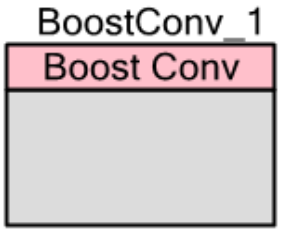
Boost Converter (BoostConv)

2.10



Features

- Produces a selectable output voltage that is higher than the input voltage
- Input voltage range between 0.5 V and 5.5 V
- Boosted output voltage range between 1.8 V and 5.25 V
- Source up to 50 mA depending on the selected input and output voltage parameter values
- Two modes of operation: Active and Standby
- Boost Converter component is not supported on PSoC 5

General Description

The Boost Converter (BoostConv) component allows you to configure and control the PSoC boost converter hardware block. The boost converter enables input voltages that are lower than the desired system voltage to be boosted to the desired system voltage level. The converter uses an external inductor to convert the input voltage to the desired output voltage.

The BoostConv component is enabled by default at chip startup with an output voltage of 1.8 V for PSoC 3, and of 1.9 V for PSoC 5LP. This allows the chip to start up in scenarios where the input voltage to the boost is below the minimum allowable voltage to power the chip. The configuration parameters defined in the component customizer (default V_{IN} = 1.8 V, V_{OUT} = 3.3 V, Switching Frequency = 400 kHz) will not take effect until the BoostConv_Start() API is called. The BoostConv component parameters can also be adjusted during run time using the provided APIs.

The boost converter has two main operating modes:

- **Active** – Active mode is the normal mode of operation where the boost regulator actively generates a regulated output voltage.
- **Standby** – Standby mode is a low-power mode of operation.

When to Use the Boost Component

Use the BoostConv component when the available voltage source for a system is less than the required voltage level to operate the system. The BoostConv component accepts a battery or other input voltage and produces a higher output voltage.

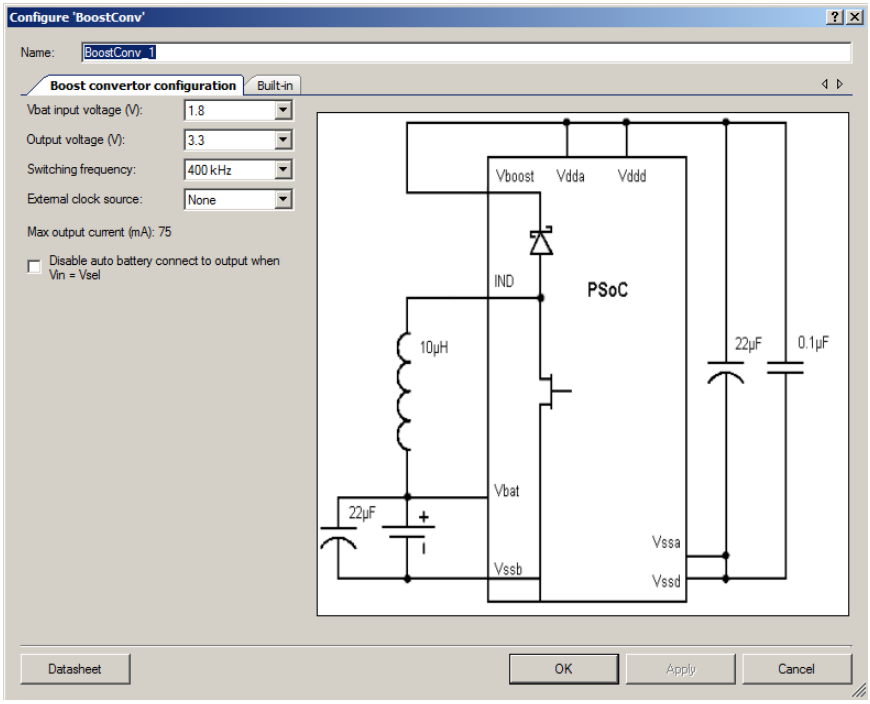
As an example, the system may use a 0.5-V solar cell as the primary power source and rely on the boost block to power the 1.8-V PSoC core. In another application, a 3.3-V system could use the BoostConv component to power a 5.0-V LCD glass.

