterns between different devices. This is more effective when the different devices run from the same clock source, otherwise the devices will slowly drift apart, thus a blinking counter must be reset periodically.

The SIO controller features three output modes: static mode, blink mode, and link activity mode. For static or blink mode, the serial output ports have no direct relation to the Ethernet port modules. Bits in serial output port 0 can be used for LEDs for any port module because they are basically controlled by software. However, link active modes blink output when there is activity on the corresponding port module (Tx or Rx). The mapping between serial output port and Ethernet port module is 1:1 (static mapping). For example, serial output port 0 connects to port module 0, serial output 1 connects to port module 1, and so on.

The link active modes use an envelope signal to gate the selected blinking pattern (blink mode 0 or blink mode 1). When the envelope signal is asserted, the output blinks. When this is de-asserted, the output is fixed at 0 or 1 depending on the SIO controller's port configurations and the hardware LED's active mode.

To ensure that even a single packet makes a visual blink, an activity pulse from the port module is extended to a minimum of 100 ms. If another packet is sent while the envelope signal is asserted, it is extended by another 100 ms, as shown in the following illustration.

***Figure 6 •*** **Link Active Mode Timing**




## 2.2.5 Software Example for Serial Output/LED

The following pseudo code example supports a 24-port switch with two LEDs per port. LED0 (corresponds to bit0 in the serial stream) is 10/100link/activity and LED1 (corresponds to bit1 in the serial stream) is 1000link/activity. The blink frequency for LED0 and LED1 is 5 Hz. Also the example presents a software alive LED and a die temperature alarm LED. The software alive LED blinks at 0.5 Hz to indicate that the software is running and the die temperature alarm LED has the following states:

- UnAlarm—alarm LED off, temperature below alarm_threshold_0
- Alarm_level_1—LED blinks at 2.5 Hz, temperature between alarm_threshold_0 and alarm_threshold_1
- Alarm_level_2—LED blinks at 5 Hz, temperature above alarm_threshold_1

In this example, it is assumed that all the LEDs have been designed as active low by hardware.

The following code initializes the SIO and sets the serial output mode for all LEDs dynamically.

```
Write(SIO_CLOCK::SIO_CLK_FREQ, 0x64); /* Set SG_CLK to be 2.5M,250M/100 = 2.5M
*/
Write(SIO_CONFIG::SIO_BURST_GAP, 0x1f); /* set length of burst gap to be 33ms
*/
Write(SIO_PORT_ENABLE,0x1ffff); /* enable 25 serial output ports, the 25th port
is for software alive and alarm LED */
Write(SIO_CONFIG::SIO_PORT_WIDTH, 1); /* set port width to 2 */
Write(SIO_CONFIG::SIO_BMODE_1, 2); /* set blink mode 1 frequency, used for
10/100link/activity, 1000Link/activity and temperature Alarm_level_2 */
Write(SIO_CONFIG::SIO_BMODE_0, 3); /* set blinking mode 0 frequency used for
temperature Alarm_level_1 */
/* Turn off all LED, assume the hardware design for all LEDs are active low */
for (n = 0, n <25, n++) {
Write(SIO_PORT_CONFIG[n]::BIT_SOURCE_0, 1); /* force off for LED0 */
Write(SIO_PORT_CONFIG[n]::BIT_SOURCE_1, 1); /* force off for LED1 */
Write(SIO_CONFIG::SIO_AUTO_REPEAT, 1); /* enable continuous bursts mode */
/* Checking port status and setting link/activity LED mode every 100ms */
While(timer_100ms expired){
for (n = 0, n <24, n++) {
if(link status for port[n] changed) {
if(port[n] linkup && port[n] speed == 10M/100M){
Write(SIO_PORT_CONFIG[n]::BIT_SOURCE_0, 5); /* Activate LED0 for
10/100Link/activity */
```

```
Write(SIO_PORT_CONFIG[n]::BIT_SOURCE_1, 1); /* turn off LED1 */
}
else if(port[n] linkup && port[n] speed == 1000M){
Write(SIO_PORT_CONFIG[n]::BIT_SOURCE_1, 5); /* Activate LED1 for
1000Link/activity */
Write(SIO_PORT_CONFIG[n]::BIT_SOURCE_0, 1); /* turn off LED0 */
}
else if(port[n] linkdown)){
Write(SIO_PORT_CONFIG[n]::BIT_SOURCE_0, 1); /* turn off LED0 */
Write(SIO_PORT_CONFIG[n]::BIT_SOURCE_1, 1); /* turn off LED1 */
}
else{ /* No link status change, no action */ }
}
}
}
/* Invert alive LED and update temperature Alarm status once per second */
While(timer_1s expired){
Write(SIO_PORT_CONFIG[24]::BIT_SOURCE_0,(SIO_PORT_CONFIG[24]::BIT_SOURCE_0 +
1)&1);
/* invert alive LED */
If(T<alarm_threshold_0){
Write(SIO_PORT_CONFIG[24]::BIT_SOURCE_1, 1); /* turn off temperature alarm LED
*/
}
else if(alarm_threshold_0<=T<alarm_threshold_1){
Write(SIO_PORT_CONFIG[24]::BIT_SOURCE_1, 2); /* set alarm LED in blinking mode
0 */
}
else{
Write(SIO_PORT_CONFIG[24]::BIT_SOURCE_1, 3); /* set alarm LED in blinking mode
1 */ }
}
```

