AN89611

PSoC® 3 and PSoC 5LP – Getting Started With Chip Scale Packages (CSP)

Author: Mark Ainsworth

Associated Project: No

Associated Part Family: All PSoC 3 and PSoC 5LP devices in CSP packages

Software Version: PSoC Creator™ 3.0 and higher

Related Resources: For a complete list, click here.

To get the latest version of this application note, or the associated project file, please visit http://www.cypress.com/go/AN89611.

This application note provides guidelines for using Cypress PSoC® 3 and PSoC 5LP devices in wafer-level chip scale packages (CSP). Included are instructions for using the I²C bootloader that is factory installed in these devices.

Contents

Introduction 1
CSP Devices 2
Advantages and Considerations.....2
Best Practices for CSP.....3
PSoC Creator and CSP Devices.....4
Development Best Practices 4
Selecting CSP Devices.....6
Factory-Installed Bootloader 9
CSP Bootloader Specifications.....9
How to Use the Bootloader.....10
Summary.....16
Related Resources.....16
Appendix A –Schematics and PCB Footprints 17
Worldwide Sales and Design Support.....20

Introduction

Cypress is now offering its PSoC 3 and PSoC 5LP family of products in wafer-level chip scale packages (WLCSP, or CSP for short). These devices are designed to pack the maximum mixed-signal SoC capability per cubic millimeter. They feature package sizes as small as 4.25 × 4.98 × 0.6 mm to fit into tiny spaces on very small PCBs or flexible printed circuits (FPC). However, their small size mandates special manufacturing techniques and design considerations.

This application note gives an overview of guidelines and best practices for CSP. For more detailed guidance, see AN69061, Design, Manufacturing, and Handling Guidelines for WLCSP.

For a number of reasons (see Factory-Installed Bootloader), an I²C bootloader is factory-installed in the flash memory of all PSoC 3 and PSoC 5LP CSP devices. For more information on I²C bootloaders for PSoC, see AN60317, I²C Bootloader or AN73854, Introduction to Bootloaders.

PCB manufacturers with no specific knowledge of PSoC or PSoC Creator™, the design environment for PSoC, may use this application note as well asAN69061.

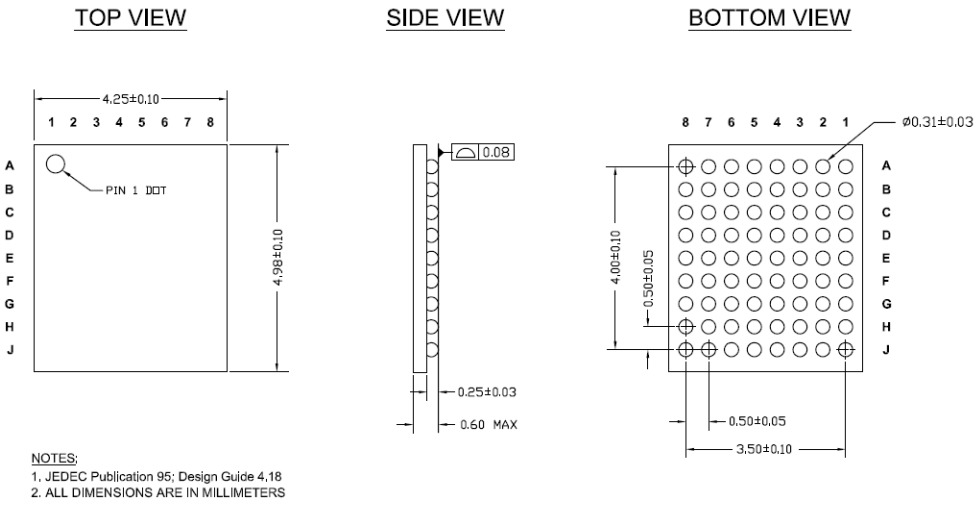
To use the factory-installed bootloader, a basic knowledge of PSoC 3 or PSoC 5LP and PSoC Creator is recommended. If you are new to PSoC, see AN54181, Getting Started with PSoC 3 or AN77759, Getting Started with PSoC 5LP. If you are new to PSoC Creator, see the PSoC Creator home page.

CSP Devices

Advantages and Considerations

The CSP is a true die-scale package that offers the smallest footprint and volume of any IC package. For example, for PSoC 3 the footprint of a 68-pin QFN package is 64 mm^2 (8 × 8 mm) while the equivalent CSP with 72 balls (Figure 1) is only 4.25 × 4.98 mm or 21.2 mm^2, for a 67% reduction in PCB footprint area. The package height is only 0.6 mm, which allows the device to be placed near or under connectors where PCB space is normally unused.

