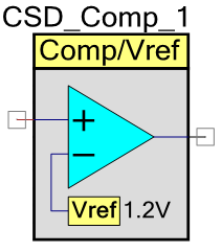
PSoC 4 CSD Comparator

1.0



Features

- Internal or external reference voltage
- Adjustable reference voltages
- Input voltage via AMUXBUSA or AMUXBUSB

General Description

The PSoC 4 CSD Comparator component uses the comparator in the CapSense® Capacitive Sigma-Delta Modulator (CapSense CSD) block to provide a comparator with adjustable reference voltage. The reference voltage can come from the internal bandgap reference or an off-chip external voltage reference. The reference voltage may be adjusted during run time. The comparator can take reference voltage input from 0.6 V to VDDA – 0.6 V.

When to Use a CSD Comparator

- Applications where upper and/or lower voltage limits must be detected
- Applications requiring adjustable compare voltages

Input/Output Connections

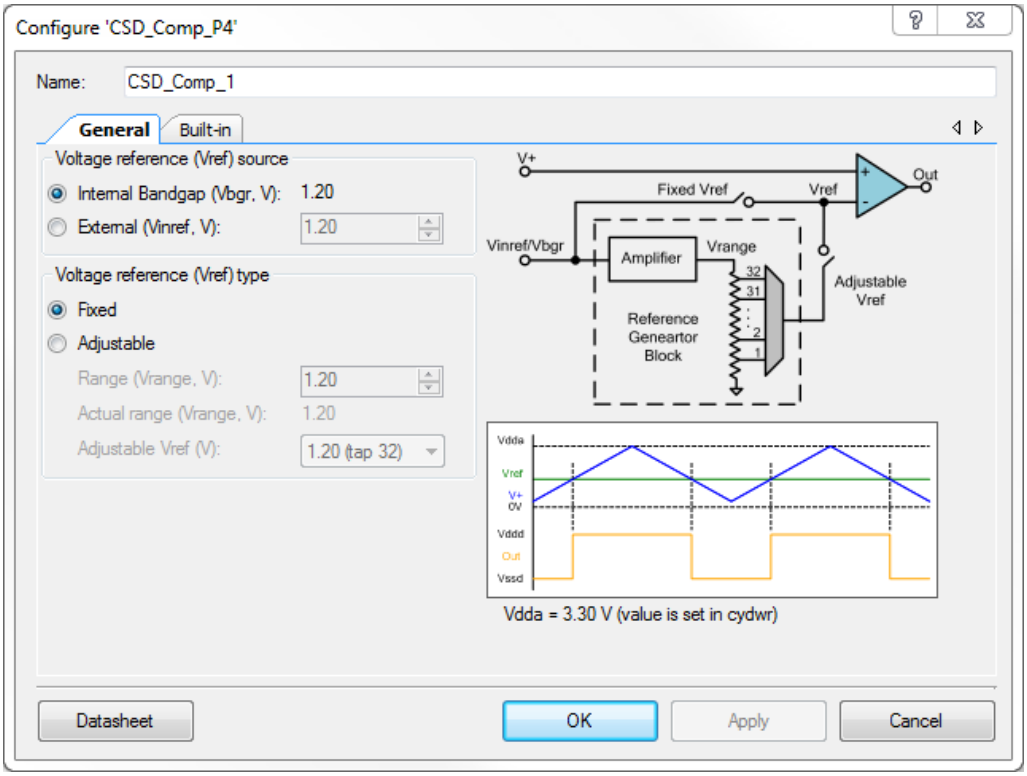
This section describes the various input and output connections for the CSD Comparator component. An asterisk (*) in the following list indicates that it may not be shown on the component symbol for the conditions listed in the description of that I/O.

Terminal Name	I/O Type	Description
Vplus	Analog Output	Reference Voltage Output.
Vminus *	Analog Output	Reference Voltage Output. This input is hidden by default and visible only when the “Reference Voltage Type” parameter is set to “External” in the GUI.
CmpOut	Digital Output	Comparator Output. This output goes high when the Vplus input voltage is greater than the Vminus.

Component Parameters

Drag a CSD Comparator component onto your design and double click it to open the Configure dialog. This dialog has the following tabs with different parameters.

General Tab



Parameter Name	Description
Voltage Reference Source	The source for reference voltage to the comparator. External reference can be from off chip external reference at a dedicated pin or through any other pin connected through AMUXBUSB.
Input Reference Voltage (VInRef)	Input reference voltage. Available if the source selected is external reference.
Voltage Reference Type	If it is “Fixed”, the VInRef will be the reference voltage to the comparator. Note that “Fixed” mode is a low power mode as the amplifier and resistor ladder network are bypassed. If it is “Adjustable”, the component generates multiple reference voltage options.
Reference Voltage Range	Voltage reference range that is expected from the amplifier. This option is not available if “Fixed” option is selected for Reference voltage option.

