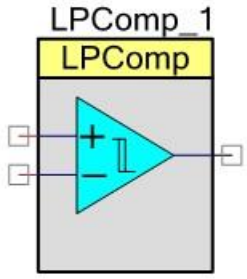
PSoC 4 Low Power Comparator (LPComp)

2.20



Features

- Low input offset
- User controlled offset calibration
- Multiple speed modes
- Low-power mode
- Wake from low power modes
- Multiple interrupt and output modes

General Description

The Low Power Comparator (LPComp) Component provides access to the low power comparators implemented using the fixed function LP comparator block that is present in PSoC 4.

When to Use a LPComp

The main purpose of these comparators is to offer fast detection of a voltage change in normal operating modes and ultra-low power operation in hibernate mode. The two connection options are described as follows:

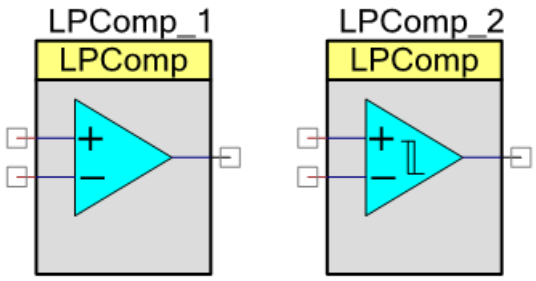
- Compare two voltages on external pins.
- Compare a voltage from an external pin against an internally generated signal.

The comparator output value can be inspected by the CPU. The comparator interrupt output signal is ORed together with interrupt output signal from the other low power comparator on the device as combined interrupt source. Interrupt is not cleared automatically. It is user responsibility to do that. This combined signal is available as the output of the global signal reference Component and can be used as an interrupt/wakeup source from deep-sleep or hibernate. Refer to the Operation in Low Power Mode section for more details.

For all except PSoC 4100 / PSoC 4200 families, the comparator output is available directly, or after synchronous edge detection as level or pulse. It can be used to trigger an interrupt, routed to digital logic or sent to a pin. Refer to the Functional Description section for details.

Input/Output Connections

This section describes the various input and output connections for the LPComp. The symbol for this Component is annotated to denote the selection of hysteresis.



Positive Input – Analog Input

This input is usually connected to the voltage that is being compared. This input can be routed from a GPIO or from an internal source. When connected to an internal source the GPIO that is dedicated to this input will be consumed and not available for other uses.

Negative Input – Analog Input

This input is usually connected to the reference voltage. This input can be routed from a GPIO or from an internal source. When connected to an internal source the GPIO that is dedicated to this input will be consumed and not available for other uses.

Comparator Out – Output

For PSoC 4100 / PSoC 4200 families it is the edge detected comparison output. For these devices the direct result of the comparator is not available as a hardware signal. This output is normally connected to an interrupt.