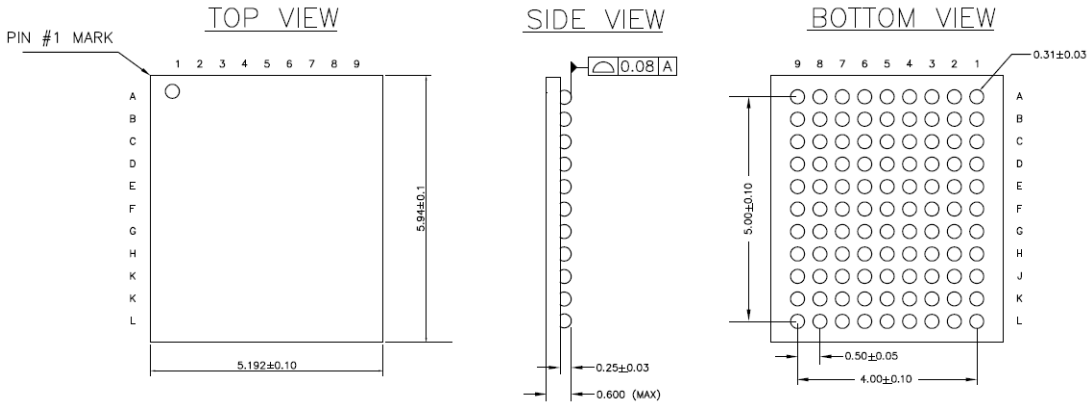
Figure 1. PSoC 3 72-CSP Package Diagram



For PSoC 5LP, the difference is even greater – an 84% reduction in footprint area on the PCB:

- 100-TQFP: 14 × 14 mm or 196 mm^2
- 99-CSP: 5.192 × 5.940 mm or 30.8 mm^2 (Figure 2); the package height is also only 0.6 mm

Figure 2. PSoC 5LP 99-CSP Package Diagram



Similar to ball grid array (BGA) packages, CSP devices have solder balls, or “bumps,” on the active circuitry side of the device. The CSP balls can be on 0.5-mm or 0.4-mm pitches; the PSoC packages have 0.5-mm pitches for easier PCB layout and lower manufacturing cost.

Although a CSP has the advantage of a smaller board surface area and use on FPCs, there are considerations for its use. It requires special handling processes and is difficult to rework, which increases manufacturing costs. The next section summarizes some best practices for using CSP devices; for details see AN69061, Design, Manufacturing, and Handling Guidelines for WLCSP.

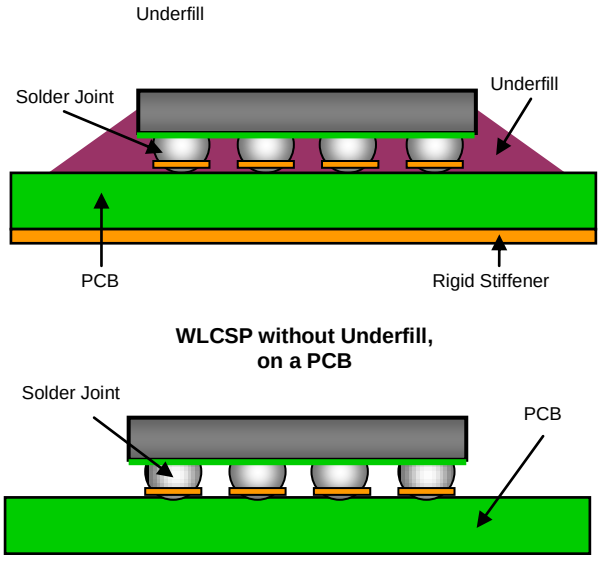
Best Practices for CSP

Underfill

Underfill is an epoxy used to fill the gaps between the CSP device and the PCB,to improve reliability.

A CSP device can be mounted on a PCB with or without underfill, as Figure 3 shows. However, using underfill is recommended. Without underfill, the difference in coefficient of thermal expansion (CTE) between the PCB and the device can cause increased mechanical stress during thermal cycles.