Input/Output Connections

The BoostConv component requires no connections in the project schematic view. Fixed-function pins support the boost converter block circuit. The system circuit must provide connections for the input voltage (Vbat), output voltage (Vout), and inductor pin (Ind), and battery ground (Vssb). Refer to the schematic representation shown in the Functional Description section.

Component Parameters

Drag a BoostConv component onto your design and double-click it to open the **Configure** dialog.



Vbat input voltage (V)

This is the V_{BAT} or other voltage source that is used as the input voltage to the boost converter block. This system circuit connects this voltage to the Vbat PSoC pin. The input voltage can be between 0.5 V and 5.5 V. This value is used to calculate the estimated maximum output current. The default value is 1.8 V.

Output voltage (V)

This is the target output voltage that the boost converter block will maintain. Use the drop-down list to select the desired output voltage. Output voltage levels are provided in 0.1-V increments from 1.8 V to 3.6 V and in 0.25-V increments from 4.00 V to 5.25 V. The default value is 3.3 V.

An external Schottky diode is required for output voltages above 3.6 V.

The output voltage value can be modified at run time by using the BoostConv_SelVoltage() function.

Switching frequency

This is the switching frequency at which the boost converter block will operate. This value is an enumerated type and can be set to any of the following frequencies:

- 100 kHz
- 400 kHz
- 2 MHz
- External 32 kHz

The 100-kHz, 400-kHz, and 2-MHz switching frequencies are generated using an oscillator internal to the boost converter block. The External 32-kHz switching frequency is intended for Standby mode automatic thump regulation.

External clock source

The External 32 kHz to the Boost frequency is the source of the switching signal when the boost converter block is configured to use an external clock. This value can be set to any of the following frequencies:

- None
- ECO 32kHz
- ILO 32kHz

Note The external 32-kHz clock (ECO or ILO) selection is only supported in PSoC 3 and PSoC 5LP Production silicons.

Max output current (mA)

This is an estimate of the maximum output current available from the boost converter block based on the specified **Vbat input voltage** and **Output voltage** values. This is a read-only value.

Disable auto battery connect to output when Vin = Vsel

When set, disables auto battery connection to output when Vin = Vsel. For more details, see Functional Description.

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 “BoostConv_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 “BoostConv.”

Function	Description
BoostConv_Start()	Starts the BoostConv component and puts the boost block into Active mode.
BoostConv_Stop()	Disables the BoostConv component. Turns off power to the boost converter circuitry.
BoostConv_EnableInt()	Enables the boost block undervoltage interrupt generation.
BoostConv_DisableInt()	Disables the boost block undervoltage interrupt generation.
BoostConv_SetMode()	Sets the boost converter mode to Active or Standby.
BoostConv_SelVoltage()	Selects the target output voltage the boost converter will maintain.
BoostConv_SelFreq()	Sets the switching frequency to one of four possible values: 100 kHz, 400 kHz, and 2 MHz (generated internal to the boost converter block) or 32 kHz (sourced external to the boost converter block from the chip ECO-32kHz or ILO-32kHz oscillator).
BoostConv_EnableAutoThump()	Enables automatic thump mode (only available when the boost block is in Standby mode and the switching frequency is set to 32 kHz).
BoostConv_DisableAutoThump()	Disables automatic thump mode.
BoostConv_ManualThump()	Forces a single pulse of the boost converter switch transistors.
BoostConv_ReadStatus()	Returns the boost block status register.
BoostConv_SelExtClk()	Sets the source of 32-kHz frequency: the 32-kHz ECO or 32-kHz ILO.
BoostConv_ReadIntStatus()	Returns the contents of the boost block interrupt status register.
BoostConv_Init()	Initializes BoostConv registers with initial values provided from customizer.
BoostConv_Enable()	This function enables the boost block (only valid when in Active mode). Component is enabled by default.
BoostConv_Disable()	Disables the boost block.