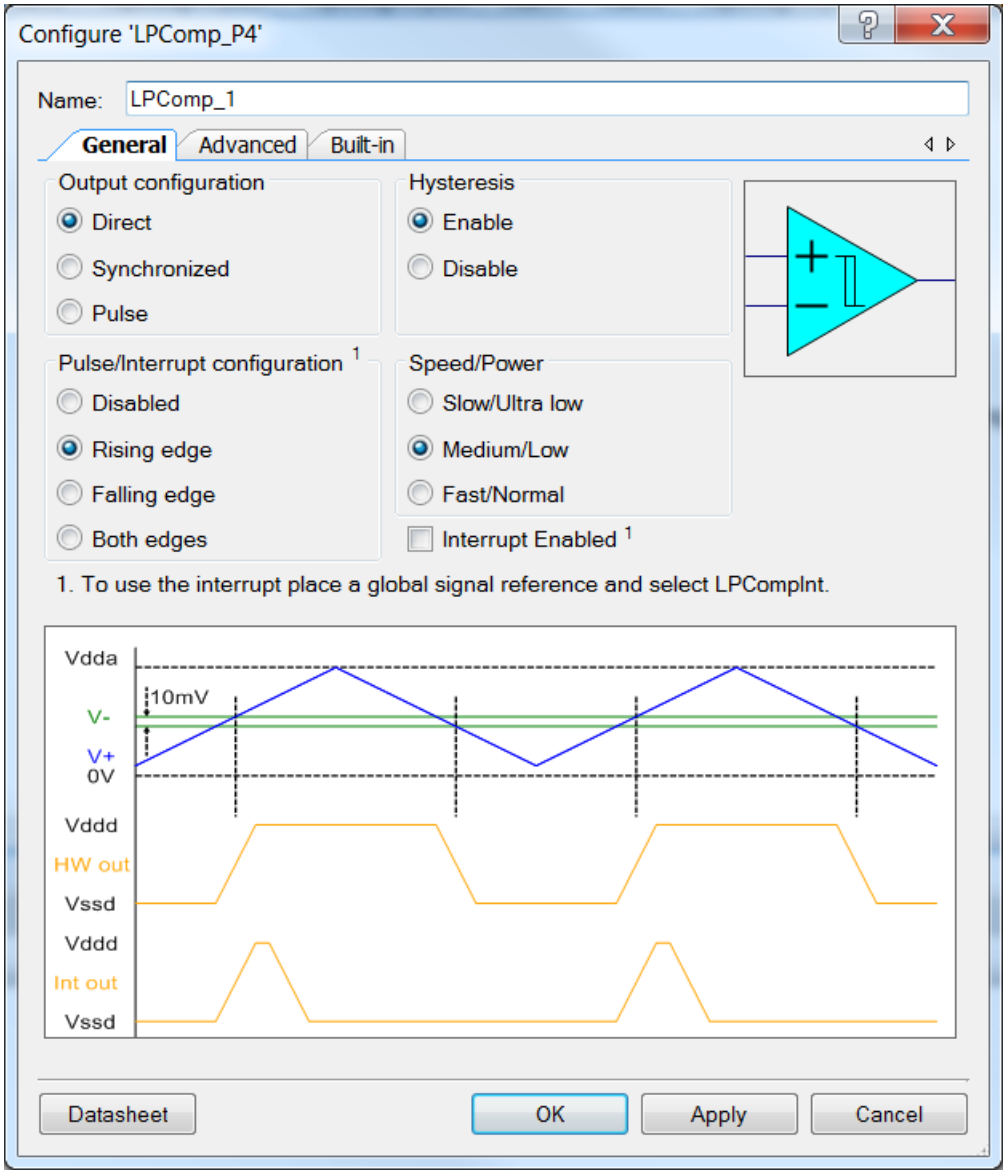
For all other PSoC 4 devices, the Component output is configurable (refer to the Output Configuration section): either direct comparator out or after synchronous edge detection as level or pulse. It can be used to trigger an interrupt, routed to digital logic or sent to a pin. Also, this output can be used as an interrupt source for those devices that support the Digital Signal Interconnect (DSI). For non-DSI devices, the Global Signal Reference (GSR) is the only way to use interrupts. Refer to the Functional Description section for details.

Note This output cannot be used to awake a part from the Low Power modes. The GSR interrupt signal must be used instead.

Component Parameters

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

General Tab



The LPComp provides the following parameters.

Output Configuration

This parameter defines mode of the LPComp output: Direct, Synchronized, or Pulse. For PSoC 4100 / PSoC 4200 families, a pulse is generated when the selected edge occurs on the internal comparison signal. Refer to the Functional Description section for details.

Hysteresis

This parameter allows you to add approximately 10 mV of hysteresis to the LPComp. This helps to ensure that slowly moving voltages or slightly noisy voltages will not cause the output of the LPComp to oscillate when the two input voltages are nearly equal.

Pulse/Interrupt Configuration

This parameter defines the event that will cause a pulse to be generated on the interrupt terminal or direct comparator output. The parameter allows you to select interrupt/output mode: Disabled, Rising edge, Falling edge, or Both edges.

