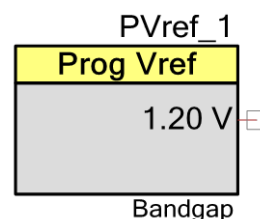


PSoC 4 Programmable Voltage Reference (PVref_P4)

1.0

Features

- Reference is based on either Bandgap or Vdda
- Each instantiation may be a different value
- Adjustable by 1/16 steps between Vss and Bandgap or Vdda
- Up to four individual references
- References may be changed during runtime



General Description

The PSoC 4 Programmable Voltage Reference (PVref_P4) component provides an adjustable stable reference voltage for the analog resources in your design. Each component provides a voltage reference that is based on either Vdda or the Bandgap voltage in 1/16 increments. This reference may be connected to any high impedance input such as the input to an opamp or comparator and cannot be routed to a GPIO unless it is buffered with an opamp.

When to Use a PVref_P4

Use the Programmable Reference any time a stable known voltage is required for comparison, such as with comparators, VDACS, and ADCs.

Input/Output Connections

This section describes the various input and output connections for the PVref_P4 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.

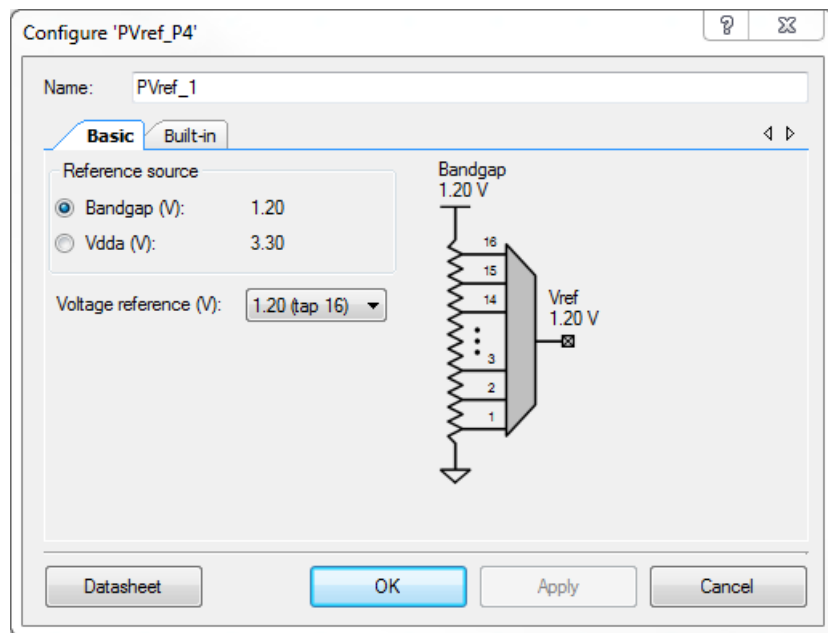
Terminal Name	I/O Type	Description
Vout	Analog Output	Reference Voltage Output.

PRELIMINARY

Component Parameters

Drag a PVref_P4 component onto your design and double click it to open the Configure dialog. This dialog has the following tabs with different parameters.

Basic Tab



Parameter Name	Description
Reference Source	This option selects the reference source to be based on either the internal bandgap voltage or the Vdda supply. The default is bandgap voltage.
Output Voltage	This parameter is used to select the desired output voltage from the voltage divider. Default is the full reference source ($16/16 \cdot V_{ref}$).

PRELIMINARY



Application Programming Interface

API functions allow configuration of the component using the CPU.

By default, PSoC Creator assigns the instance name **PVref_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 **PVref_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 [PVref_P4_Init](#)(void)
- void [PVref_P4_Enable](#)(void)
- void [PVref_P4_Start](#)(void)
- void [PVref_P4_Stop](#)(void)
- void [PVref_P4_SetTap](#)(uint32 dividerTap)
- void [PVref_P4_SetVRefSrc](#)(uint32 reference)

Function Documentation

void PVref_P4_Init (void)

Initialize/Restore the default Voltage Reference configuration as defined in the customizer. If the reference is set to Vdda, it will enable all Vdda-based references.

Side Effects

If the reference is Vdda-based, the function will enable all Vdda-based references because they share the same resistor tree.

void PVref_P4_Enable (void)

Enables the Voltage Reference.

void PVref_P4_Start (void)

Initializes all parameters required to setup the component as defined in the customizer. If the reference is set to Vdda, it will enable all Vdda based references.



PRELIMINARY

Global Variables

PVref_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 [PVref_P4_Start\(\)](#) is called. This allows the component initialization without re-initialization in all subsequent calls to the [PVref_P4_Start\(\)](#) routine.

Side Effects

If the reference is Vdda-based, the function will enable all Vdda-based references because they share the same resistor tree.

void PVref_P4_Stop (void)

Disables the Voltage Reference.

void PVref_P4_SetTap (uint32 dividerTap)

Selects which of the equally spaced resistor divider taps will be used for the reference. The output will range in 1/16th steps between Vref/16 to Vref.

Parameters:

<i>dividerTap</i>	<p>One of the 16-tap voltage dividers to select a voltage between 1/16 of the reference source to the full reference source in steps of 1/16.</p> <ul style="list-style-type: none"> • 1 : 1/16 Vref • ... : ... • 16 : Vref
-------------------	---

void PVref_P4_SetVRefSrc (uint32 reference)

Selects whether the reference at the top of the resistor divider is based on the bandgap or Vdda.

