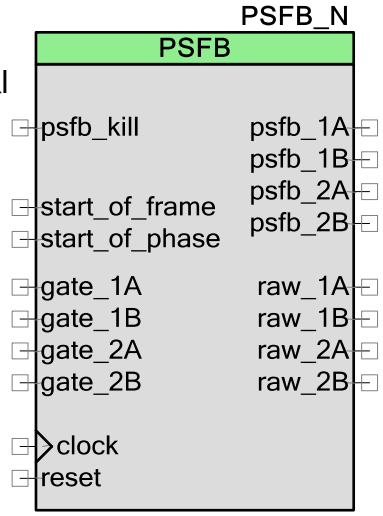
PhaseShiftFullBridge
1.00

Features



- Generates phase-shift modulation from an internal programmable pulse-width modulation source
- Converts external pulse-width modulated signals to phase-shift full-bridge modulation
- Individual signal gating for externally programmable gate delay
- Global asynchronous kill input to immediately disable outputs for handling faults and fast shutdown events

General Description

The PhaseShiftFullBridge component is designed to provide an easy method of generating gate drive signals for phase-shift modulated full-bridge designs. And the modulation technique is built completely from hardware resources.

The component generates two sets of complementary outputs to drive a full-bridge. When the internal modulation is enabled, the relative phase relationship between the waveforms is programmed using an internal pulse-width modulation source. The component uses an 8-bit digital PWM implementation built from UDBs to generate the phase signal for the modulation. The frequency of the modulated signals is half the frequency set in the PWM.

An external modulation input option is available to enable the use of alternate modulation sources, converting an external pulse-width modulation or pulse-position modulation source into phase-shift modulation. Thus analog modulation schemes are possible. Synchronized and interleaved designs are also possible.

An option is provided to include delays in gate signal transitions to prevent shoot-through. This feature is also available for use with soft switching designs to achieve zero-voltage switching.

A kill input is available to immediately disable the outputs. This feature is useful for terminating the outputs immediately in the event of a fault or some other fast cutoff requirement.

When to Use a PhaseShiftFullBridge Component

The most common use of the PhaseShiftFullBridge component is for power applications. Typically phase-shifted full-bridge designs are employed in larger DC-DC power converters that employ some form of soft-switching.

Input/Output Connections

This section describes the various input and output connections for the Phase-Shift Full-Bridge (PSFB) Modulator. Some I/Os may be hidden on the symbol under the conditions listed in the description of that I/O.

Note All signals are active high unless otherwise specified.

psfb_kill – Input

This is an active-low asynchronous input that gates the psfb_1A, psfb_1B, psfb_2A, and psfb_2B outputs of the PSFB component. This is useful for immediately driving the gate signals low, which is commonly necessary when a fault condition occurs in a system. This signal does not stop any internal functionality of the component however. When driven high the outputs will return to the condition where they would be had the output never been 'killed'.

Note This input is available only when the **EnableHardwareKill** parameter is set to **true**.

start_of_frame – Input

The start_of_frame input is provided by an external source. It sets a time reference for modulation. In most situations this is usually a narrow pulse trigger to indicate the starting frame for modulation. The new frame reference is triggered from the negative edge.

Note This input is available only when the **ExternalModulation** parameter is set to **true**.

start_of_phase – Input

The start_of_phase input is provided by an external source. This is the phase modulation input. It is typically derived from a pulse relative to the start_of_frame event. A positive PWM typically follows this model. Note that the output is level sensitive with the output condition latched when this input goes low. Note that the start_of_frame input must also be low.

Note This input is available only when the **ExternalModulation** parameter is set to **true**.

gate_1A – Input

The gate_1A input is a control input for the 1A gate control signal. This signal could be used to gate off the 1A signal independent from any other psfb_XX output signal. Typically this is used to generate rising edge delay control of the psfb_1A output.

Note This input is available only when the **ExternalDelay** parameter is set to **true**.

gate_1B – Input

The gate_1B input is a control input for the 1B gate control signal. This signal could be used to gate off the 1B signal independent from any other psfb_XX output signal. Typically this is used to generate rising edge delay control of the psfb_1B output.