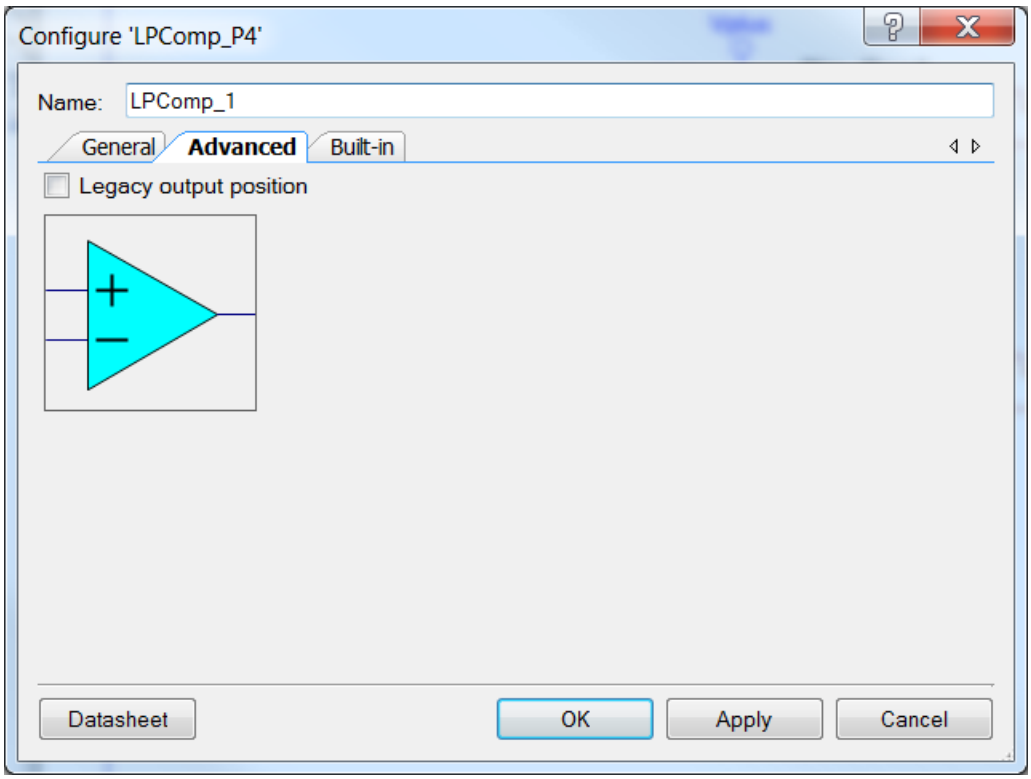
Speed/Power

This parameter provides a way to optimize speed verses power consumption. The Speed/Power parameter allows you to select the speed/power level: Slow/Ultra low, Medium/Low, Fast/Normal.

Interrupt Enabled

This parameter allows you to configure the Interrupt Mask (except PSoC 4100 / PSoC 4200 families).

Advanced Tab

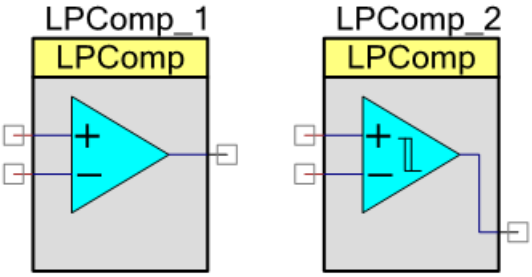


Legacy output position

This parameter allows you to configure the comparator output position for legacy support.

By default, the Legacy output position option is selected when a design is being migrated from an existing design for PSoC 4100 / PSoC 4200 families. If the LPComp Component is placed in a design for the first time, the Legacy output position option will not be selected by default.

The symbol view will change for this option as follows:



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 "LPComp_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 "LPComp"

Functions

Function	Description
LPComp_Start()	Performs all of the required initialization for the Component and enables power to the block.
LPComp_Stop()	Turns off the LP Comparator block.
LPComp_Init()	Initializes or restores the Component according to the customizer settings.
LPComp_Enable()	Activates the hardware and begins Component operation.
LPComp_GetCompare()	This function returns a nonzero value when the voltage connected to the positive input is greater than the negative input voltage.
LPComp_SetSpeed()	Sets the power and speed to one of three settings.
LPComp_SetHysteresis()	Enables or disables the hysteresis setting.
LPComp_ZeroCal()	Performs custom calibration of the input offset to minimize error for a specific set of conditions.
LPComp_LoadTrim()	Writes a value into the comparator offset trim register.
LPComp_SetOutputMode()	Set comparator output mode (Direct, Synchronized or Pulse)
LPComp_SetInterruptMode()	Sets the interrupt edge detect mode.
LPComp_GetInterruptSource()	Gets the interrupt requests.
LPComp_ClearInterrupt()	Clears the interrupt request.
LPComp_SetInterrupt()	Sets a software interrupt request.
LPComp_SetInterruptMask()	Set interrupt mask. (Not applicable for PSoC 4100 / PSoC 4200.)
LPComp_GetInterruptMask()	Returns interrupt mask. (Not applicable for PSoC 4100 / PSoC 4200.)
LPComp_GetInterruptSourceMasked()	Returns interrupt request register masked by interrupt mask. (Not applicable for PSoC 4100 / PSoC 4200.)

void LPComp_Start(void)

Description: Performs all of the required initialization for the Component and enables power to the block. The first time the routine is executed, the Component is initialized to the configuration from the customizer. When called to restart the comparator following a LPComp_Stop() call, the current Component parameter settings are retained.

void LPComp_Stop(void)

Description: Turns off the LP Comparator.

Side Effects: The function doesn't change Component power settings.

void LPComp_Init(void)

Description: Initializes or restores the Component according to the customizer settings. It is not necessary to call LPComp_Init() because the LPComp_Start() API calls this function and is the preferred method to begin Component operation.

void LPComp_Enable(void)

Description: Activates the hardware and begins Component operation. It is not necessary to call LPComp_Enable() because the LPComp_Start() API calls this function, which is the preferred method to begin Component operation.

uint32 LPComp_GetCompare(void)

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

This function reads the direct (unflopped) comparator output, which can also be metastable (since it may result in incorrect data).

Return Value: Comparator output state. This value is not impacted by the interrupt edge detect setting.

void LPComp_SetSpeed(uint32 speed)

Description: Sets the power and speed to one of three settings.