Figure 3. CSP Mounting With and Without Underfill



A reliable underfill evenly cocoons the solder joint and absorbs the CTE mismatch between the device and the board. Underfill protects the small bumped solder joint from excessive solder fatigue, unwanted package moisture, ionic contaminants, radiation, solder bump extrusion, thermal shock, mechanical shock, and vibration, which are all common aspects of surface mount technology (SMT).

Underfill does increase process and cost, but this is mitigated by the increase in reliability.

SMT Guidelines

Following are considerations and recommendations for the surface mounting of CSP devices:

- Stencil design: Follow the stencil design guidelines in IPC-7525.
- Electromagnetic shielding: Consider the dimensions between the WLCSP, PCB, pads, and components; type of shielding; shield design; and shield placement.
- Solder paste: Use the maximum volume of solder paste possible without risking solder paste bridging. This maximizes the standoff, or the distance between the device and the PCB, which in turn increases board-level reliability.
- Placement: Pick-and-place equipment may need to be optimized for CSP parts. Avoid setting a bond force, which can overdrive the package to the surface of the board and damage the part.

  Note Cypress ships all PSoC 3 and PSoC 5LP CSP parts on tape and reel instead of in trays, because of the potential for damage to the parts from pick-and-place machines.
- Reflow: All Cypress CSP devices are 100 percent Pb-free (Sn-Ag-Cu or Sn-Ag). Therefore, Pb-free solder paste with a Pb-free reflow profile is recommended.
- Rework: Rework of CSP devices is more difficult because of the exposed silicon substrate and the presence of underfill. It is important to use a controlled and qualified process to carry out rework to prevent mechanical and ESD damage to the device.

For a detailed discussion of each of these considerations, see AN69061, Design, Manufacturing, and Handling Guidelines for WLCSP as well as Amkor documentation.

ThermalResistance

CSPs have a junction-to-ambient thermal resistance (θ_{JA}) similar to that of QFN packages, with a significantly lower junction-to-case resistance (θ_{JC}). See a PSoC 3 or PSoC 5LP device datasheet for details.

The PSoC V_{DDIO} pin current limitations apply to CSPs in the same manner as the other PSoC packages. For details and example thermal calculations, see AN61290, Hardware Design Considerations.

PCB Layout

Laying out PCBs and FPCs for CSP requires special techniques. Start by confirming that your PCB fabricator supports high-density interconnect (HDI) – 1-mil trace and pitch capability is recommended. Traces from the inside rows and columns should be done on PCB internal layers, using via-in-pad, filled and plated; this is also known as “blind and buried” microvias.

See Related Resources, especially AN34359, for PCB layout tips and other information.

PSoC Creator and CSP Devices

This section gives instructions and best practices for using PSoC Creator, the design environment for PSoC 3 and PSoC 5LP, to create designs with CSP devices.

Development Best Practices

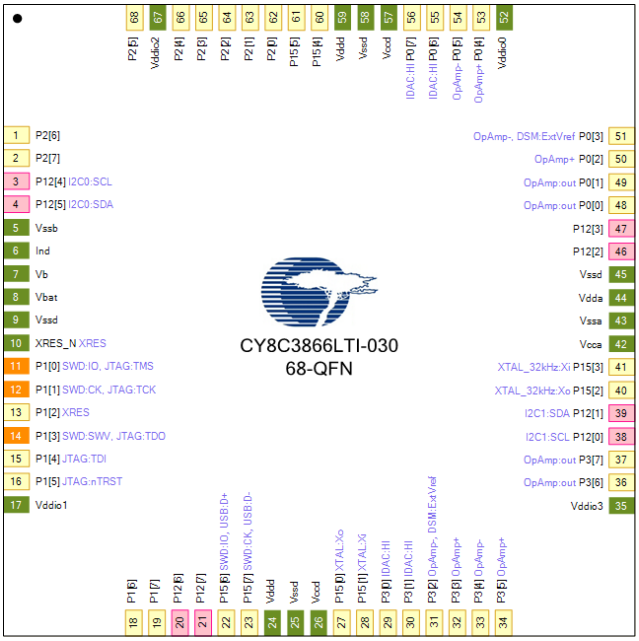
Although it is possible to develop a PSoC Creator project directly for the CSP, debugging such a project may be difficult, for reasons described in Factory-Installed Bootloader. A better method is to first develop and debug the project for a similar device in a 68-QFN or 100-TQFP package, and then port the project to a similar CSP device. This method has the following advantages:

- You can use a development kit, which gives you the ability to debug your design before your product PCB is available. The Cypress CY8CKIT-001, with daughter board modules for various devices, offers maximum flexibility. High-precision analog kits – CY8CKIT-030 or CY8CKIT-050 for PSoC 3 and PSoC 5LP, respectively – are also available.
- A powerful feature of PSoC 3 and PSoC 5LP is the ability to route any signal – analog or digital – to almost any I/O pin. This is very useful for debugging. Using one of these kits, you can route intermediate signals to spare pins for testing with an oscilloscope or logic analyzer.
- You can program and debug the device using the JTAG or SWD port, with an easy-to-use connector to a device such as the CY8CKIT-002 MiniProg3. Note that this overwrites the Factory-Installed Bootloader.