Global Variables

Function	Description
BoostConv_initVar	Indicates whether the Boost Converter has been initialized. The variable is initialized to 0 and set to 1 the first time BoostConv_Start() is called. This allows the component to restart without reinitialization after the first call to the BoostConv_Start() routine. If reinitialization of the component is required, then the BoostConv_Init() function can be called before the BoostConv_Start() or BoostConv_Enable() function.

void BoostConv_Start(void)

Description: Starts the BoostConv component and puts the boost block into Active mode. The component is in this state when the chip powers up. This is the preferred method to begin component operation. BoostConv_Start() sets the initVar variable, calls the BoostConv_Init() function, and then calls the BoostConv_Enable() function.

Parameters: None

Return Value: None

Side Effects: If the initVar variable is already set, this function: (1) Sets the initial value of the target output voltage (from the customizer) and mode (Active mode) or restores target output voltage and mode saved in the BoostConv_Stop() function; (2) Calls the BoostConv_Enable() function.

void BoostConv_Stop(void)

Description: Saves boost converter target output voltage and mode. Disables the BoostConv component.

Parameters: None

Return Value: None

Side Effects: Turns off power to the boost converter circuitry. Sets the boost converter to Standby mode.

void BoostConv_EnableInt(void)

Description: This function enables the boost block output undervoltage interrupt generation.

Parameters: None

Return Value: None

Side Effects: None

void BoostConv_DisableInt(void)

Description: This function disables the boost block output undervoltage interrupt generation.

Parameters: None

Return Value: None

Side Effects: None

void BoostConv_SetMode(uint8 mode)

Description: This function sets the boost converter mode: Active or Standby.

Parameters: uint8 mode: Sets the operational mode for the boost block:

Mode	Notes
BoostConv_BOOSTMODE_ ACTIVE	In the active mode, the boost block maintains the selected output voltage.
BoostConv_BOOSTMODE_STANDBY	Low power state, only bandgap and comparator circuitry is active. Automatic thump mode is used with the external 32-kHz clock to regulate output voltage

Return Value: None

Side Effects: For Standby mode, this function enables automatic thump mode and sets the switching frequency clock source to the 32-kHz external clock. For Active mode this function disables automatic thump mode.

void BoostConv_SelVoltage(uint8 voltage)

Description: This function selects the target output voltage the boost converter will maintain.

Parameters: uint8 voltage: The target output voltage for the boost converter block. Output voltages above 3.6 V require an external Schottky diode.

Power Setting	Value	Notes
BoostConv_VOUT_OFF	0x00	Off – HI-Z
BoostConv_VOUT_1_8V	0x03	1.8 V
BoostConv_VOUT_1_9V	0x04	1.9 V
BoostConv_VOUT_2_0V	0x05	2.0 V
BoostConv_VOUT_2_1V	0x06	2.1 V
BoostConv_VOUT_2_2V	0x07	2.2 V
BoostConv_VOUT_2_3V	0x08	2.3 V
BoostConv_VOUT_2_4V	0x09	2.4 V
BoostConv_VOUT_2_5V	0x0A	2.5 V
BoostConv_VOUT_2_6V	0x0B	2.6 V
BoostConv_VOUT_2_7V	0x0C	2.7 V
BoostConv_VOUT_2_8V	0x0D	2.8 V
BoostConv_VOUT_2_9V	0x0E	2.9 V
BoostConv_VOUT_3_0V	0x0F	3.0 V
BoostConv_VOUT_3_1V	0x10	3.1 V
BoostConv_VOUT_3_2V	0x11	3.2 V
BoostConv_VOUT_3_3V	0x12	3.3 V
BoostConv_VOUT_3_4V	0x13	3.4 V
BoostConv_VOUT_3_5V	0x14	3.5 V
BoostConv_VOUT_3_6V	0x15	3.6 V
BoostConv_VOUT_4_0V	0x16	4.00 V (external Schottky diode required)
BoostConv_VOUT_4_25V	0x17	4.25 V (external Schottky diode required)
BoostConv_VOUT_4_5V	0x18	4.50 V (external Schottky diode required)
BoostConv_VOUT_4_75V	0x19	4.75 V (external Schottky diode required)
BoostConv_VOUT_5_0V	0x1A	5.00 V (external Schottky diode required)
BoostConv_VOUT_5_25V	0x1B	5.25 V (external Schottky diode required)

Return Value: None

Side Effects: Function will take affect only when the Boost is enabled.

void BoostConv_SelFreq(uint8 frequency)

Description: This function sets the switching frequency to one of four possible values.