Note This input is available only when the **ExternalDelay** parameter is set to **true**.

gate_2A – Input

The gate_2A input is a control input for the 2A gate control signal. This signal could be used to gate off the 2A signal independent from any other psfb_XX output signal. Typically this is used to generate rising edge delay control of the psfb_2A output.

Note This input is available only when the **ExternalDelay** parameter is set to **true**.

gate_2B – Input

The gate_2B input is a control input for the 2B gate control signal. This signal could be used to gate off the 2B signal independent from any other psfb_XX output signal. Typically this is used to generate rising edge delay control of the psfb_2B output.

Note This input is available only when the **ExternalDelay** parameter is set to **true**.

clock – Input

This is the clock input to the component. The clock is necessary to run the internal digital modulation logic when it is used.

reset – Input

This is an active high on this input to the component disables the included PWM block output.

Note This input is available only when the **ExternalModulation** parameter is set to **false**.

psfb_1A – Output

This is the 1A gate drive signal output.

psfb_1B – Output

This is the 1B gate drive signal output.

psfb_2A – Output

This is the 2A gate drive signal output.

psfb_2B – Output

This is the 2B gate drive signal output.

raw_1A – Output

The raw_1A output is the raw output signal for channel 1A. This output can be used as a trigger for delay generating circuits.

Note This input is available only when the **ExternalDelay** parameter is set to **true**.

raw_1B – Output

The raw_1B output is the raw output signal for channel 1B. This output can be used as a trigger for delay generating circuits.

Note This input is available only when the **ExternalDelay** parameter is set to **true**.

raw_2A – Output

The raw_2A output is the raw output signal for channel 2A. This output can be used as a trigger for delay generating circuits.

Note This input is available only when the **ExternalDelay** parameter is set to **true**.

raw_2B – Output

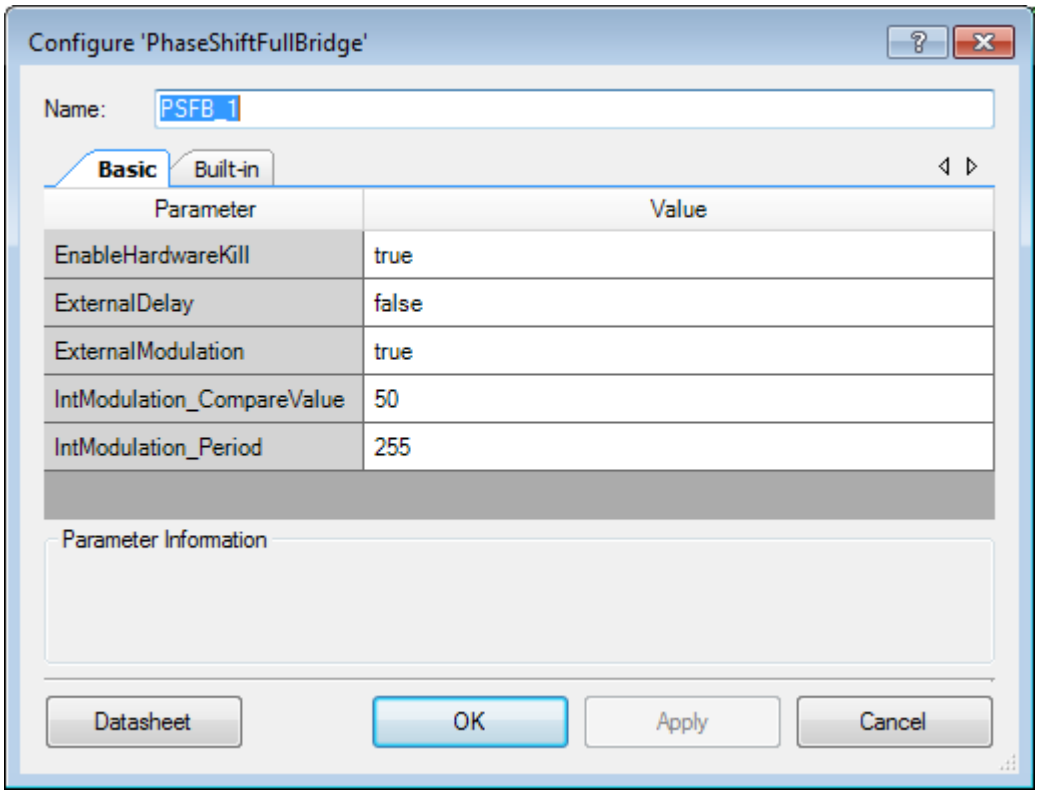
The raw_2B output is the raw output signal for channel 2B. This output can be used as a trigger for delay generating circuits.