If you want to use your own PCB instead of a kit, the CSP pinouts have been designed to facilitate PCB layout for easy porting between packages. See the PSoC Creator screen shots in Figure 4 and Figure 5 on page 5. See also Appendix A for schematic and PCB footprint data.

Note The PSoC Creator diagrams of the devices show the CSPs in bottom (balls) view; all other devices are shown in top view.

Figure 4. PSoC 3 Pin Assignments for 68-QFN and 72-CSP



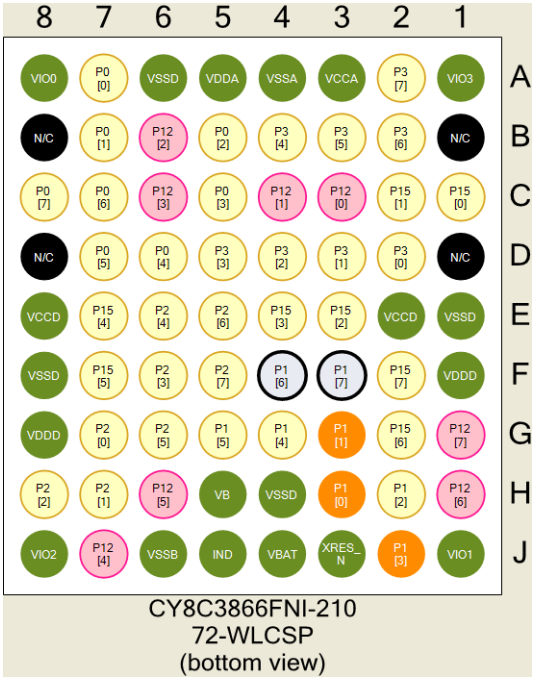
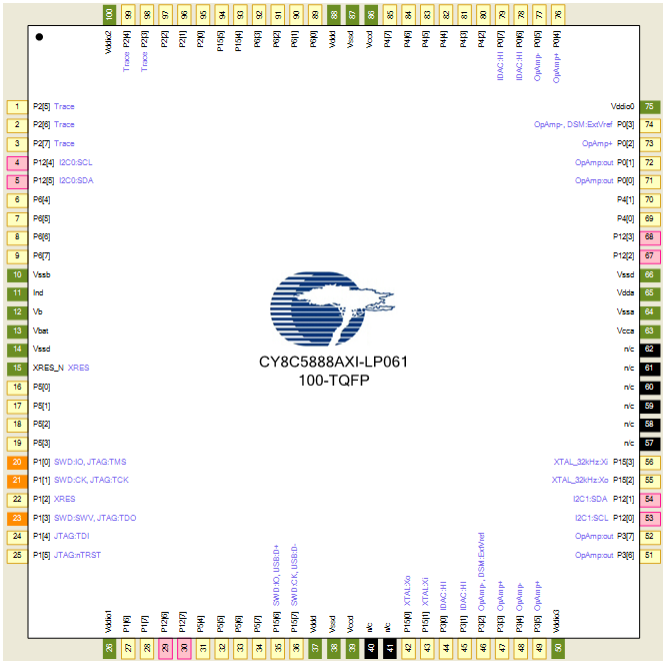
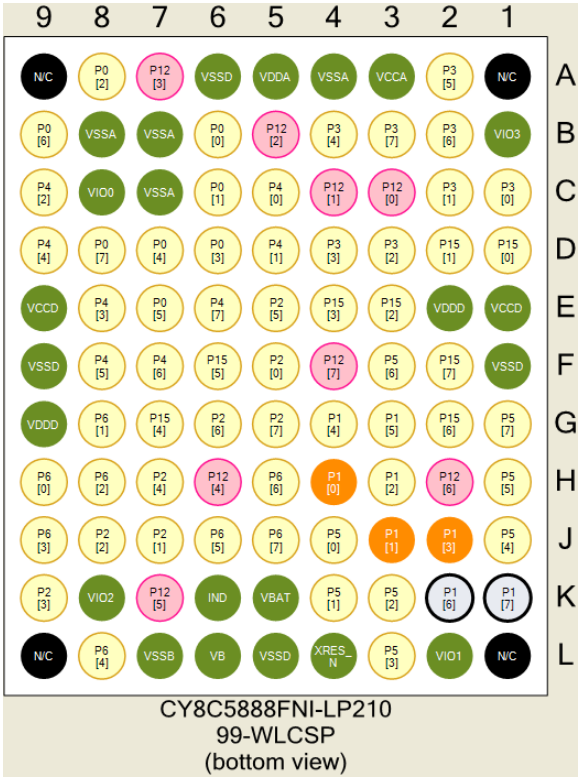