Parameter Name	Description
Actual Reference Voltage Range	Actual voltage reference range provided by the amplifier. The component calculates the closest value that the reference generator block can generate based on VDDA and VInRef parameters. This option is not available if “Fixed” option is selected for Reference voltage option.
Adjustable Vref tap	This is the tap value from the resistive divider network of the reference generator block that decides the reference voltage that goes to the inverting input of the comparator. This option is not available if “Fixed” option is selected for Reference Voltage.

Application Programming Interface

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

By default, PSoC Creator assigns the instance name **CSD_Comp_P4** 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 **CSD_Comp_P4**.

Modules

- General APIs
  General APIs are used for run-time configuration of the component during active power mode.
- Power Management APIs
  Power management APIs perform the necessary configurations to the components to prepare it for entering low power modes.

General APIs

General APIs are used for run-time configuration of the component during active power mode.

These include, initializing, starting, stopping, reading from registers and writing to registers.

Functions

- void CSD_Comp_P4_Init(void)
- void CSD_Comp_P4_Enable(void)
- void CSD_Comp_P4_Start(void)
- void CSD_Comp_P4_Stop(void)
- void CSD_Comp_P4_SetReferenceVoltageType(uint32 refVoltageType)
- void CSD_Comp_P4_SetVoltageTap(uint32 voltageTap)
- uint32 CSD_Comp_P4_GetCompare(void)

Function Documentation

void CSD_Comp_P4_Init (void)

Initializes or restores the component according to the customizer Configure dialog settings. It is not necessary to call CSD_Comp_P4_Init() because the CSD_Comp_P4_Start() API calls this function and is the preferred method to begin component operation.

Side Effects

This API includes blocking timeout required to run CSD IP block. Minimum timeout required for CSD IP block start is specified by CSD_Comp_P4_CSD_IP_STARTUP_TIMEOUT define.

void CSD_Comp_P4_Enable (void)

Activates the hardware and begins the component operation. It is not necessary to call CSD_Comp_P4_Enable() because the CSD_Comp_P4_Start() API calls this function, which is the preferred method to begin the component operation.

Side Effects

This API includes a blocking timeout required to run Reference Generator and CSD CMP blocks. The minimum timeout required for REFGEN block start is specified by CSD_Comp_P4_REFGEN_STARTUP_TIMEOUT defines.

void CSD_Comp_P4_Start (void)

Performs all of the required initialization for the component and enables power to the block. The first time the routine is executed, the power level is set. When called to restart the comparator following a CSD_Comp_P4_Stop() call, the current component parameter settings are retained.

Global Variables

CSD_Comp_P4_initVar - This variable is used to indicate the initial configuration of this component. The variable is initialized to zero and set to 1 the first time CSD_Comp_P4_Start() is called. This allows the component initialization without re-initialization in all subsequent calls to the CSD_Comp_P4_Start() routine.

void CSD_Comp_P4_Stop (void)

Turn off the Comparator block.

void CSD_Comp_P4_SetReferenceVoltageType (uint32 *refVoltageType*)

The API enables the REFGEN block in the bypass mode if the input voltage reference is fixed. Or else, it enables the REFGEN block in the high power mode.

Parameters:

refVoltageType	Comparator response time parameter. • CSD_Comp_P4_FIXED_REF_VOLTAGE - The fixed reference voltage type • CSD_Comp_P4_ADJUSTABLE_REF_VOLTAGE - The adjustable reference voltage type

Side Effects

This API stops the component for reconfiguration. All interrupts are disabled for the time this API is called.

void CSD_Comp_P4_SetVoltageTap (uint32 *voltageTap*)

Sets the reference voltage if the REFGEN block is enabled.

Parameters:

voltageTap	The tap number in the range from 1 to 32. The parameter can be calculated as following: Tap value = Round (Required reference voltage in mV / CSD_Comp_P4_VREF_STEP_MVOLTS) Note: The required reference voltage in mV should be in the range from 0.6V to Vrange (Actual reference voltage range from the component customizer). CSD_Comp_P4_VREF_STEP_MVOLTS value is provided in the component header file.

Side Effects

This API has no effect if the REFGEN block is in the bypass mode. This API stops the component for reconfiguration. All interrupts are disabled for the time this API is called.

uint32 CSD_Comp_P4_GetCompare (void)

This function returns a nonzero value when the voltage connected to the positive input is greater than Vref.

Returns:

uint32: Comparator output state. The nonzero value when the positive input voltage is greater than Vref; otherwise, the return value is zero.

