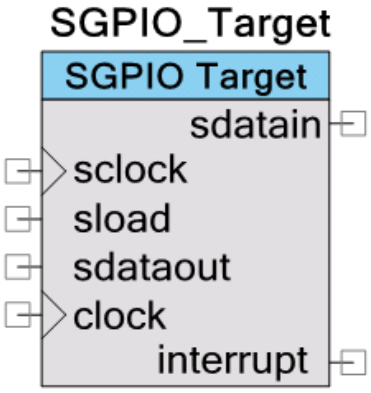
SGPIO Target
v1.30

Features

- Serial GPIO bus Target interface component to be used in conjunction with Serial Attached SCSI (SAS) or Serial ATA (SATA) hard drives
- Compliant with Small Form Factor committee specification SFF-8485
- Supports up to 4 SAS/SATA hard drives
- Support for vendor specific L[3:0] bits
- Support for 64 ms bus timeout function



Contents

General Description

The SGPIO_Target component is an interface component that supports the SFF-8485 Serial GPIO (SGPIO) industry standard, providing support for 4 SAS/SATA hard drives. SGPIO is a 4-wire bus used between a host bus adaptor (HBA) and a backplane. Of the 4 signals, 3 are driven by the SGPIO Initiator in the HBA (sclock, sload and sdataout) and 1 by the SGPIO Target in the backplane controller (sdatain). Typically, the HBA is a storage controller located inside a server, desktop, rack or workstation computer which interfaces with hard disk drives (HDDs) to store and retrieve data.

When to use an SGPIO_Target

The SGPIO_Target component should be used in HDD enclosure applications that support the SFF-8485 SGPIO industry standard. Each SGPIO_Target component supports 4 HDDs. Multiple SGPIO_Targets can be instantiated in the same PSoC device to extend support for more HDDs and more HBAs.

Input/Output Connections

This section describes the various input and output connections for the SGPIO_Target.

Input	May Be Hidden	Description
sclock	N	SGPIO reference clock provided by the SGPIO Initiator, typically located in the HBA. According to the SFF-8485 specification, this clock frequency can be no more than 100 kHz
sload	N	SGPIO load signal primarily used for synchronization purposes between the Initiator and Target. This signal also optionally transmits 4 bits of vendor specific information (termed L[3:0] in the SFF-8485 specification)
sdataout	N	SGPIO Initiator data out. This data is primarily used to send LED indicator data to the SGPIO Target. There are 3 bits defined per drive, resulting in a 12-bit serial stream for a 4 drive application
clock	N	This clock is used for internal logic timing. Must be set at least 8x the frequency of sclock. Typically 1MHz

Output	May Be Hidden	Description
sdatain	N	SGPIO Initiator data in. This data is used to receive data from the SGPIO Target regarding HDD status such as presence, errors etc.
interrupt	N	Active high output pin asserted when a new 12-bit frame of sdataout is received from the Initiator

Parameters

Drag an SGPIO_Target component onto your design and double-click it to open the Configure dialog.

Basic Tab

This tab is used to set the single operational parameter of the component.