Note This input is available only when the **ExternalDelay** parameter is set to **true**.

Component Parameters

Drag a **PhaseShiftFullBridge** onto your design and double click it to open the Configure dialog. Figure 1 shows the Configure dialog.

Figure 1. Configure PhaseShiftFullBridge Dialog



The PhaseShiftFullBridge provides the following parameters.

Basic Options

EnableHardwareKill

This parameter, when set to **true**, enables you to externally disable the modulation outputs using the **psfb_kill** input. This parameter is set to **false** by default.

EnableHardwareKill	true

ExternalDelay

This parameter, when set to **true**, enables you to use externally configured delay circuits to generate gating signals for each channel. This parameter is set to **false** by default.

ExternalDelay	false

ExternalModulation

This parameter, when set to **true**, enables you to use externally configured modulation circuits to generate phase information for the modulator. This parameter is set to **false** by default.

ExternalModulation	true

IntModulation_CompareValue

When the **ExternalModulation** parameter is set to **false**, this parameter sets the compare value for the internally generated modulation. The default value is **50**. This parameter has no affect if **ExternalModulation** is set to **true**.

IntModulation_CompareValue	50
IntModulation_Period	255

IntModulation_Period

When the **ExternalModulation** parameter is set to **false**, this parameter sets the period value for the internally generated modulation. Note that the period of PSFB modulator outputs is twice the period programmed. The default value is **255**. This parameter has no affect if **ExternalModulation** is set to **true**.

IntModulation_CompareValue	50
IntModulation_Period	255

Clock Selection

There is no internal clock in this component. You must attach a clock source.

WARNING When configured to use the fixed-function block in the device, the PWM component has the following restrictions:

- The clock input must be from a local clock that is synchronized to the bus clock or directly sourced from the bus clock (configure the clock type as Existing and the source as BUS_CLK).
- If the frequency of the clock matches the bus clock, then the clock must be a direct connection to the bus clock (again configure the clock type as Existing and the source as BUS_CLK). A local clock with a frequency that matches the bus clock generates an error during the build process.

Resources

The PhaseShiftFullBridge uses the following device resources.

Configuration	Digital Blocks				
	Datapaths	Macro cells	Status Registers	Control Registers	Counter7
ExternalModulation Disabled	1	7	1	1	-

Configuration	PSoC 3 (Keil_PK51)		PSoC 5 (GCC)		PSoC 5LP (GCC)	
	Flash (Bytes)	RAM (Bytes)	Flash (Bytes)	RAM (Bytes)	Flash (Bytes)	RAM (Bytes)
ExternalModulation Disabled	242	6	440	9	392	9

Application Programming Interface

Application Programming Interface (API) routines allow you to configure the component using software. 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 "PSFB_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 "PSFB."

Functions

Function	Description
PSFB_Init()	Initializes the component with default customizer values.
PSFB_Start()	Initializes the component with default customizer values and enables the modulation operation.
PSFB_Stop()	Disables the modulation operation.
PSFB_Enable()	Enables the PSFB block operation.
PSFB_Sleep()	Stops and saves the user configuration of the component.
PSFB_Wakeup()	Restores and enables the user configuration of the component.
PSFB_SaveConfig()	Saves the current user configuration of the component.
PSFB_RestoreConfig()	Restores the current user configuration of the component.
PSFB_ReadControlRegister()	Returns the current state of the control register.
PSFB_WriteControlRegister(control)	Sets the bit field of the control register.
PSFB_ReadPeriod()	Reads the period value.
PSFB_WritePeriod(val)	Writes the period value.
PSFB_ReadCompare()	Reads the compare value.
PSFB_WriteCompare(val)	Writes the compare value.

void PSFB_Init(void)[1]

Description: Initializes the parameters to those set in the customizer placed on the schematic.