Power Management APIs

Power management APIs perform the necessary configurations to the components to prepare it for entering low power modes.

These APIs must be used if the intent is to put the chip to sleep, then to continue the component operation when it comes back to active power mode.

This component does not stop the CSD IP block. One possible way to turn off the entire CSD block before sleep is to use a specific define (CSD_Comp_P4_CSD_ENABLE) for the IP block configuration register (CSD_Comp_P4_CSD_CONFIG_REG):

```
CSD_Comp_P4_CSD_CONFIG_REG &= ~CSD_Comp_P4_CSD_ENABLE;
```

Functions

- void CSD_Comp_P4_Sleep(void)
- void CSD_Comp_P4_Wakeup(void)

Function Documentation

void CSD_Comp_P4_Sleep (void)

The CSD_Comp_P4_Sleep() function checks to see if the component is enabled and saves that state. Then it calls the _Stop() function. It is recommended to call the CSD_Comp_P4_Sleep() function before calling the CyPmSleep() or the CyPmDeepSleep() function.

void CSD_Comp_P4_Wakeup (void)

If the component was enabled before calling the CSD_Comp_P4_Sleep() function, the CSD_Comp_P4_Wakeup() function will re-enable the component.

Global Variables

Global variables used in the component.

The following global variables are used in the component.

Variables

- uint8 **CSD_Comp_P4_initVar**

Code Examples and Application Notes

Code Examples

PSoC Creator provides access to code examples in the Code Example dialog. For component-specific examples, open the dialog from the Component Catalog or an instance of the component in a schematic. For general examples, open the dialog from the Start Page or **File** menu. As needed, use the **Filter Options** in the dialog to narrow the list of projects available to select.

Refer to the "Code Example" topic in the PSoC Creator Help for more information.

There are also numerous code examples that include schematics and example code available online at the Cypress Code Examples web page. Examples that use this component include:

- CE210197- Adjustable Reference Comparator

Application Notes

Cypress provides a number of application notes describing how PSoC can be integrated into your design. You can access the Cypress Application Notes search web page at www.cypress.com/appnotes.

API Memory Usage

Shows the Flash, SRAM and stack usage of the component.

The component memory usage varies significantly depending on the compiler, device, number of APIs used and component configuration. The following table provides the memory usage for all APIs available in the given component configuration.

The measurements have been done with an associated compiler configured in Release mode with optimization set for Size. For a specific design, the map file generated by the compiler can be analyzed to determine the memory usage.

PSoC 4 (GCC)

Configuration	PSoC 4000S/PSoC 4100S		PSoC Analog Coprocessor	
	Flash Bytes	SRAM Bytes	Flash Bytes	SRAM Bytes
Default	508	2	500	2

Functional Description

The adjustable reference option in the component provides flexibility to compare input voltage with different levels of reference voltages. The component can be used within the following use cases:

Comparator with Fixed Reference

The CSD Comparator compares the input voltage applied to the non-inverting terminal with a fixed reference voltage applied to the inverting terminal. The fixed reference can come from

1. On chip reference Vbgr of 1.2 V
2. Off chip external reference Ext_Vref
3. External Vref from any pin using AMUXBUSB

Comparator with Adjustable Reference

The CSD Comparator compares the input voltage applied to the non-inverting terminal with a variable reference voltage generated by the reference generator “REFGEN” block. The input reference voltage to the REFGEN can come from either one of the sources mentioned above. The adjustable reference voltage can vary from 0.6V to VDDA-0.6V.

Comparator with single input voltage

The CSD Comparator can take input from a single pin which gets connected to the comparator internally through AMUXBUSA or B.

Comparator with multiple input voltage

The CSD Comparator can take input from multiple pins. The user needs to place AMUX component to connect multiple pins to the input of the comparator.

Note CSD_Comp_P4_SetReferenceVoltageType() and CSD_Comp_P4_SetVoltageTap() APIs stop the component before the configuration change and then enable it again to prevent glitches at the component output.