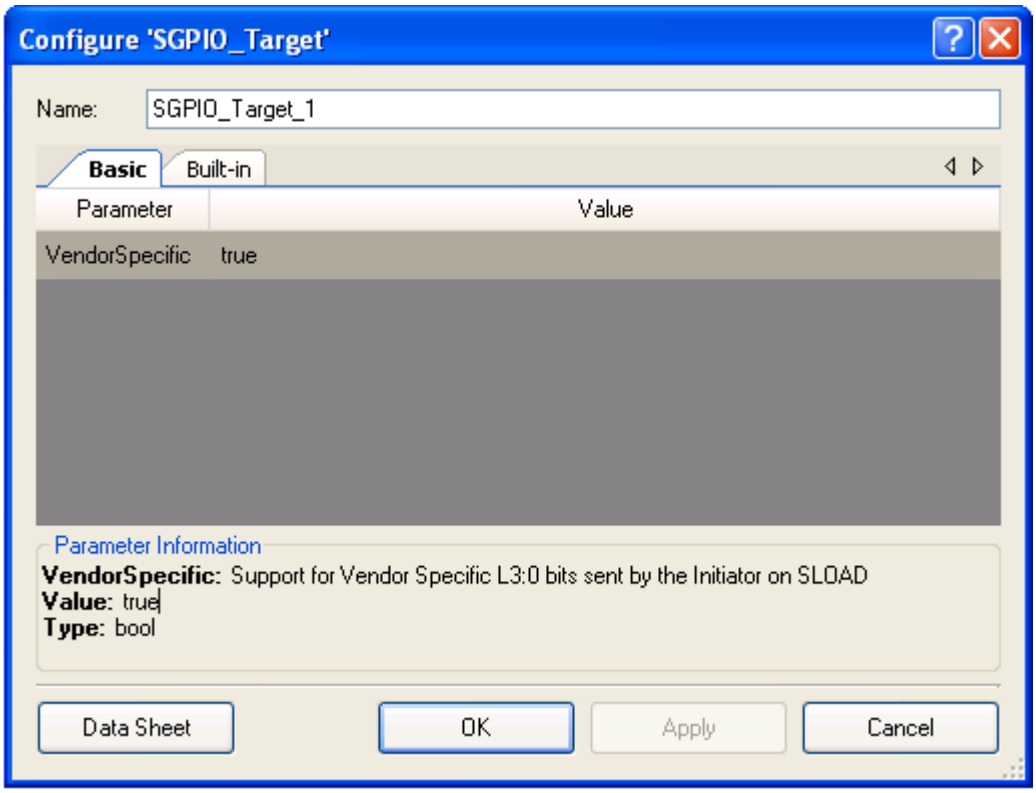


Figure 1 – SGPIO Target Basic Tab

Vendor Specific

This parameter is used to enable or disable capturing of the L[3:0] vendor specific bits transmitted on the sload signal by the SGPIO Initiator immediately following the sload sync pattern. Valid range for this parameter is true or false. Default setting is true.

Placement

There are no specific placement requirements for this component.

Resources

The SGPIO_Target uses the following device resources:

Configuration	Digital Blocks				API Memory (Bytes)	
	Data Paths	Macro Cells	Status Registers	Control/Count7 Registers	Flash	SRAM
Default	2	13	1	2	826	2

Application Programming Interface

Application Programming Interface (API) routines allow you to configure the component using firmware. The following table lists and describes the interface to each function. The subsequent sections cover each function in more detail.

By default, PSoC Creator assigns the instance name SGPIO_Target_1 to the first instance of a component in a given design. You can rename it to any unique value that follows the syntactic rules for identifiers. The instance name becomes the prefix of every global function name, variable, and constant symbol. For readability, the instance name used in the following table is SGPIO_Target.

Function	Description
SGPIO_Target_Start()	Start the component
SGPIO_Target_Stop()	Stop the component and disable hardware blocks
SGPIO_Target_Init()	Initializes the component
SGPIO_Target_Enable()	Enables hardware blocks
SGPIO_Target_EnableInt()	Enables interrupts from the component
SGPIO_Target_DisableInt()	Disables interrupts from the component
SGPIO_Target_GetIntStatus()	Returns pending interrupt status
SGPIO_Target_GetStatus()	Returns all status bits from the component
SGPIO_Target_WriteTxData(uint16 txData)	Writes a new 12-bit sdatain value into the transmit FIFO ready for transmission on the next frame to the SGPIO Initiator
SGPIO_Target_ReadRxData()	Returns the most recently received 12-bit sdataout frame or sload L[3:0] vendor specific data (if enabled). If vendor specific data is enabled, that data is loaded after the sdataout in that frame
SGPIO_Target_ClearTxBuffer()	Resets and empties the transmit FIFO
SGPIO_Target_ClearRxBuffer()	Resets and empties the receive FIFO
SGPIO_Target_ResetTimeout()	Resets the SGPIO bus activity detection logic

Function	Description
SGPIO_Target_GetTimeoutStatus()	Returns the status of SGPIO bus activity detection logic
SGPIO_Target_SaveConfig()	Save current state of component before entering low power mode
SGPIO_Target_RestoreConfig()	Restores previous state of the component after waking from low power mode
SGPIO_Target_Sleep()	Put component into low power mode
SGPIO_Target_Wakeup()	Wake component from low power mode

void SGPIO_Target_Start(void)

Description: Enables the SGPIO_Target component. Calls the Init() API if the component has not been initialized before. Calls Enable().

Parameters: None

Return Value: None

Side Effects: None

void SGPIO_Target_Stop (void)

Description: Disables the SGPIO_Target component.

Parameters: None

Return Value: None

Side Effects: None

void SGPIO_Target_Init (void)

Description: Initializes the SGPIO_Target component.

