PSoC™ 4 lookup table

About this document

Scope and purpose

This example demonstrates how to set up the LUT Component to specify custom combinatorial functions and state machines in a PSoC™ 4 device. Control and status registers are used to show how the LUT functions. In this example, the LUT is set up to behave like a 3-bit up/down counter. Output values are displayed on the terminal via a UART.

Requirements

Tool: **PSoC™ Creator** 4.4

Programming Language: C (Arm® GCC 5.4.1)

Associated Parts: **PSoC 4** devices with universal digital blocks (UDB).

Related Hardware: **CY8CKIT-042 PSoC 4 pioneer kit**

Table of contents

About this document **1**
Table of contents **1**
1 Hardware and software setup **2**
1.1 Hardware setup 2
1.2 Software setup 2
2 Operation **3**
3 Design and Implementation **4**
3.1 Components and settings 4
4 Reusing this example **6**
References **7**
Document history **8**

1 Hardware and software setup

1.1 Hardware setup

This code example is set up for CY8CKIT-042. If you are using a different kit, see **Reusing this example**. In this kit, the USB-UART bridge in the KitProg2 module is used:

1. Connect the \UART:rx\ pin P0[4] to P12[7] on header J8.
2. Connect the \UART:tx\ pin P0[5] to P12[6] on header J8.

Other kits use different pins for the UART. Make sure that you select the correct pins for your kit.

1.2 Software setup

This design requires a terminal emulator such as PuTTY or Tera Term running on your computer.

2 Operation

1. Ensure that the correct pins are connected for your kit, as noted in the **Hardware and software setup** section.
2. Connect the USB cable between the PC and the PSoC 4 pioneer kit.
3. Build the project and program it into the PSoC 4 device. Choose **Debug** > **Program**. For more information on device programming, see PSoC Creator Help.
4. Open a terminal emulator on your computer and configure the program to the appropriate COM port. Configure the baud rate to 115200, data bits to 8, no parity bits, stop bit as 1, and no control flow.
5. Each time SW2 is pressed, observe that the mode changes between counting up or down, as **Figure 1** shows:

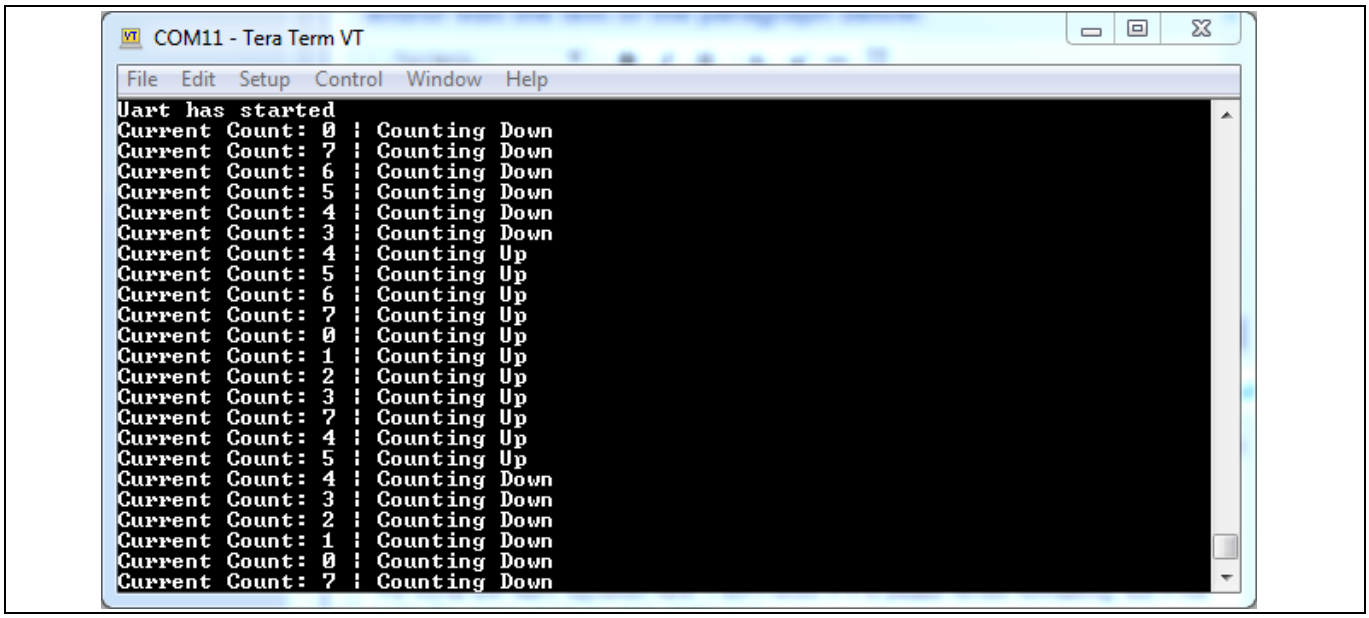


Figure 1 Project terminal output

3 Design and Implementation

In this example, the LUT is used to implement a 3-bit up/down counter, as **Figure 2** shows. The LUT is configured with four inputs: in0, in1, and in2 are feedback inputs, in3 controls the counting mode (up or down). The LUT configuration table (**Figure 3**) implements the bidirectional counter.