Parameters: uint8 switch_freq: The desired switching frequency.

Switch Frequency	Notes
BoostConv__SWITCH_FREQ_100KHZ	Generated internal to the boost converter block with a dedicated oscillator
BoostConv__SWITCH_FREQ_400KHZ	
BoostConv__SWITCH_FREQ_2MHZ	
BoostConv__SWITCH_FREQ_32KHZ	Comes from the ECO-32kHz or ILO-32kHz

Return Value: None

Side Effects: None

void BoostConv_EnableAutoThump(void)

Description: This function enables automatic thump mode. The AutoThump mode is available only when the boost block is in the Standby mode. The switching frequency clock source for the boost block must be set to the 32-kHz external clock. In this mode, standby boost operation is accomplished by generating a boost switch pulse on each edge of the switching clock when the output voltage is below the selected value.

Parameters: None

Return Value: None

Side Effects: None

void BoostConv_DisableAutoThump(void)

Description: This function disables automatic thump mode.

Parameters: None

Return Value: None

Side Effects: None

void BoostConv_ManualThump(void)

Description: This function forces a single pulse of the boost converter switch transistors.

Parameters: None

Return Value: None

Theory:

Side Effects: Thump produces one ~500-ns pulse when set. This routine writes a '0' followed by a '1' to the bit 7 "thump" bit in the boost block BOOST_CR0 register.

uint8 BoostConv_ReadStatus(void)

Description: This function returns the contents of the boost block status register.

Parameters: None

Return Value: uint8 boost block status register: BOOST_SR:

Bit	Name	Description
7	BoostConv_RDY	When set, internal circuits have been initialized
6	BoostConv_START	When set, converter is in startup mode
5	–	Reserved
4	BoostConv_OV	Output above overvoltage limit when 1, below limit when 0
3	BoostConv_VHI	Output is above vhigh limit when 1, below limit when 0
2	BoostConv_VNOM	Output is above nominal when 1, below nominal when 0
1	BoostConv_VLO	Output is above vlow limit when 1, below limit when 0
0	BoostConv_UV	Output is above undervoltage limit when 1, below limit when 0

Side Effects: None

void BoostConv_SelExtClk(uint8 source)

Description: This function sets the source of 32-kHz frequency: the chip's ECO-32kHz or ILO-32kHz.

Parameters: uint8 source: The source of 32-kHz frequency.

Name	Description
BoostConv__EXTCLK_ECO	Set chip ECO-32kHz as the source of 32-kHz frequency
BoostConv__EXTCLK_ILO	Set chip ILO-32kHz as the source of 32-kHz frequency

Return Value: None

Side Effects: None

void BoostConv_ReadIntStatus(void)

Description: This function returns the contents of the boost block interrupt status register.

Parameters: None

Return Value: uint8 Boost interrupt status register BOOST_SR2 bit 0: When set, a Boost Output Undervoltage event has occurred.

Side Effects: None

void BoostConv_Init(void)

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

Parameters: None

Return Value: None

Side Effects: All registers will be set to values according to the customizer Configure dialog.

void BoostConv_Enable(void)

Description: This function enables the boost block when in Active mode. The component is enabled by default. Activates the hardware and begins component operation. It is not necessary to call BoostConv_Enable() because the BoostConv_Start() API calls this function, which is the preferred method to begin component operation.

Parameters: None

Return Value: None

Side Effects: None

void BoostConv_Disable(void)

Description: This function disables the boost block.