Figure 5. PSoC 5LP Pin Assignments for 100-TQFP and 99-CSP



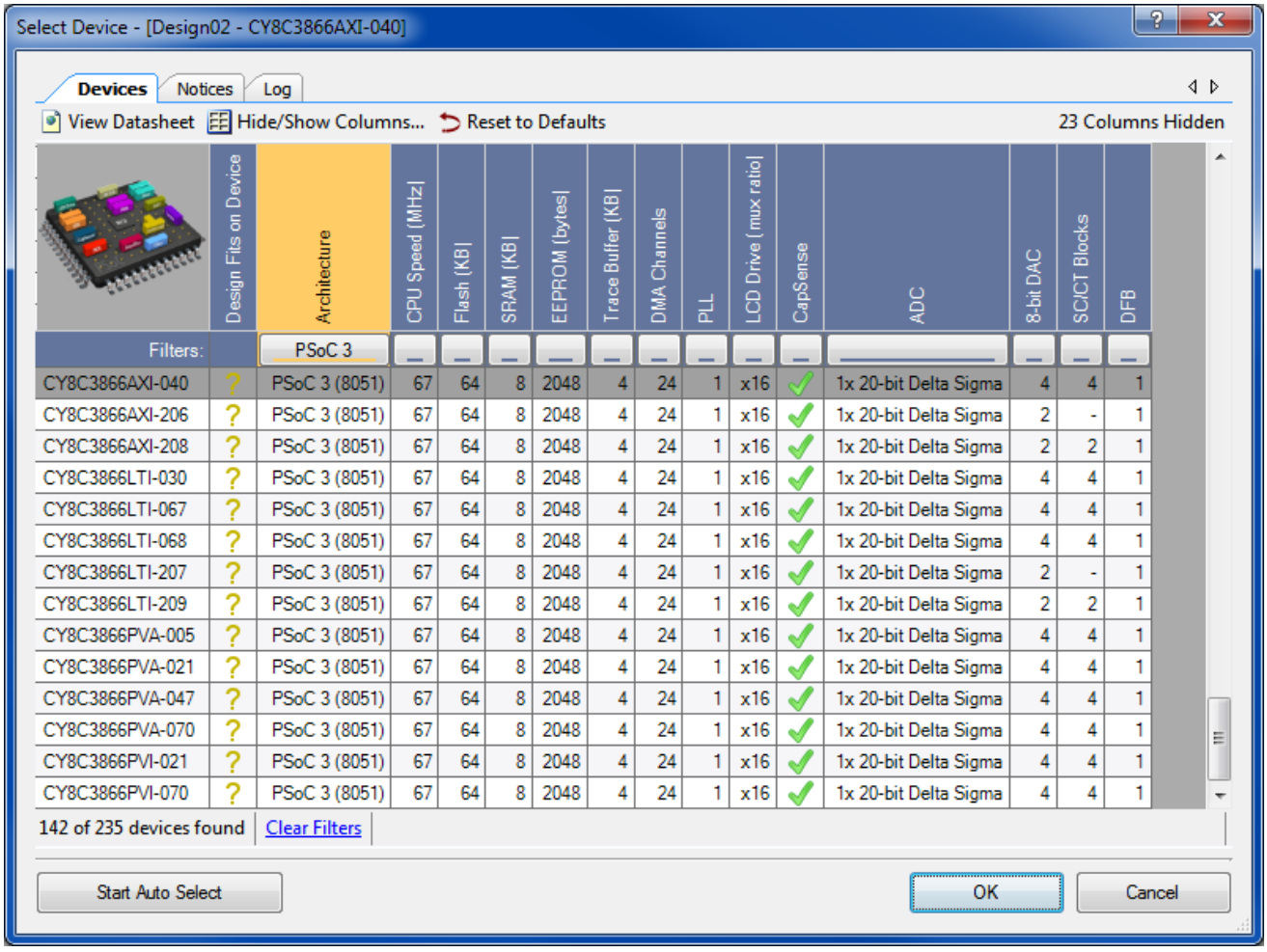


Note The 99-CSP assigns three pins to V_{SSA} that are N/C in the 100-TQFP.