Parameters: None

Return Value: None

Side Effects: All registers will be reset to their initial values. This reinitializes the component.

void PSFB_Start(void)[1]

Description: Initializes the component with default customizer values. This also enables the PSFB operation by setting the enable bit of the control register.

Parameters: None

Return Value: None

Side Effects: Sets the enable bit in the control registers of the included PWM.

void PSFB_Stop(void)[1]

Description: Disables the PSFB operation by resetting the seventh bit of the control register.

Parameters: None

Return Value: None

Side Effects: Clears the enable bit in the control register.

void PSFB_Enable(void)[1]

Description: Enables the component block operation by setting the seventh bit of the control register.

Parameters: None

Return Value: None

Side Effects: None

void PSFB_Sleep(void)[1]

Description: Stops and saves the user configuration.

Parameters: None

Return Value: None

Side Effects: None

void PSFB_Wakeup(void)[1]

Description: Restores and enables the user configuration.

Parameters: None

Return Value: None

Side Effects: None

void PSFB_SaveConfig(void)[1]

Description: Saves the current user configuration of the component. The period, compare, counter, and control register values are saved.

Parameters: None

Return Value: None

Side Effects: None

void PSFB_RestoreConfig(void)[1]

Description: Restores the current user configuration of the component.

Parameters: None

Return Value: None

Side Effects: None

uint8 PSFB_ReadControlRegister(void)[1]

Description: Returns the current state of the control register of the component.

Parameters: None

Return Value: uint8: Current control register's value

Side Effects: None

void PSFB_WriteControlRegister(uint8 control)[1]

Description: Sets the bit field of the control register.

Parameters: uint8 control: Control register bit field,

The status register bits are:

[7] : PWM Enable
[6] : Reset
[5:3] : Compare Mode 2
[2:0] : Compare Mode 2

Return Value: None

Side Effects: None

uint8 PSFB_ReadPeriod(void)[1]

Description: Reads the period value used by the component.

Parameters: None

Return Value: uint8: Period value

Side Effects: None

void PSFB_WritePeriod(uint8 period)[1]

Description: Writes the period value used by the component.

Parameters: uint 8 period: the new period value

Return Value: None

Side Effects: None

uint8 PSFB_ReadCompare(void)[1]

Description: Reads the compare value for the compare output.

Parameters: None

Return Value: uint8: Current compare value

Side Effects: None

void PSFB_WriteCompare(uint8 compare)[1]

Description: Writes the compare values for the compare output.

Parameters: uint8 compare: Compare value

Return Value: None

Side Effects: Using the PSFB_WriteCompare() API when the PSFB is running will cause the comparison to use the new compare value immediately and that result will propagate to the output terminal on the next clock.

[1]**Note** The functions have the intended functionality only when the **ExternalModulation** parameter is set to **false**; otherwise, functions have no affect on the component.

DC and AC Electrical Characteristics

Specifications are valid for –40 °C ≤ T_A ≤ 85 °C and T_J ≤ 100 °C, except where noted. Specifications are valid for 1.71 V to 5.5 V, except where noted.

PhaseShiftFullBridge AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
f_{CLOCK}	Operating frequency		DC	-	67	MHz
	Period		-	-	255	-
	Compare Value		-	-	255	-

Component Changes

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

Version	Description of Changes	Reason for Changes / Impact
1.00	Initial Design	NA

© Cypress Semiconductor Corporation, 2013. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control, or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

PSoC® is a registered trademark, and PSoC Creator™ and Programmable System-on-Chip™ are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

Any Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.