Parameters: None

Return Value: None

Side Effects: None

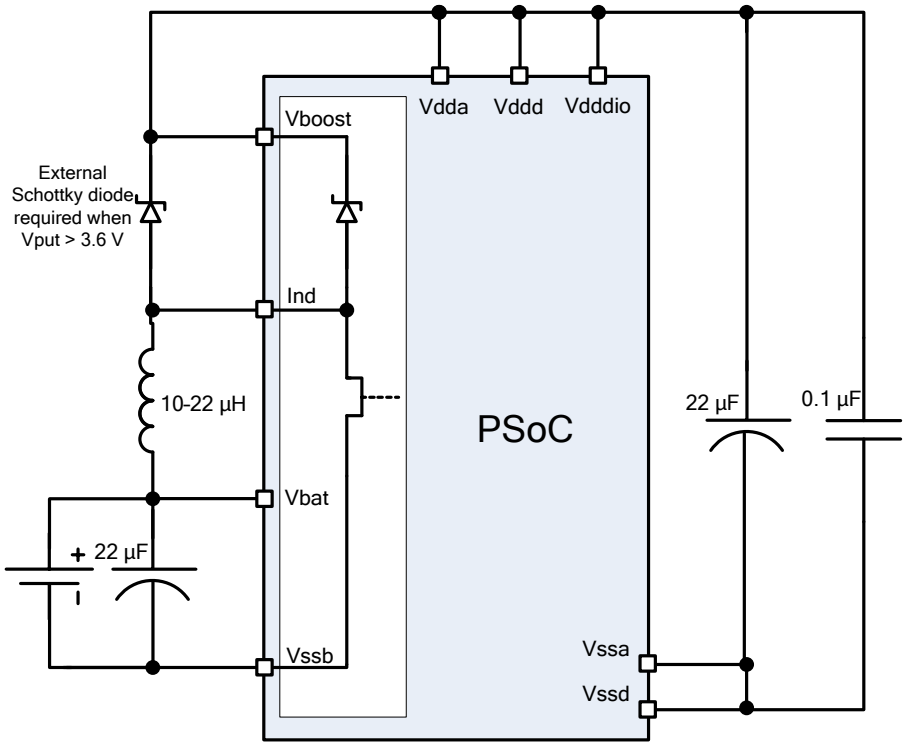
Sample Firmware Source Code

PSoC Creator provides many 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.

Functional Description

Figure 1. Application for Boost Converter



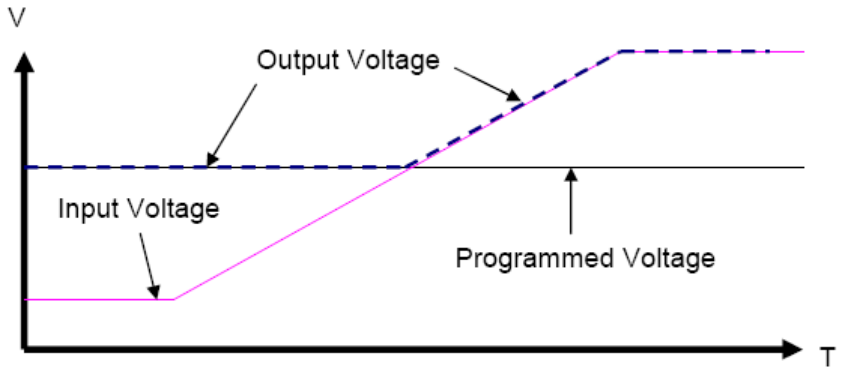
The boost block circuit is enabled by default to support scenarios in which processor startup is powered by the V_{BOOST} voltage. The boost block is configured for Active mode with an output voltage of 1.8 V for PSoC 3, and of 1.9 V for PSoC 5LP by default. When a BoostConv component is placed in a project, it provides access to the configuration registers for the boost hardware block. The BoostConv_Start() function configures the BoostConv component with the settings made in the component configuration dialog.

Boost block registers are powered from the V_{BOOST} supply. The V_{BOOST} output cannot be allowed to fall below 1.4 V. You can expect to lose register contents if V_{BOOST} is allowed to fall below 1.4 V. The firmware must reload these registers if such these conditions occur.