The firmware does the following:

- Sets up UART operation and interrupt handler
- Updates the control register and LUT output
- Displays the LUT output and count mode, if there is a change in the LUT output

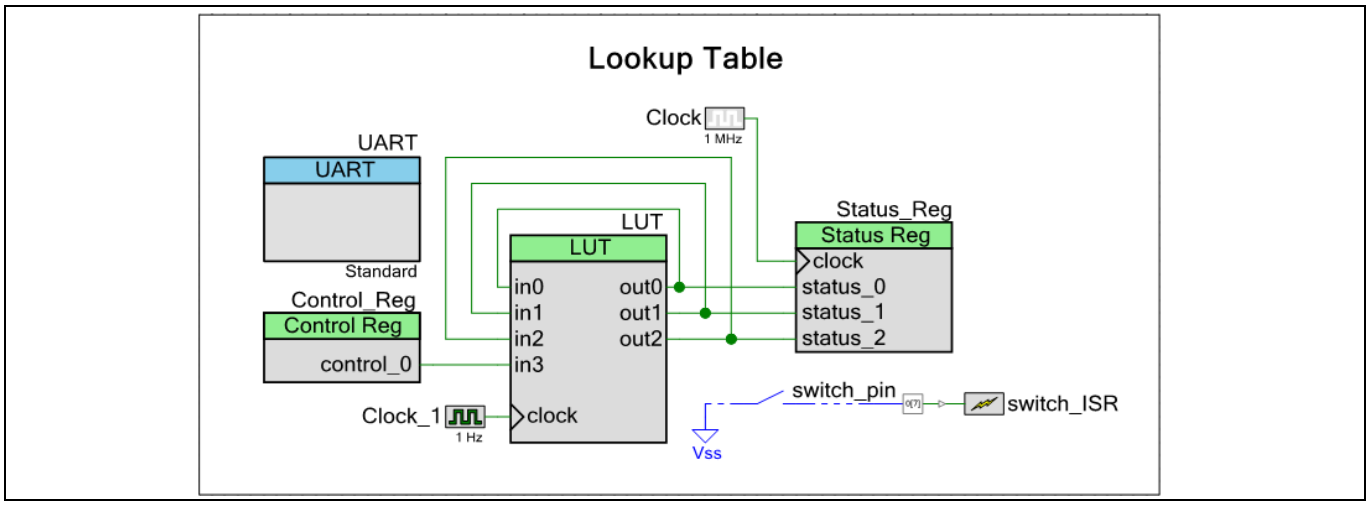


Figure 2 PSoC Creator project schematic

3.1 Components and settings

Table 1 lists the PSoC Creator Components used in this example, how they are used in the design, and the non-default settings required so they function as intended.

Table 1 PSoC Creator Components

Component	Instance name	Purpose	Non-default settings
Lookup Table	LUT	Handles Lookup Table operations	See **Figure 3**. Check the **Register Outputs** box.
Control Register	Control_Reg	CPU writes digital signals	Set **Outputs** to 3
Status Register	Status_Reg	CPU reads digital signals	Set **Inputs** to 3
UART (SCB mode)	UART	Handles UART operation	None