Parameters: None

Return Value: None

Side Effects: None

void SGPIO_Target_Enable (void)

Description: Enables hardware blocks within the SGPIO_Target component.

Parameters: None

Return Value: None

Side Effects: None

void SGPIO_Target_EnableInt(void)

Description: Enables generation of the interrupt signal. An active high interrupt signal will be generated when a new 12-bit frame of data has been received from the Initiator on sdataout.

Parameters: None

Return Value: None

Side Effects: None

void SGPIO_Target_DisableInt(void)

Description: Disables generation of the interrupt signal.

Parameters: None

Return Value: None

Side Effects: None

uint8 SGPIO_Target_GetIntStatus(void)

Description: Returns the state of the interrupt signal for polling mode operation

Parameters: None

Return Value: 0 = no interrupt pending
Non-zero = interrupt pending

Side Effects: None

uint8 SGPIO_Target_GetStatus(void)

Description: Returns all status bits of the component

Parameters: None

Return Value:

Bit Field	Information
b0	Sync bit. 1=SGPIO bus synchronized to sload. 0=not synchronized
b1	SGPIO bus activity sticky bit. 1=no activity, 0=activity
b2	Receive FIFO not empty. 1=at least one 12-bit word is in the receive FIFO, 0=no data in the receive FIFO. Receive FIFO is 4x 12-bit words deep
b3	Transmit FIFO not full. 1=the transmit FIFO has space for at least one 12-bit word. 0=the transmit FIFO is full. Transmit FIFO is 4x 12-bit words deep
b7:4	Reserved. Returns all zeroes

Side Effects: None

void SGPIO_Target_WriteTxData(uint16 txData)

Description: Writes a new 12-bit sdatain data word into the transmit FIFO ready for transmission on the next frame. The FIFOs are 4 deep (4x 12-bits), however to remain properly synchronized with the SGPIO initiator, it is recommended that this API is called once per interrupt generated.

Parameters: uint16 txData

Bit Field	Information
b11:0	12-bit data for transmission to Initiator on sdatain in next frame (3 bits per drive). Drive 0 data is in the least significant bits.
b15:12	Reserved. Don't care.

Return Value: None

Side Effects: None

uint16 SGPIO_Target_ReadRxData(void)

Description: Returns the most recently received 12-bit sdataout frame and sload vendor specific L[3:0] data (if enabled). If vendor specific data is enabled, that data is loaded after the sdataout frame is loaded into the receive FIFO. That is, when an interrupt is received from this component and vendor specific data is enabled, the 1st call to this API will return the sdataout data and the 2nd call will return L[3:0] vendor specific data. If vendor specific data is disabled, this API should be called only once per interrupt.

Parameters: None

Return Value:

Bit Field	Information
b11:0	When the data returned is 12-bit sdataout data received from the Initiator, there are 3 bits per drive available. Drive 0 data is in the least significant bits. When the data returned is vendor specific L[3:0] data, that data is returned in the 4 least significant bits. The most significant bit (b11) will always be set to 1 indicating that the sload sync pulse was also received.
b15:12	All zeroes

Side Effects: Calling this API will automatically de-assert the interrupt signal

void SGPIO_Target_ClearTxBuffer(void)

Description: Resets and empties the transmit FIFO.

Parameters: None

Return Value: None

Side Effects: None

void SGPIO_Target_ClearRxBuffer(void)

Description: Resets and empties the receive FIFO.

Parameters: None

Return Value: None

Side Effects: Calling this API will automatically de-assert the interrupt signal

void SGPIO_Target_ResetTimeout(void)

Description: Resets the SGPIO bus activity logic in the verilog hardware. To meet the SFF-8485 specification of detecting 64 msec of no activity, this API should be called once per msec. At the subsequent msec interval, the SGPIO_Target_GetTimeoutStatus() API should be called to determine if there has been any bus activity in the past msec. If no activity was detected, a “no activity” counter variable in SRAM should be incremented. If no activity continues for 64 msec, then the timeout condition has been detected.

Parameters: None

Return Value: None

Side Effects: None

uint8 SGPIO_Target_GetTimeoutStatus(void)

Description: Returns the status of SGPIO bus activity detection logic

Parameters: None

Return Value: 0 = SGPIO bus activity detected since the last SGPIO_Target_ResetTimeout() API call
Non-zero = no SGPIO bus activity detected

Side Effects: None

void SGPIO_Target_SaveConfig(void)

Description: This API is used to save the configuration of the SGPIO_Target component prior to PSoC entering low-power sleep mode. Currently, the only parameter stored is the interrupt enable state of the component.