Parameters: speed: Enumerated speed mode value

Name	Description
LPComp_LOW_SPEED	Slow speed / Ultra low power
LPComp_MED_SPEED	Medium speed / Low power
LPComp_HIGH_SPEED	High speed / Normal power

void LPComp_SetHysteresis(uint32 hysteresis)

Description: Enables or disables the hysteresis setting.

Parameters: hysteresis: Hysteresis enable. See table below.

Name	Description
LPComp_HYST_ENABLE	Enable hysteresis
LPComp_HYST_DISABLE	Disable hysteresis

uint32 LPComp_ZeroCal(void)

Description: Performs custom calibration of the input offset to minimize error for a specific set of conditions: comparator reference voltage, supply voltage, and operating temperature. A reference voltage in the range at which the comparator will be used must be applied to the negative input. The positive input will be disconnected from the external source and shorted internally to the negative input to perform the offset calibration.

Return Value: The value from the comparator trim register after the offset calibration is complete. This value has the same format as the input parameter for the LPComp_LoadTrim() API routine.

Side Effects: During this function the positive input terminal is disconnected. This may impact other Components that are also using this signal.

void LPComp_LoadTrim(uint32 trimVal)

Description: This function writes a value into the comparator offset trim register.

Parameters: trimVal: Value to be stored in the comparator offset trim register. This value has the same format as the parameter returned by the LPComp_ZeroCal() API routine.

void LPComp_SetOutputMode (uint32 mode)

Description: Set comparator output mode.

Parameters: mode: Comparator output mode value. See table below.

Name	Description
LPComp_OUT_DIRECT	Direct output
LPComp_OUT_SYNC	Synchronized output
LPComp_OUT_PULSE	Pulse output

void LPComp_SetInterruptMode(uint32 mode)

Description: Sets the interrupt edge detect mode. This also controls the value provided on the output.

Parameters: mode: Enumerated edge detect mode value. See table below.

Name	Description
LPComp_INTR_DISABLE	Disable
LPComp_INTR_RISING	Rising edge detect
LPComp_INTR_FALLING	Falling edge detect
LPComp_INTR_BOTH	Detect both edges

uint32 LPComp_GetInterruptSource(void)

Description: Gets the interrupt requests. This function is used with the combined interrupt signal from the global signal reference. This function from either Component instance can be used to determine the interrupt source for both the interrupts combined.

Return Value: Interrupt source. Each Component instance has a mask value: LPComp_INTR.

void LPComp_ClearInterrupt(uint32 interruptMask)

Description: Clears the interrupt request. This function is used with the combined interrupt signal from the global signal reference. This function from either Component instance can be used to clear either or both interrupts.

Parameters: interruptMask: Mask of interrupts to clear. Each Component instance has a mask value: LPComp_INTR.

void LPComp_SetInterrupt(uint32 interruptMask)

Description: Sets a software interrupt request. This function is used with the combined interrupt signal from the global signal reference. This function from either Component instance can be used to trigger either or both software interrupts.

Parameters: interruptMask: Mask of interrupts to set. Each Component instance has a mask value: LPComp_INTR.

void LPComp_SetInterruptMask (uint32 interruptMask)

Description: Configures which bits of interrupt request register will trigger an interrupt event. This API is not applicable for PSoC 4100 / PSoC 4200 devices.

Parameters: interruptMask: bit-mask of interrupt sources to be enabled. Each Component instance has a mask value: LPComp_INTR_MASK.

uint32 LPComp_GetInterruptMask (void)

Description: Returns interrupt mask. This API is not applicable for PSoC 4100 / PSoC 4200 devices.

Return Value: Mask of enabled interrupt source. Each Component instance has a mask value: LPComp_INTR_MASK.

uint32 LPComp_GetInterruptSourceMasked (void)

Description: Returns interrupt request register masked by interrupt mask.

Returns the result of bitwise AND operation between corresponding interrupt request and mask bits. This API is not applicable for PSoC 4100 / PSoC 4200 devices.

Return Value: Mask of enabled interrupt source. Each Component instance has a mask value: LPComp_INTR_MASKED.

Global Variables

Function	Description
LPComp_initVar	Indicates whether the Low Power Comparator has been initialized. The variable is initialized to 0 and set to 1 the first time LPComp_Start() is called. This allows the Component to restart without reinitialization after the first call to the LPComp_Start() routine. If reinitialization of the Component is required, call LPComp_Init() before calling LPComp_Start(). Alternatively, the Comparator can be reinitialized by calling the LPComp_Init() and LPComp_Enable() functions

Sample Firmware Source Code

PSoC Creator provides numerous example projects that include schematics and example code in the Find Example Project 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 "Find Example Project" topic in the PSoC Creator Help for more information.