## 2.3 Using SIO Input

As previously described, the SIO controller has both the output and the input function blocks. Serial input uses a serial interface to extend the number of general purpose input pins. The extended input pins can be used to monitor status from SFP modules. There are three status outputs from each SFP module: Tx_fault, Module_detect, and LOS. The required number of general purpose input pins could be large if customer design supports a lot of SFP modules. Besides SFP module status monitoring, the SIO input can also be used for push buttons, DIP switch configurations, and more. Additional inputs make the switch more configurable and intelligent.

### 2.3.1 Common Settings for Serial Input and Output

The serial input block works simultaneously with the output block. When the serial output data is being shifted out on SG_DO, serial input data is at the same time being shifted into the switch through SG_DI. The serial input data is stored in register S_IN0 through S_IN3 for software read.

The following common settings are shared by the SIO controller serial input and output.

- SIO clock frequency and burst gap configuration—note that SG_LD assertion is followed immediately after a serial clock burst (SG_CLK). At the next serial clock burst, the latched serial inputs can be shifted from the external shift registers into the switch. Longer burst gaps result in longer input latency. The maximum input latency can be 33ms (maximum burst gap). Although input latency can be reduced by disabling burst gap by setting register bit SIO_CONFIG::SIO_BURST_GAP_DIS, but this causes problems for the serial LEDs (outputs) if the shift registers without load capability are used (for example, 74164). Burst gap optimization must address both serial input and output requirements. For more information about configurations for SIO clock and burst gap, refer to datasheet.

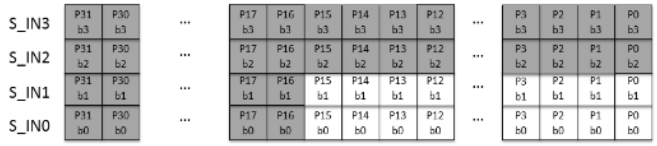
- SIO port number (serial ports enabled/disabled) and port width configuration—the serial port/bit enabling/disabling is effective for both input and output. For more details, refer to the SIO Controller section in the datasheet.
- SIO burst mode—once enabled, single and continuous bursts are effective for both serial input and output. Single burst mode enables the software to read serial input at any time. Due to the input latency, described earlier, two consecutive single burst commands must be issued in order to sample the input values by single burst mode. Due to the first burst, the external serial-to-parallel registers latch the input values. The second burst shifts the input values into the switch. Set the register bit SIO_CONFIG::SIO_SINGLE_SHOT to trigger a single burst. This register is self cleared.
- Polarity of SG_LD—both serial input and serial output use the load signal SG_LD. For serial output, SG_LD latches shift register values to the parallel output pins. For serial input, SG_LD loads parallel input values from the parallel input pins to the shift register. The polarity of SG_LD is configurable through register bit SIO_CONFIG::SIO_LD_POLARITY.