Selecting CSP Devices

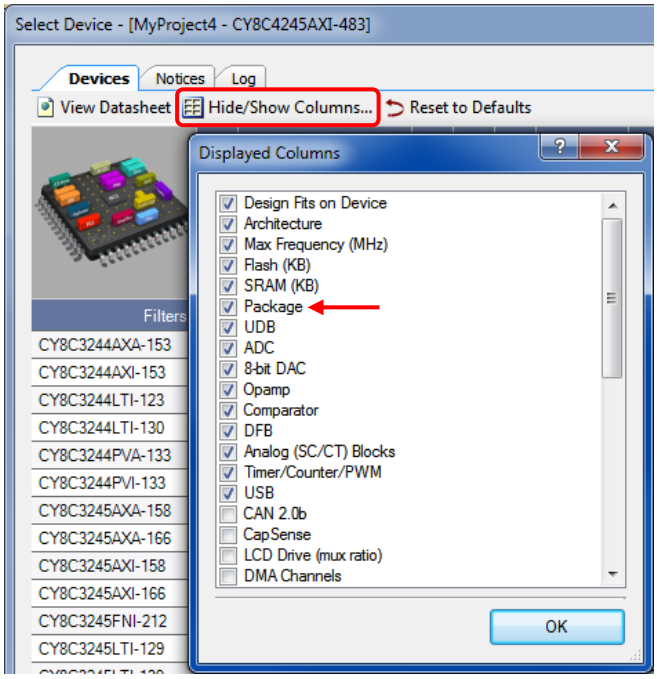
PSoC Creator 3.0 supports the PSoC 3 and PSoC 5LP devices currently available in CSPs. To select one of these devices for your project, open the **Select Device** dialog – click menu item Project > Device Selector. Figure 6 shows an example dialog.

Figure 6. PSoC Creator Device Selector



You can filter the selection menu for **Architecture** (PSoC 3, PSoC 4, or PSoC 5LP) and **Package**. If the Package column is not displayed, click **Hide/Show Columns** and check **Package**, as Figure 7 shows.

Figure 7. Show Packages in Device Selector



Click the bar in the **Package** column to open the filter box, and check only the packages that you want, as Figure 8 shows.