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 LPComp Component has the following specific deviations:

MISRA-C: 2004 Rule	Rule Class (Required/Advisory)	Rule Description	Description of Deviation(s)
19.7	A	A function is used in preference to a function-like macro.	Deviated since function-like macros are used to allow more efficient code.

API Memory Usage

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 4100/PSoC 4200		Other PSoC 4 Devices	
	Flash Bytes	SRAM Bytes	Flash Bytes	SRAM Bytes
Default	552	8	700	8

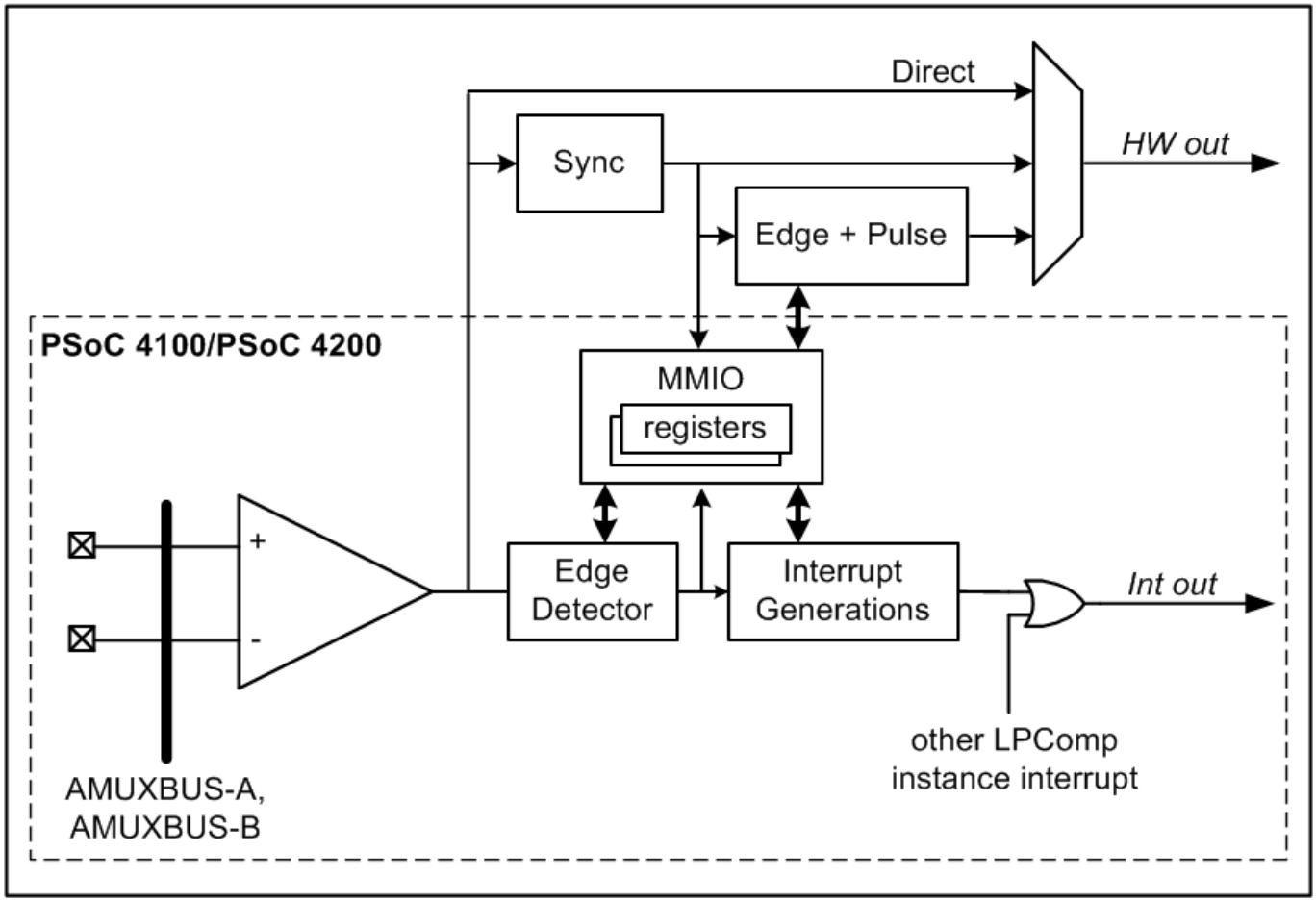
Functional Description

The Low Power Comparator is intended to be operational in all power modes, including Hibernate/Deep Sleep. The main purpose of this Component is to offer fast detection capability in normal operating modes and ultra-low power operation in hibernate mode.

LPComp can be offset calibrated. However, even without trim, the offset is around 10 mV, and thus it is not recommended to trim it.

Block Diagram and Configuration

The following is a high-level block diagram.



In PSoC 4 devices (except PSoC 4100 / PSoC 4200) each comparator has one HW output. It comes from three sources:

- Direct comparator output
- Synchronized to SYSCLK using two flip-flops
- Edge detected pulse which have period 2*SYSCLK.