The boost converter hardware uses the fixed-function pins on the chip shown in the schematic above. These signals are not shown on the BoostConv component.

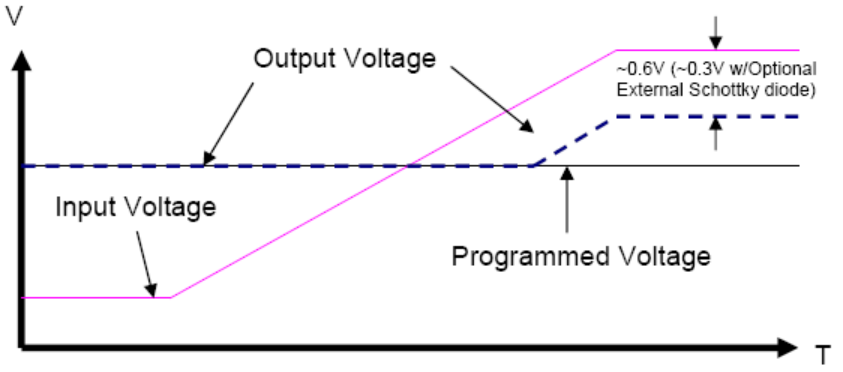
Operation with Input Voltage Greater Than Programmed Output Voltage

When Control register 2 (BOOST_CR2): Bit 1 (eqoff) = 0, the output voltage will track the input voltage when the input is greater than the programmed output voltage. This is shown below:



- Output Voltage = Programmed voltage when Input < Programmed
- Output Voltage = Input voltage when Input > Programmed

When Control register 2 (BOOST_CR2): Bit 1 (eqoff) = 1, the output voltage does not track until either the optional external Schottky diode or inherent internal silicon diode between the inductor pin and output are forward biased. The effect of this is that the output voltage tracks input with a diode drop, as shown below:



- Output Voltage = Programmed voltage when Input < Programmed
- Output Voltage = Input voltage – diode drop when Input > Programmed + diode drop

Resources

The BoostConv component uses the dedicated boost converter hardware block in the silicon.

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 the 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.

Configuration	PSoC 3 (Keil_PK51)		PSoC 5 (GCC)		PSoC 5LP (GCC)	
	Flash Bytes	SRAM Bytes	Flash Bytes	SRAM Bytes	Flash Bytes	SRAM Bytes
Default	324	3	N/A	N/A	484	5

DC and AC Electrical Characteristics

Unless otherwise specified in the tables below, operating conditions are: V_{BAT} = 2.4 V, V_{OUT} = 2.7 V, I_{OUT} = 40 mA, F_{SW} = 400 kHz, L_{BOOST} = 10 µH, C_{BOOST} = 22 µF || 0.1 µF.

DC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
V_{BAT}	Input voltage includes startup	T = –35 °C to +65 °C	0.5	–	3.6	V
		Over entire temperature range	0.68	–	3.6	V
I_{OUT}	Load current [1, 2]	V_{BAT} = 1.6–3.6 V, V_{OUT} = 3.6–5.0 V, external diode	–	–	50	mA
		V_{BAT} = 1.6–3.6 V, V_{OUT} = 1.6–3.6 V, internal diode	–	–	75	mA
		V_{BAT} = 0.8–1.6 V, V_{OUT} = 1.6–3.6 V, internal diode	–	–	30	mA

1. For output voltages above 3.6 V, an external diode is required.
2. Maximum output current applies for output voltages ≤ 4x input voltage.