Parameters:

<i>reference</i>	<p>The reference source.</p> <ul style="list-style-type: none"> • PVref_P4_BANDGAP_REFERENCE : Bandgap reference • PVref_P4_VDDA_REFERENCE : Vdda reference
------------------	---

Side Effects

If the reference is based on Vdda, it will affect all Vdda-based references because they share the same resistor tree.

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.

Functions

- void [PVref_P4_Sleep](#)(void)
- void [PVref_P4_Wakeup](#)(void)

PRELIMINARY



Function Documentation

void PVref_P4_Sleep (void)

Prepares the component to enter Sleep mode. If the reference is based on Vdda, the function will open the switch at the bottom of the resistor tree to eliminate the current flow through the resistor. If the reference is based on the bandgap, this function will do nothing.

Side Effects

If the reference is based on Vdda, it will affect all Vdda-based references because they share the same resistor tree.

void PVref_P4_Wakeup (void)

Restores the component after exiting Sleep mode. By calling this function for any Vdda-based reference, the function will enable all Vdda-based references because they all are based on the same resistor divider. For bandgap-based references, nothing will happen.

Side Effects

If the reference is based on Vdda, it will affect all Vdda-based references because they share the same resistor tree.

Global Variables

Global variables used in the component.

The following global variables are used in the component.

Variables

- uint8 PVref_P4_initVar

PVref_P4_backup_struct Struct Reference

Data Fields

- uint8 enableState

Code Examples and Application Notes

This section lists the projects that demonstrate the use of the component.

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:

- CE203972 - Comparator with Programmable Voltage Reference

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 Analog Coprocessor	
	Flash Bytes	SRAM Bytes
Default	232	2

PRELIMINARY



Functional Description

The PVref_P4 component provides an adjustable analog voltage reference in your design. You can instantiate several PVref_P4 components on the schematic (particular number depends on the selected device resources and can be obtained from the device datasheet) and set the reference source to be either Vdda or the bandgap voltage. You may also select one tap of the 16-tap voltage divider to select a voltage between 1/16 of the reference source to the full reference source in steps of 1/16. The PVref_P4 component output may be connected to any high impedance input such as the input to an opamp or comparator. To route component output to a GPIO - it should be buffered with an opamp.

Note There are general predefined settings for the Programmable Reference Block, such as "Reference Source", "Voltage Reference Value", "Bandgap Gain", "Enable Vdda reference during Deep Sleep", which are located under the System Editor of the Design-Wide Resources file. For more details, refer to the PSoC Creator Help.

Note Changes to these settings will affect all PVref_P4 component instances. The general settings in the System Editor can be tuned by System APIs as well. For more details refer, to System Reference Guide.

Note The actual value of the Vdda-based reference is dependable on real value of the Vdda voltage. The Vdda voltage is changeable and should be accurately treated to get precise reference.

Low Power Mode Behavior

The bandgap reference will be automatically disabled in deep sleep. In case of Vdda reference source to eliminate the current flow through the resistor it can be switched off by calling PVref_P4_Sleep() API. There is an option to leave Vdda reference enabled in deep sleep. To do this "Vdda reference active during deep sleep" option should be enabled in System tab at the Design Wide Resources window and PVref_P4_Sleep() API shouldn't be called prior to enter device into deep sleep mode.

Block Diagram and Configuration

A simplified diagram of the PVref_P4 hardware is shown in the component customizer. The reference voltages are derived from a resistor string. There are 2 separate identical resistor strings. The resistor string is made up of 16 resistor legs each with a poly taps. The top of the first resistor string is connected to bandgap reference and the top of the second resistor is connected to Vdda reference. The bottom of the first resistor string is connected to Vssa and the bottom of the second resistor string is connected through a switch Vdda_enable to Vssa. Each resistor string has four 16:1 muxes. This 16:1 mux selects one of 16 voltages from the resistor string. The output reference voltages are not buffered and must be connected to only high impedance nets.



PRELIMINARY

Industry Standards

Lists the industry standard compliance of the PVref_P4 component.

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 PVref_P4 component does not have any specific deviations.

Registers

Describes the UDB registers used in the PVref_P4 component.

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

Resources

The PVref_P4 component uses the dedicated PRB (Programmable Reference Block) IP in the silicon.

DC and AC Electrical Characteristics

Specifications are valid for $-40^{\circ}\text{C} = T_A = 85^{\circ}\text{C}$ and $T_J = 100^{\circ}\text{C}$, except where noted.

Specifications are valid for 1.71 V to 5.5 V, except where noted.

Note 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.

DC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
	Total resistor string resistance		-	86.45	-	kOhms

PRELIMINARY



AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
Resistor divider ratio error	Deviation from 1/16 taps.		-	TBD	-	%
Bandgap resistor divider current	Bandgap divided by the resistance of the entire resistor divider string.		-	TBD	-	uA
Vdda resistor divider current	Vdda divided by the resistance of the entire resistor divider string.		-	TBD	-	uA

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.a	Edited datasheet.	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. 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.



PRELIMINARY