Individual comparator interrupt outputs are ORed together as single asynchronous interrupt source before sent out and used to wakeup system in low power mode. For all devices except PSoC 4100 /4200 families, the individual comparator interrupt is masked by INTR_MASK. The masked result is captured in INTR_MASKED register. Writing 1 to INTR register bit will clear the interrupt. Refer to the appropriate device *Technical Reference Manual (TRM)* for a detailed description of the registers.

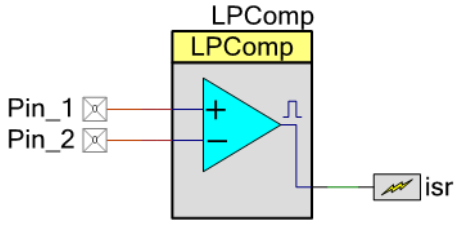
Interrupt Service Routine

The LPComp supports interrupts on the various events, depends on *Pulse/Interrupt configuration* settings: Rising edge, Falling edge, or both edges. Interrupt signal goes high when any of the enabled interrupt configurations are true.

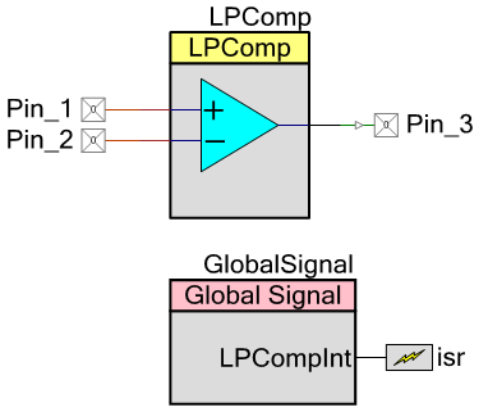
Note Interrupt is not cleared automatically. It is user responsibility to do that. Interrupt is cleared by writing a '1' in corresponding interrupt register bit position. The preferred way to clear interrupt sources is usage *LPComp_ClearInterrupt(LPComp_INTR)* API.

There are two ways for access to the LPComp interrupt depending on the *Legacy output position* option.

Legacy output position option is set.



Legacy output position option is NOT set.



The following code is suggested:

```
CY_ISR(LPComp_Interrupt)
{
    /* Interrupt does not clear automatically.
    * It is user responsibility to do that.
    */
    LPComp_ClearInterrupt(LPComp_INTR);

    /*
    * Add user interrupt code to manage interrupt.
    */
}
```

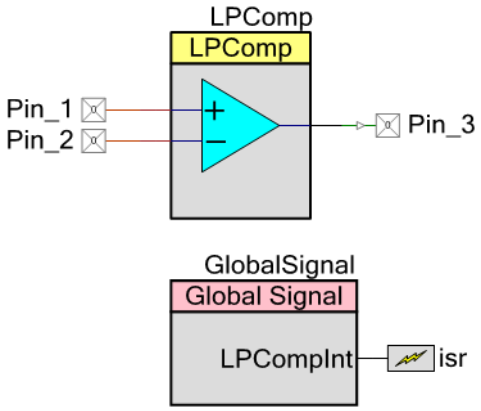
```
void main()
{
    isr_StartEx(LPComp_Interrupt);
    LPComp_Start();
    CyGlobalIntEnable; /* Enable global interrupts. */

    for(;;)
    {
        /* Place your application code here. */
    }
}
```

Operation in Low Power Mode

Low Power Comparator operates in Deep Sleep/Hibernate Mode **only if Speed/Power option is set to Slow/Ultra low**. This Component can be used along with the Global Signal Component as a wakeup source from Deep Sleep/Hibernate mode as described below.

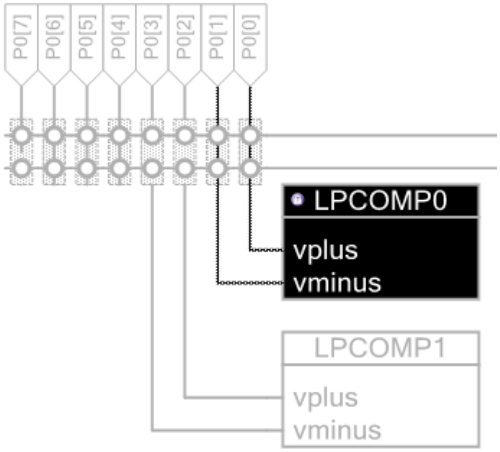
Create schematic next schematic.



The option “Combined low power comparator interrupt (LPCompInt)” should be selected as a source for an interrupt in the Global Signal Reference Component. This Component triggers the output each time any of the enabled low power comparators generates an interrupt.