Figure 8. Package Filter Menu in Device Selector

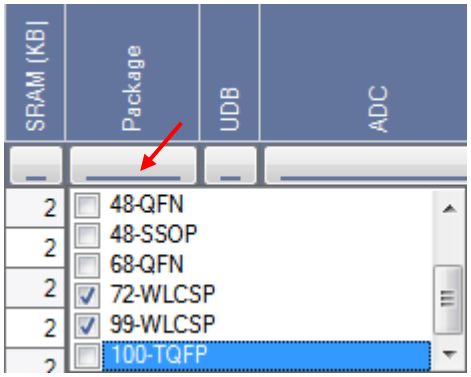
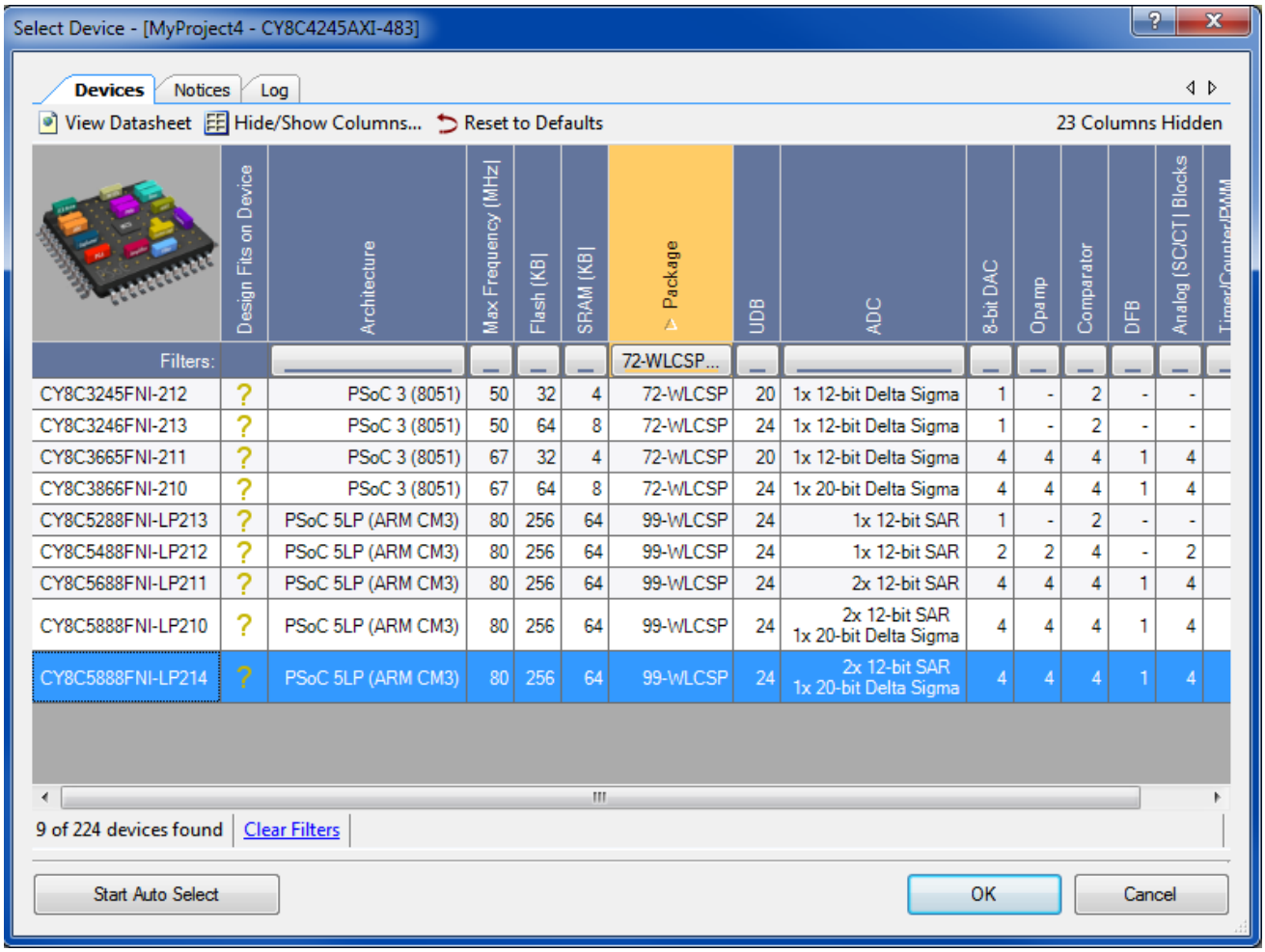


Figure 9 shows the Device Selector filtered for just the PSoC devices with CSPs. Note that:

- All CSP devices have “FN” in the part number.
- All PSoC 5LP CSP parts are 80 MHz max frequency; see a PSoC 5LP device datasheet for details.

Figure 9. Device Selector Filtered for All CSPs



After you select your device, do the rest of your design the same as for any other device or package.

Factory-Installed Bootloader

Programming a CSP device can be difficult for two reasons:

1. Off-board programming carries a risk of mechanical handling damage in programmer sockets. The programmer manufacturer must carefully design the socket to prevent mechanical overstress (MOS). The process and equipment for placing and removing the device in the socket must be optimized.

   Note Cypress is not responsible for damages incurred from off-board programming due to improper device handling.

2. CSP devices can be programmed using SWD or JTAG in the same manner as any other PSoC 3 or PSoC 5LP part. However, the standard connector for doing so (for example to a Cypress CY8CKIT-002 MiniProg3) may not be compatible with a very small PCB or an FPC.

   As an alternative, you can use a custom connector or test points for SWD or JTAG programming.

For these reasons, an I^2C bootloader is factory-installed in the flash memory of all PSoC 3 and PSoC 5LP devices in CSPs. This bootloader allows you to program the PSoC using another MCU on your board as a host and I^2C master. Or, you can run the I^2C lines to a smaller (3-pin) connector or test points for an off-board host; see Install the Bootloadable on page 15.