## 2.3.2 Serial Input Bit Order and Format

As shown in Figure 1, page 2, the default order of the input bit stream is port0 bit0, port0 bit 1, …, portN bit 2, portN bit 3. The lowest bit is shifted into the switch first. This can be reversed by setting SIO_CONFIG::SIO_REVERSE_INPUT, which changes the bit order to portN bit 3, portN bit2, …, port0 bit1, port0, port0 bit0. Bit order for serial input and output can be reversed independently.

The sampled serial input data is stored in four 32-bit registers, S_IN0 through S_IN3. The following illustration shows the data format of the four registers when only SIO port 0 through port 15 are enabled with a port width of 2 bits. The bits for disabled ports/bits are undefined. In the following illustration, the bits in grey represent the undefined/disabled bits.

***Figure 7 •*** **Serial Input Data Register**




## 2.3.3 Serial Input and Interrupts

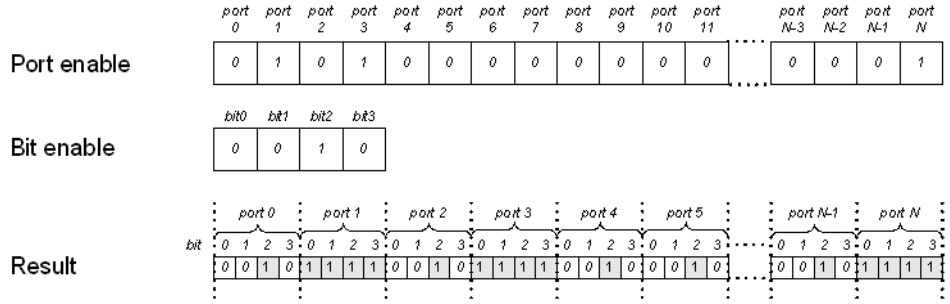
The SIO input can generate interrupts based on the serial input data values. All interrupts are level-sensitive and can be enabled per port and per bit position.

The 32-bit register SIO_PORT_INT_ENA enables interrupts per serial input port. When a bit is set in this register, interrupts for all four input bits of the corresponding serial port are enabled.

The 4-bit register SIO_CONFIG::SIO_INT_ENA enables serial input interrupts per bit position. The lowest bit in SIO_CONFIG::SIO_INT_ENA represents bit0 of all the SIO ports and the highest bit in SIO_CONFIG::SIO_INT_ENA represents bit3 of all ports. Setting a bit in SIO_CONFIG::SIO_INT_ENA will enable interrupts for the corresponding input bit for all ports. The result of the per port and per bit position interrupts are shown in the following illustration.

Interrupt polarity is configured for each serial input bit in SIO_INT_POL_0 through SIO_INT_POL_3. Bit ordering is the same as shown in the illustration. Setting a bit configures an interrupt at logic value “0” for the serial input bit. Clearing a bit configures it to an interrupt at logic value “1”.

***Figure 8 •*** **Serial Input Interrupt Masks**




The SIO controller has one interrupt output connected to the main interrupt controller that gets asserted when one or more interrupts are active. To determine which input bit is causing the interrupt, the CPU must read the “sticky-bit” interrupt registers SIO_INT_REG_0 through SIO_INT_REG_3. The registers have one bit per serial input bit and can only be cleared by software. A bit is cleared by writing a “1” to the bit position. The interrupt output will remain high until all interrupts in SIO_INT_REG_x are cleared.

## 2.3.4 SIO Input for LOS

The SIO controller can propagate loss of signal detection inputs directly to the port module’s signal detection input. This is useful when, for example, SFP modules are used in the design. The mapping between SIO ports and port modules is the same as for link activity outputs (port 0 connects to port module 0, port 1 connects to port module 1, and so on).