Note Make sure that both input signals are valid during operation in Low Power Mode (that is, that the the signal sources (especially Components) work in Low Power Mode). Otherwise, if any of the input signals is not determined, then the LPComp Component’s output would also be incorrect.

The LPComp Component input pins must be assigned to dedicated inputs only (P0[0], P0[1] for LPCOMP0, and P0[2], P0[3] for LPCOMP1 in case of the PSoC 4100/PSoC 4200). Refer to the device datasheet for the part being used for the specific physical pin connections.



The wakeup event is when the voltage connected to the positive input (Pin_1) is greater than the negative input voltage (Pin_2).

The following code is suggested:

```
CY_ISR(LPComp_Interrupt)
{
    /* Interrupt does not clear automatically.
    * It is user responsibility to do that.
    */
    LPComp_ClearInterrupt(LPComp_INTR);

    /*
    * Add user interrupt code to manage interrupt.
    */
}

void main()
{
    isr_StartEx(LPComp_Interrupt);
    LPComp_Start();
    CyGlobalIntEnable;     /* Enable global interrupts. */
    LPComp_SetInterruptMode(LPComp_INTR_RISING);
    LPComp_SetInterruptMask(LPComp_INTR_MASK);

    for(;;)
    {
        /* Place your application code here. */

        CySysPmDeepSleep(); /* Enter Deep Sleep mode*/

        /* Place your application code here. */
    }
}
```

Placement

Each comparator is directly connected to specific GPIOs for its inputs. The output connection is routed to the digital fabric. Refer to the device datasheet for the part being used for the specific physical pin connections.

Registers

See the chip Technical *Reference Manual (TRM)* for more information about 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. For detailed UDB registers descriptions used in the Component, refer to the Registers section of this datasheet.

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

This Component uses one of the LP Comparators from the pair of comparators in the LP Comparator hardware block.

DC and AC Electrical Characteristics

Specifications are valid for 2.2 V to 5.5 V @ –40 °C ≤ T_A < 0 °C, except where noted. Specifications are valid for 1.8 V to 5.5 V @ 0 °C ≤ T_A ≤ 85 °C and TJ ≤ 100 °C, 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	Conditions	Min	Typ	Max	Units
$V_{OFFSET1}$	Input offset voltage, factory trim	For PSoC 4000S / PSoC 4100S families	–	–	±10	mV
$V_{OFFSET2}$	Input offset voltage, custom trim		–	–	±4	mV
$V_{OFFSET2}$	Input offset voltage, custom trim	For PSoC 4 BLE family	–	–	±6	mV
$V_{OFFSET2}$	Input offset voltage. Common Mode Voltage Range from 0 to VDD-1	For PSoC 4100M / PSoC 4200M / PSoC 4200L families	–	–	±4	mV
V_{HYST}	Hysteresis when enabled		–	10	35	mV
V_{HYST}	Hysteresis when enabled. Common Mode Voltage Range from 0 to VDD -1.	For PSoC 4100M / PSoC 4200M / PSoC 4200L families	–	10	35	mV
V_{ICM1}	Input common mode voltage in normal mode	Modes 1 and 2	0	–	V_{DDD} – 0.1	V
V_{ICM1}	Input common mode voltage in normal mode	Modes 1 and 2 For PSoC 4200L family	0	–	V_{DDD} – 0.2	V
V_{ICM2}	Input common mode voltage in low power mode		0	–	V_{DDD}	V
V_{ICM3}	Input common mode voltage in ultra low power mode		0	–	V_{DDD} – 1.15	V
V_{ICM3}	Input common mode voltage in ultra low power mode	For PSoC 4000S / PSoC 4100S families: VDDD ≥ 2.2 V at –40 °C	0	–	V_{DDD} – 1.15	V
CMRR	Common mode rejection ratio	VDDD ≥ 2.7V	50	–	–	dB
CMRR	Common mode rejection ratio	VDDD < 2.7V	42	–	–	dB
I_{CMP1}	Block current, normal mode		–	–	280	µA
I_{CMP1}	Block current, normal mode	For PSoC 4 BLE, PSoC 4100M / PSoC 4200M, and PSoC 4000S / PSoC 4100S families	–	–	400	µA
I_{CMP1}	Block current, normal mode	For PSoC 4200L family	-	280	400	µA
I_{CMP2}	Block current, low power mode		–	–	50	µA
I_{CMP2}	Block current, low power mode	For PSoC 4 BLE, PSoC 4100M / PSoC 4200M, and PSoC 4000S / PSoC 4100S families	–	–	100	µA
I_{CMP2}	Block current, low power mode	For PSoC 4200L family	–	50	100	µA
I_{CMP3}	Block current, ultra low power mode		–	–	6	µA