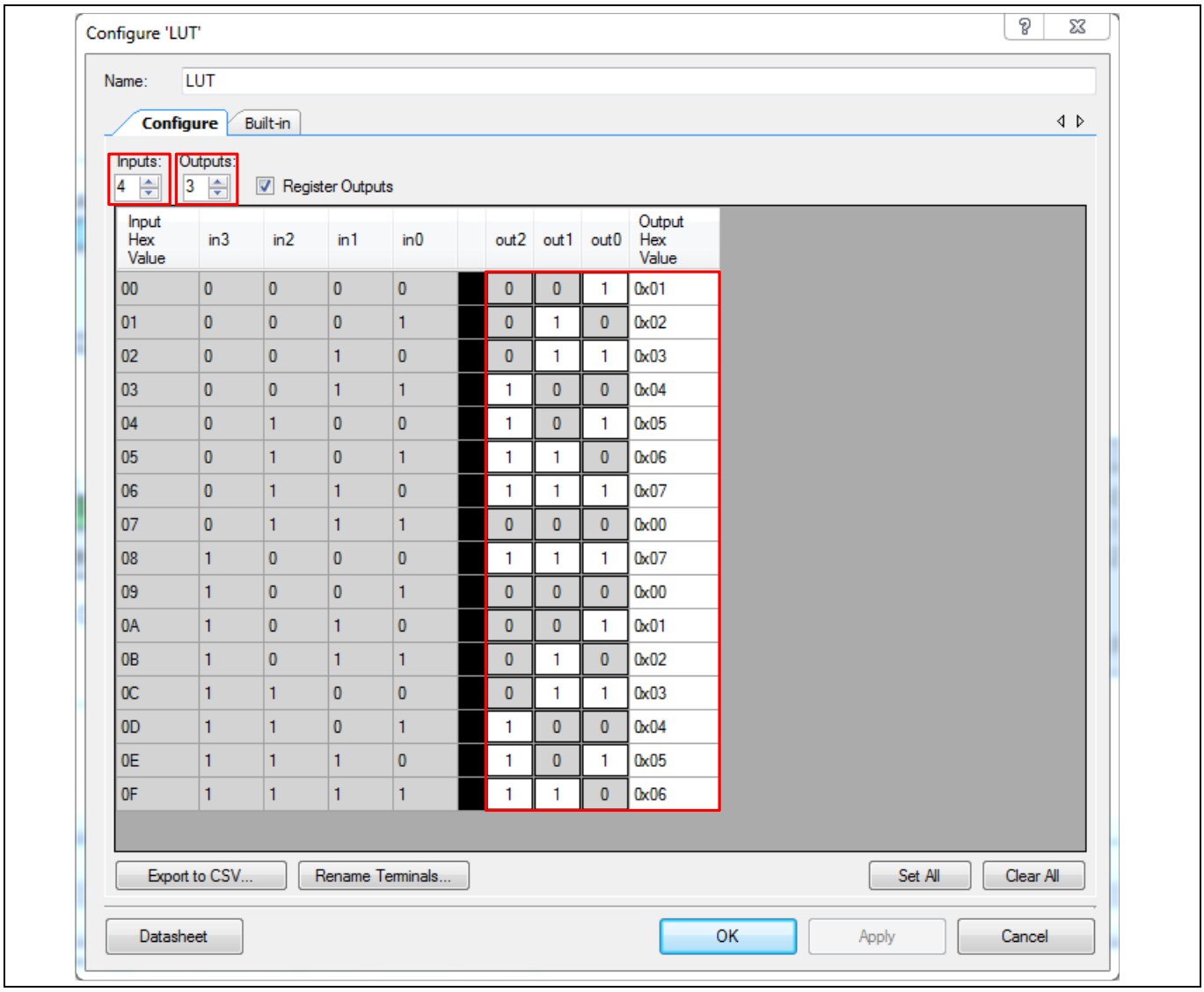


Figure 3 LUT parameter settings

For information on the hardware resources used by a Component, see the Component datasheet.

4 Reusing this example

This example is designed for the CY8CKIT-042 pioneer kit. To port the design to a different PSoC 4 device, kit, or, both, do the following:

1. In PSoC Creator, select **Project** > **Device Selector** to change the target device. Select your device as listed in **Table 2**.

Table 2 Development kits and associated devices

Development kit	Device
CY8CKIT-042	CY8C4245AXI-483
CY8CKIT-042-BLE-A	CY8C4248LQI-BL583
CY8CKIT-044	CY8C4247AZI-M485
CY8CKIT-046	CY8C4248BZI-L489

2. Change the Resources Pins settings as needed. **Table 3** shows the pin assignments required for UART operation on other PSoC 4 devices.

Table 3 Pin assignments for different kits

Pin Name	Development kit			
	CY8CKIT-042	**CY8CKIT-042-BLE-A**	**CY8CKIT-044**	**CY8CKIT-046**
\UART:rx\	P0[4]	P1[4]	P7[0]	P3[0]
\UART:tx\	P0[5]	P1[5]	P7[1]	P3[1]

Additionally, for CY8CKIT-042, the PSoC™ 4 UART signals need to be physically connected to KitProg2 (PSoC™ 5LP) UART signals using jumper wires. Connect P12[7] and P12[6] on header J8 to P0[4] and P0[5] on header J4. For other kits, the pins are already connected.

Note: This project cannot be built for PSoC 4 devices with no UDBs.

In some cases, a resource used by a code example (for example, a Universal Digital Block (UDB)) is not supported on another device. In that case the example will not work. If you build the code targeted at such a device, you will get errors. See the device datasheet for information on what a device supports.

References

Application notes

[1] **AN79953** – Getting started with PSoC 4: Describes PSoC 4 and shows how to build the attached code example.

PSoC Creator Component datasheets

[2] **Lookup Table:** This Component can be set up to perform any logic function with up to five inputs and eight outputs. Outputs can be registered.

[3] **SCB:** A multifunction hardware block that implements the following communication components: I2C, SPI, UART, and EZI2C.

[4] **Control Register:** The Control Register allows the firmware to output digital signals.

[5] **Status Register:** The Status Register allows the firmware to read digital signals.

Device documentation

[6] **PSoC 4 datasheets**

[7] **PSoC 4 technical reference manuals**

Development kit (DVK) documentation

[8] **CY8CKIT-042 PSoC 4 pioneer kit**

[9] **PSoC 4 kits**

Tool documentation

[10] **PSoC™ Creator user guide**

[11] **PSoC™ Creator quick start guide**

Revision history

Document revision	Date	Description of change
**	2018-09-14	New code example
*A	2024-07-04	Update to Infineon template. Minor update to code comments. Updated weblinks.

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2024-07-04

Published by

Infineon Technologies AG

81726 München, Germany

© 2024 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document?

Go to: **erratum@infineon.com**

Document reference

002-24787 Rev. *A

IMPORTANT NOTICE

The information contained in this document is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this document must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this document.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.