Only bit 0 of each serial input port can be configured to forward directly to the corresponding port module’s loss of signal input. Enabling serial input for LOS can be configured per port or per port module. The following signal detect configuration bits for the port module should be configured:

- PCS1G_SD_CFG::SD_ENA must be set to enable Signal Detect for the PCS of the port module. When this bit is cleared, the PCS assumes an active Signal Detect at all times.
- PCS1G_SD_CFG::SD_SEL must be set to enable serial input bit 0 as the signal detect input to the PCS. Otherwise, the signal detect signal for the PCS is generated internally.
- PCS1G_SD_CFG::SD_POL configures the polarity of the signal detect input.This bit must be cleared because LOS from an SFP module is active low. Additionally, the corresponding interrupt polarity registers of the serial input bit in SIO_INT_POL_0 through SIO_INT_POL_3 must be cleared.

## 2.3.5 Serial Input Examples

External parallel-to-serial shift registers must sample and store the parallel input values before they are shifted into the switch. The 8-bit shift register 74165 can be used for that purpose. It latches parallel values into the shift register on the rising edge of /PL (typically connected to SG_LD) and shifts data out on the rising edge of CP (typically connected to SG_CLK).

Because the same serial ports/bits must be enabled at the same time for serial input and output, the shift register width for input must theoretically be equal to the shift register width for output. The number of serial outputs is generally more than the number of serial input required in a design, but this does not necessarily mean that the design must contain the same number of physical shift registers for serial input as it does for serial output. Several techniques are described in the following paragraphs to reduce the number of shift registers for serial input.

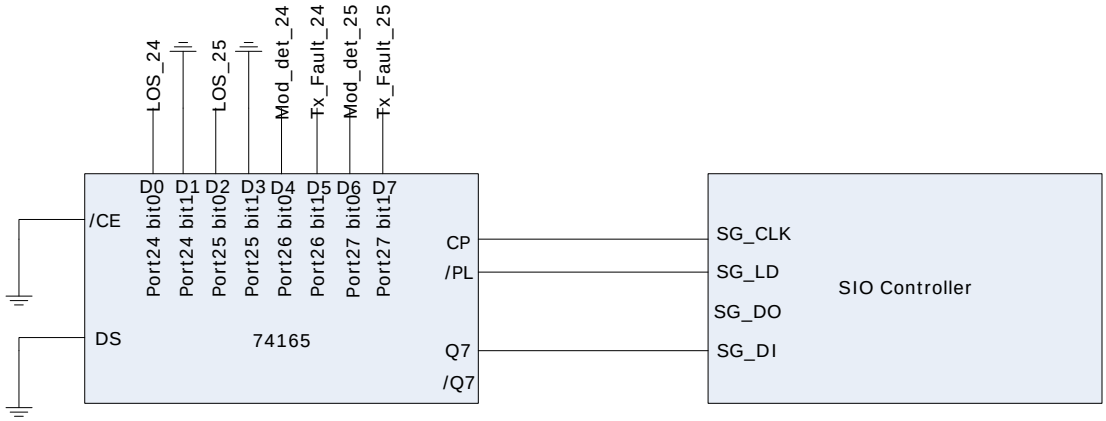
A general recommendation is to use the lowest enabled ports/bits for serial input when serial input is in default bit order (register bit SIO_CONFIG::SIO_REVERSE_INPUT cleared) or use the highest ports/bits for serial input when serial input is in reverse bit order (register bit SIO_CONFIG::SIO_REVERSE_INPUT set). In that case, the shift register width can be limited to the number of actually required serial input bits. The idea is to only have external shift registers for the required serial inputs by arranging the required serial inputs to be shifted into the switch earlier than other "don't care" serial inputs, which do not necessarily have shift registers for them because the values are not used.

For example, assume that the serial output/LED requirements are the same as in the Serial Output/LED Example, page 4 such that 28 serial ports with 2-bit port width must be enabled. Now, there is the serial input requirement of 6 inputs for two SFP modules on port module 24 and 25. The signals Tx_fault, Module_detect, and LOS from the two SFP modules must be monitored through serial input. Because the two LOS signals must be mapped to serial port 24 bit 0 and serial port 25 bit 0, use the highest ports/bits for the six inputs with reversed serial input bit order.

The following illustration shows how this can be realized by using only one 8-bit parallel-to-serial shift register.

In this example, the six input signals from the two SFP modules will be correctly input to the switch while all the other serial input bits will all get a value "0" because signal DS is grounded. LOS_24 and LOS_25 must be mapped to serial port24, bit0 and port25 bit0, respectively.