Parameter	Description	Conditions	Min	Typ	Max	Units
I_{CMP3}	Block current, ultra low power mode	For PSoC 4100S family: VDDD ≥ 2.2 V at –40 °C	–	–	6	µA
I_{CMP3}	Block current, ultra low power mode	For PSoC 4100M/ PSoC 4200M/ PSoC 4200L families	–	6	28	µA
I_{CMP3}	Block current, ultra low power mode	For PSoC 4000S family: VDDD ≥ 2.2 V at –40 °C	–	6	28	µA
Z_{CMP}	DC input impedance of comparator		35	–	–	MΩ

AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
T_{RESP1}	Response time, normal mode	50 mV overdrive	–	–	38	ns
T_{RESP1}	Response time, normal mode	50 mV overdrive, for PSoC 4100M / PSoC 4200M families	–	–	110	ns
T_{RESP1}	Response time, normal mode	50 mV overdrive, for PSoC 4200L and PSoC 4000S / PSoC 4100S families	–	38	110	ns
T_{RESP2}	Response time, low power mode	50 mV overdrive	–	–	70	ns
T_{RESP2}	Response time, low power mode	50 mV overdrive, for PSoC 4100M/ PSoC 4200M families	–	–	200	ns
T_{RESP2}	Response time, low power mode	50 mV overdrive, for PSoC 4200L and PSoC 4000S / PSoC 4100S families	–	70	200	ns
T_{RESP3}	Response time, ultra low power mode	200 mV overdrive	–	–	2.3	µs
T_{RESP3}	Response time, ultra low power mode	200 mV overdrive, for PSoC 4100M/ PSoC 4200M families	–	–	15	µs
T_{RESP3}	Response time, ultra low power mode	200 mV overdrive, for PSoC 4200L family		2.3	15	µs
T_{RESP3}	Response time, ultra low power mode	200 mV overdrive, for PSoC 4000S / PSoC 4100S families: VDDD ≥ 2.2 V at –40 °C		2.3	15	µs

Component Changes

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

Version	Description of Changes	Reason for Changes / Impact
2.20.c	Datasheet edits.	Added final characterization data for PSoC 4100S device.
2.20.b	Datasheet edits.	Updated the output terminal description.
2.20.a	Datasheet edits.	Updated information to include PSoC 4000S device.
2.20	Fixed an issue with fake interrupt during start.	In cases when Vin_positive is greater than Vin_negative for the first execution of LPComp_Start() (that is, the comparator's output becomes "high" immediately after start) and the Interrupt mode is set to "rising edge" or "both," a fake interrupt might have caused interrupt system malfunctioning (interrupts from LPComp will not appear).
	Added note about input signal correctness in Low Power Mode.	Clarification.
2.10.a	Minor datasheet update.	Updated information to include PSoC 4200L devices. Changed the description text for the characterization section.
2.10	Added support for future devices.	Changes were made to allow for support of future devices. There is no impact to the Component at this time other than version number change.
2.0.b	Datasheet edits.	Updated information to include PSoC 4100M and PSoC 4200M devices. Updated block diagram in the Functional Description section. Updated description of the Output Configuration. Updated DC and AC Electrical Characteristics section with PSoC 4100M/ PSoC 4200M data.
2.0.a	Datasheet edits.	Updated characterization data for BLE devices. Clarified that the new APIs are not applicable to PSoC 4100 / PSoC 4200 devices: LPComp_SetInterruptMask(), LPComp_GetInterruptMask(), LPComp_GetInterruptSourceMasked()
2.0	Added new APIs: LPComp_SetInterruptMask(), LPComp_GetInterruptMask(), LPComp_GetInterruptSourceMasked()	Updates to support PSoC 4200 BLE devices.

Version	Description of Changes	Reason for Changes / Impact
	Added Output Configuration settings on General Tab. For manual output control the LPComp_SetOutputMode() API was added.	
	Updated Functional Description section.	
	Added Advanced Tab with Legacy output position settings.	This provides backwards compatibility with PSoC 4100 / PSoC 4200 designs.
	Added note on customizer that in Hibernate mode when Speed/Power selection is set to “Slow/Ultra low” the "Rising edge" and "Falling edge" selections are not available.	In Hibernate mode the comparator output will actually get an interrupt for both edges even rising or falling edge options are set. The issue only applies to PSoC 4100 / PSoC 4200 devices. There is no problem if the comparator is not used in Hibernate mode or if the Pulse/Interrupt configuration is already set Both edges.
	Speed/Power settings on the customizer and description of the defines for SetSpeed() API were renamed to “Slow/Ultra low”, “Medium/Low” and “Fast/Normal” to match device datasheet.	
	Filter option was removed from the Component General Tab. SetFilter() API preserved for backward compatibility, but is not recommended for use.	
1.0.b	Updated description of the Interrupt output	
1.0.a	Updated datasheet.	Removed the $V_{OFFSET1}$ spec.
1.0	New Component	

© 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.