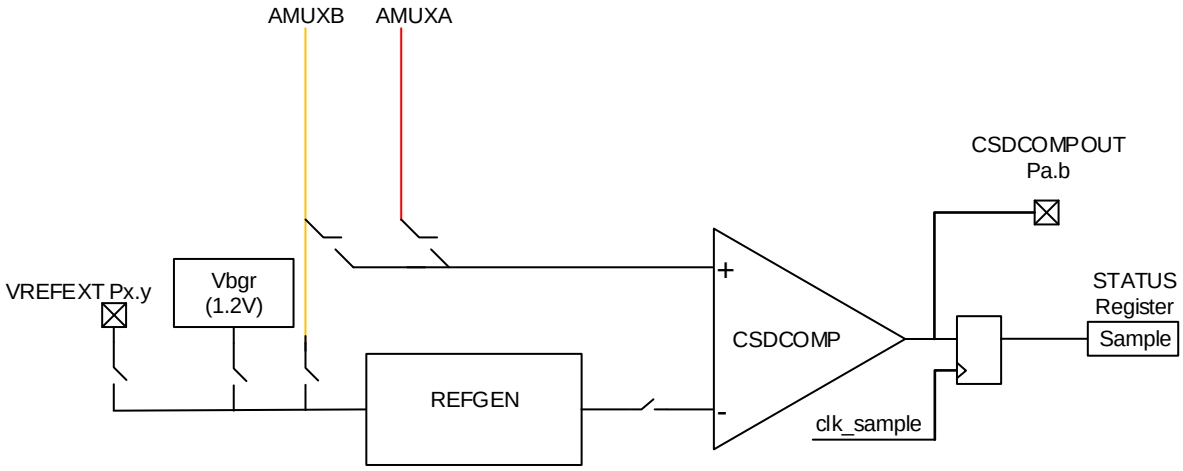
Definitions

- *CSDV2* **-** Capacitive Sigma Delta version 2
- *REFGEN* **-** Reference Generator

Block Diagram and Configuration

The CSD Comparator uses the CSD_CONFIG register to enable or disable the comparator. The CSD_STATUS register could be used for polling via the GetCompare() API.

The following is a simplified diagram of the CSD Comparator hardware:



Where, Px.y is the dedicated pin for external reference voltage. Pa.b is the dedicated pin on which the comparator output is routed.

Clock Selection

The CSD Comparator component uses internal cy_clock component in its implementation. It is connected to clock input of CSD Comparator block and required to feed internal CSD logic.

Placement

The CSD Comparator component is placed in the dedicated CSDv2 IP in the silicon. Components based on the CSDv2 block are mutually exclusive and can't be used together in the same design.

MISRA Compliance

This section describes the MISRA-C:2004 compliance and deviations for the component. There are two types of deviations defined:

- project deviations – deviations that are applicable for all PSoC Creator components
- specific deviations – deviations that are applicable only for this component

This section provides information on component-specific deviations. Project deviations are described in the MISRA Compliance section of the *System Reference Guide* along with information on the MISRA compliance verification environment.

The CSD Comparator component does not have any specific deviations.

This component has the following embedded components: Clock. Refer to the corresponding component datasheets for information on their MISRA compliance and specific deviations.

Registers

For fixed-function blocks, refer to the chip *Technical Reference Manual (TRM)* for more information about the registers.

Component Debug Window

PSoC Creator allows you to view debug information about components in your design. Each component window lists the memory and registers for the instance. For detailed hardware registers descriptions, refer to the appropriate device technical reference manual.

To open the Component Debug window:

1. Make sure the debugger is running or in break mode.
2. Choose **Windows > Components…** from the **Debug** menu.
3. In the Component Window Selector dialog, select the component instances to view and click **OK**.

The selected Component Debug window(s) will open within the debugger framework. Refer to the "Component Debug Window" topic in the PSoC Creator Help for more information.

Resources

The CSD Comparator component uses the dedicated CSDv2 IP in the silicon.

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.

Note Final characterization data for the PSoC Analog Coprocessor device is not available at this time. Once the data is available, the component datasheet will be updated on the Cypress web site.

DC Specifications

Parameter	Description	Min	Typ	Max	Units	Conditions
V_{REF}	Voltage reference for CSD and Comparator	0.6	1.2	V_{DDA} - 0.6	V	V_{DDA} - 0.06 or 4.4, whichever is lower
V_{REF_EXT}	External Voltage reference for CSD and Comparator	0.6	–	V_{DDA} - 0.6	V	V_{DDA} - 0.06 or 4.4, whichever is lower
V_{CSD}	Voltage range of operation	1.71	–	5.5	V	1.8 V ±5% or 1.8 V to 5.5 V

Component Changes

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

Version	Description of Changes	Reason for Changes / Impact
1.0.c	Updated the datasheet.	Added final characterization data for PSoC 4100S device.
1.0.b	Updated the datasheet.	Added final characterization data for PSoC 4000S device.
1.0.a	Updated the datasheet.	Removed Errata section. Updated screen capture and parameter descriptions. Final characterization data for PSoC 4000S, PSoC 4100S and PSoC Analog Coprocessor devices is not available at this time. Once the data is available, the component datasheet will be updated on the Cypress web site.
1.0	Initial component version.	

© Cypress Semiconductor Corporation, 2016-2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC (“Cypress”). This document, including any software or firmware included or referenced in this document (“Software”), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress’s patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage (“Unintended Uses”). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.