***Figure 9 •*** **Serial Input for Example 1**

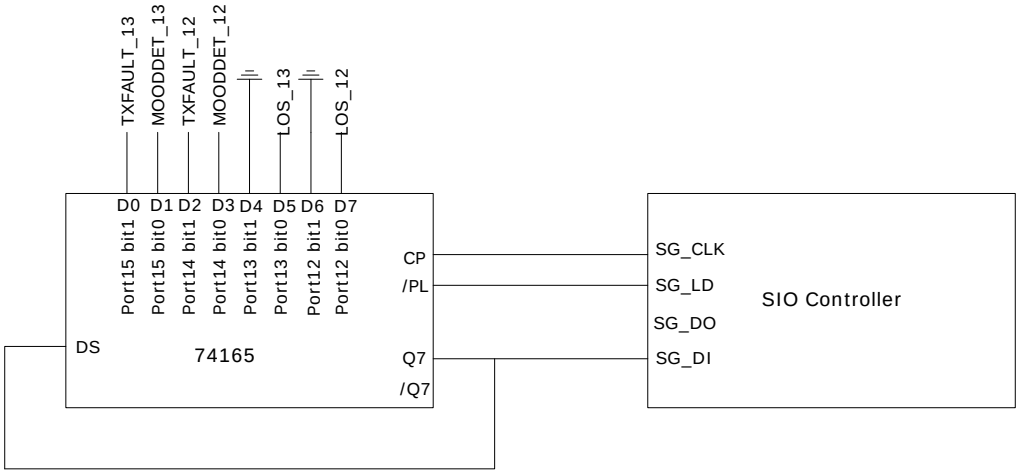



Serial inputs such as Module_detect and Tx_fault from SFP modules or push button inputs are only for software to read (we call them soft inputs) so they can be mapped to any serial port/bit. However, the LOS signal from an SFP module is forwarded directly to the PCS block of the corresponding port module so there is no freedom to place these in any port/bit position at will.

In the preceding example, the two SFP modules are on port 24 and 25, so the highest ports/bits for all the serial input signals can be used. If the two SFP modules must be supported on port module 12 and 13, a technique called shift register loopback can be used.

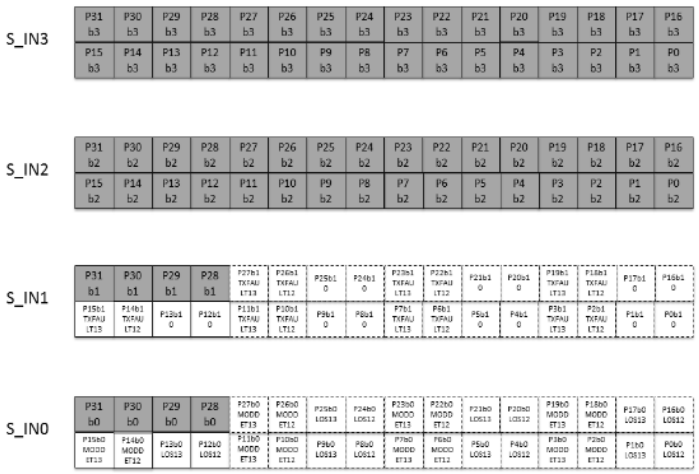
The following illustration shows an example with the same serial output/LED requirements as the previous example, but the six serial inputs from two SFP modules are from port module 12 and 13. Serial output Q7 of the shift register is looped back to its serial input DS such that the same 8-bit values will be repeated on the serial input stream and get duplicated in the internal serial input registers. In this way, the 8 serial inputs can be successfully copied into the correct port/bit positions.

***Figure 10 •*** **Serial Input for Example 2**




The following illustration shows the resulting serial input values in the switch after a complete burst. Bits in grey are disabled bits. Bits in dashed boxes are don't care bits.

***Figure 11 •*** **Input Data with Loopback**




The last example is a special case where only one serial input needs to be read. Whether it is a soft input or a LOS signal from a single SFP for any port module, this signal can be connected directly to SG_DI without a shift register. This single input bit will be read repeatedly by the SIO controller and finally reach the correct bit position where it is needed.