Parameter	Description	Conditions	Min	Typ	Max	Units
		V_{BAT} = 0.8–1.6 V, V_{OUT} = 3.6–5.0 V, external diode	–	–	20	mA
		V_{BAT} = 0.5–0.8 V, V_{OUT} = 1.6–3.6 V, internal diode	–	–	15	mA
I_{LPK}	Inductor peak current		–	–	700	mA
I_Q	Quiescent current	Boost active mode	–	200	–	µA
		Boost standby mode, 32-kHz external crystal oscillator, I_{OUT} < 1 µA	–	12	–	µA
V_{OUT}	Boost output voltage range [3, 4]					
	1.8 V		1.71	1.80	1.89	V
	1.9 V		1.81	1.90	2.00	V
	2.0 V		1.90	2.00	2.10	V
	2.4 V		2.28	2.40	2.52	V
	2.7 V		2.57	2.70	2.84	V
	3.0 V		2.85	3.00	3.15	V
	3.3 V		3.14	3.30	3.47	V
	3.6 V		3.42	3.60	3.78	V
	5.0 V	External diode required	4.75	5.00	5.25	V
Reg_{LOAD}	Load regulation		–	–	3.8	%
Reg_{LINE}	Line regulation		–	–	4.1	%
η	Efficiency	L_{BOOST} = 10 µH	70	85	–	%
		L_{BOOST} = 22 µH	82	90	–	%

3. Based on device characterization (Not production tested).

4. At boost frequency of 2 MHz, V_{BOOST} is limited to 2 × V_{BAT}. At 400 kHz, V_{BOOST} is limited to 4 × V_{BAT}.

AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
V_{RIPPLE}	Ripple voltage (peak-to-peak)	V_{OUT} = 1.8 V, F_{SW} = 400 kHz, I_{OUT} = 10 mA	–	–	100	mV
F_{SW}	Switching frequency		–	0.1, 0.4, or 2	–	MHz

Recommended External Components for Boost Circuit

Parameter	Description	Conditions	Min	Typ	Max	Units
L_{BOOST}	Boost inductor		4.7	10	47	µH
C_{BOOST}	Filter capacitor		10	22	47	µF
I_F	External Schottky diode average forward current	External Schottky diode is required for V_{OUT} > 3.6 V	1	–	–	A
V_R			20	–	–	V

Component Changes

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

Version	Description of Changes	Reason for Changes / Impact
2.10.a	Minor datasheet edit.	
2.10	Added PSoC 5LP device support.	
	Added all BoostConv APIs with CYREENTRANT keyword when they included in .cyre file.	Not all APIs are truly reentrant. Comments in the component API source files indicate which functions are candidates. This change is required to eliminate compiler warnings for functions that are not reentrant used in a safe way: protected from concurrent calls by flags or Critical Sections.
2.0.a	Datasheet corrections.	
2.0	Removed reference to PSoC 5 support.	Component is not supported by PSoC 5.
	Updated BoostConv_Start() and BoostConv_Stop() functions with set/restore mode and voltage.	An expected use case is using Boost to power off-chip devices, so firmware is expected to start/stop the boost.
	Added new parameter “Disable auto battery connect to output when Vin = Vsel.” Updated BoostConv_Init() function with disabling auto battery connect to output when $V_{IN} = V_{SEL}$.	To allow the user to configure whether the output voltage should track the input voltage when $V_{BAT} > V_{BOOST}$, or only track after the diode is forward biased (so the output tracks the input - diode drop).
1.50.a	Added autothump support note to Features in datasheet.	Silicon bug in PSoC 3 ES2 and PSoC 5.
	Added information to the component that advertizes its compatibility with silicon revisions.	The tool reports an error/warning if the component is used on incompatible silicon. If this happens, update to a revision that supports your target device.

Version	Description of Changes	Reason for Changes / Impact
	Added characterization data to datasheet.	
	Removed reference to sleep mode from datasheet.	Component doesn't support sleep mode.
	Minor datasheet edits and updates.	

1.50	Added support of PSoC 3 Production silicon. Three API functions have been added:
	void BoostConv_EnableInt(void);
	void BoostConv_DisableInt(void);
	uint8 BoostConv_ReadIntStatus(void);
		Boost Converter supports generation of undervoltage signal.

	API function has been added: void BoostConv_SelExtClk(uint8);	To support selection of external switching clock sources of Boost Converter: ILO or ECO.
	Added BoostConv_Init() function.	To comply with corporate standard and provide an API to initialize/restore the component without starting it.

© Cypress Semiconductor Corporation, 2012-2015. 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® Creator™, Programmable System-on-Chip™, and PSoC Express™ are trademarks and PSoC® is a registered trademark 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.