Parameters: None

Return Value: None

Side Effects: None

void SGPIO_Target_RestoreConfig(void)

Description: This API is used to restore the configuration of the SGPIO_Target component after PSoC exits low-power sleep mode. Currently, the only parameter restored is the interrupt enable state of the component.

Parameters: None

Return Value: None

Side Effects: None

void SGPIO_Target_Sleep(void)

Description: This API is used to configure the SGPIO_Target component for minimum power consumption prior to PSoC entering low-power sleep mode. Saves the enable state of the component, stop it and backup non-retention data to SRAM.

Parameters: None

Return Value: None

Side Effects: None

void SGPIO_Target_Wakeup(void)

Description: This API is used to configure the SGPIO_Target component for active power consumption after PSoC exits low-power sleep mode. Restore non-retention data from SRAM. Re-initialize the component and re-enable it if it was enabled before going to sleep

Parameters: None

Return Value: None

Side Effects: None

Sample Firmware Source Code

Some C language examples demonstrating the basic functionality of the SGPIO_Target reference component are shown below. These examples assume that the component has been placed in a design with the default name "SGPIO_Target ".

Note: If you rename your component you must also edit the example code as appropriate to match the API names to the component name you specify.

Example 1: Polling Mode Operation with VendorSpecific disabled. Code in main.c:

```
#include <device.h>
#include "SGPIO_Target.h"

uint16 sdout;

void main()
{
   SGPIO_Target_Start();                          /* Start SGPIO_Target */
   for(;;)
   {
         while(!SGPIO_Target_GetIntStatus());     /* Wait for receive data */
         sdout = SGPIO_Target_ReadRxData();       /* Get Initiator data */
         SGPIO_Target_WriteTxData(0x0249);        /* 001 for all 4 drives */
   }
}
```

Example 2: Interrupt Mode Operation with VendorSpecific parameter enabled through the component customizer. Assumes an ISR component is added to the top level design and named “SGPIO_INT_1”. Code in main.c:

```
#include <device.h>
#include "SGPIO_Target.h"

uint16 sdout1, sdout2;                 /* Globals used by ISR */

void main()
{
   CYGlobalIntEnable;                  /* Enable CPU interrupts */
   SGPIO_INT_1_Start();                /* Start interrupt component */
   SGPIO_Target_Start();               /* Start SGPIO_Target */
   SGPIO_Target_EnableInt();           /* Enable SPGIO_Target Interrupts */

   while(1){};                         /* Do everything in ISR */
}
```

Code in SGPIO_INT_1.c:

```
CY_ISR(SGPIO_INT_1_Interrupt)
{
    /*  Place your Interrupt code here. */
    /* `#START SGPIO_INT_1_Interrupt` */

   sdout1 = SGPIO_Target_ReadRxData();      /* Read SDOUT Data */
   sdout2 = SGPIO_Target_ReadRxData();      /* Read Vendor Specific L[3:0] Data */
   SGPIO_Target_WriteTxData(0x0249);        /* 001 encoding for all 4 drives */

    /* `#END` */
}
```

Interrupt Service Routine

The SGPIO_Target component does not provide any interrupt service routine APIs automatically. Designers wishing to use interrupts for this component should add an ISR component from the Cypress standard catalog in Creator and follow Sample Firmware Source Code example 2 above.

Registers

The SGPIO_Target has several control and status registers that are used by the firmware APIs to control operation and monitor status. None of these registers are directly accessible by user firmware.

DC and AC Electrical Characteristics

The following values are indicative of expected performance and based on initial characterization data.

SGPIO_Target DC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
Idd	Block current consumption	<tbd>	<tbd>	<tbd>	<tbd>	µA

SGPIO_Target AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
sclock	SGPIO bus clock	Normal operation	32	100	100	kHz

Revision History

This section lists the major changes in the component from the previous version.

Version	Description of Changes
1.0	First release
1.10	Changed symbol terminal names Changed reference timing clock from bus_clock to a clock at least 8x sclock Corrected figure 1 label Corrections to sample code
1.20	Added support for PSoC 5 Added low power management APIs in the _PM file
1.30	Updated for compatibility with PSoC Creator v2.0 Re-classified as “Concept” component