For more information on I^2C bootloaders for PSoC, see AN60317, I^2C Bootloader or AN73854, Introduction to Bootloaders.

Note It is also possible to directly program the PSoC through its JTAG or SWD port from another MCU on your PCB; this technique is called **host-sourced serial programming (HSSP)**. For more information, see AN73054, PSoC Programming Using an External Microcontroller (HSSP). Doing HSSP programming overwrites the factory-installed bootloader.

CSP Bootloader Specifications

Following are detailed specifications for the PSoC Creator I^2C bootloader project that is factory installed in PSoC 3 and PSoC 5LP CSP devices:

- IDE: PSoC Creator 3.0
- Compilers:
  - PSoC 3: Keil 9.51 for 8051
  - PSoC 5LP: gcc 4.7.3 for Cortex-M3
  - PSoC Creator default optimization levels
- Bootloader Configuration (for more information, see the Bootloader Component datasheet):
  - Component version 1.20
  - Single application (that is, not multi app)
  - Wait 2 seconds for a bootload command
  - Other options are Bootloader Component default
  - Occupies ~9 KB of flash starting at address 0
- Communication Component: I2C (for more information, see the I^2C Component datasheet)
  - Uses the PSoC fixed function I^2C block
  - I^2C slave mode
  - I^2C slave address 4
  - Data rate 100 kbps
  - SCLK available at P1[6]
  - SDAT available at P1[7]
  - External pull-up resistors required
- The SRAM, I^2C block, and P1[7:6] are all reusable by the bootloadable application.

How to Use the Bootloader

To use the factory-installed bootloader, do the following:

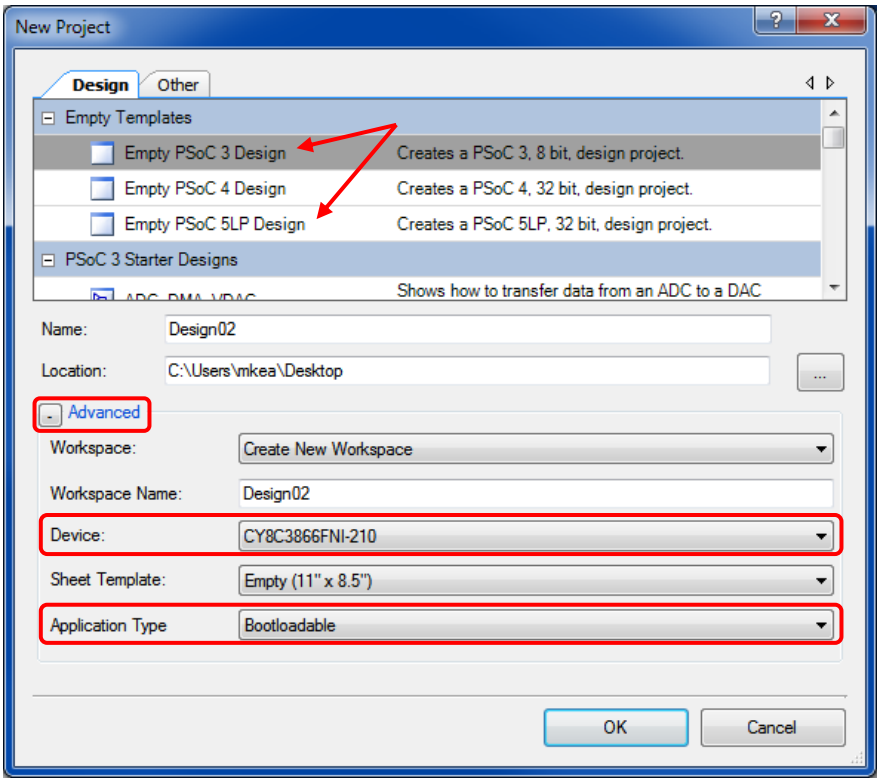
1. Create your PSoC Creator design as a **Bootloadable** project.
2. Link that design to the *.hex* and *.elf* files for the bootloader for the device that you intend to use.
3. Complete the project design hardware and firmware as is normally done, and build the project.
4. Install the resultant *.cyacd* file in your system's I^2C bootloader host.

The following sections provide details on steps 1, 2, and 4.

Create a Bootloadable Project

A PSoC Creator bootloadable project is the same as a **Normal** project except that it is relocated in flash memory to be above the bootloader (which starts at address 0). Create a bootloadable project, as Figure 10 shows.

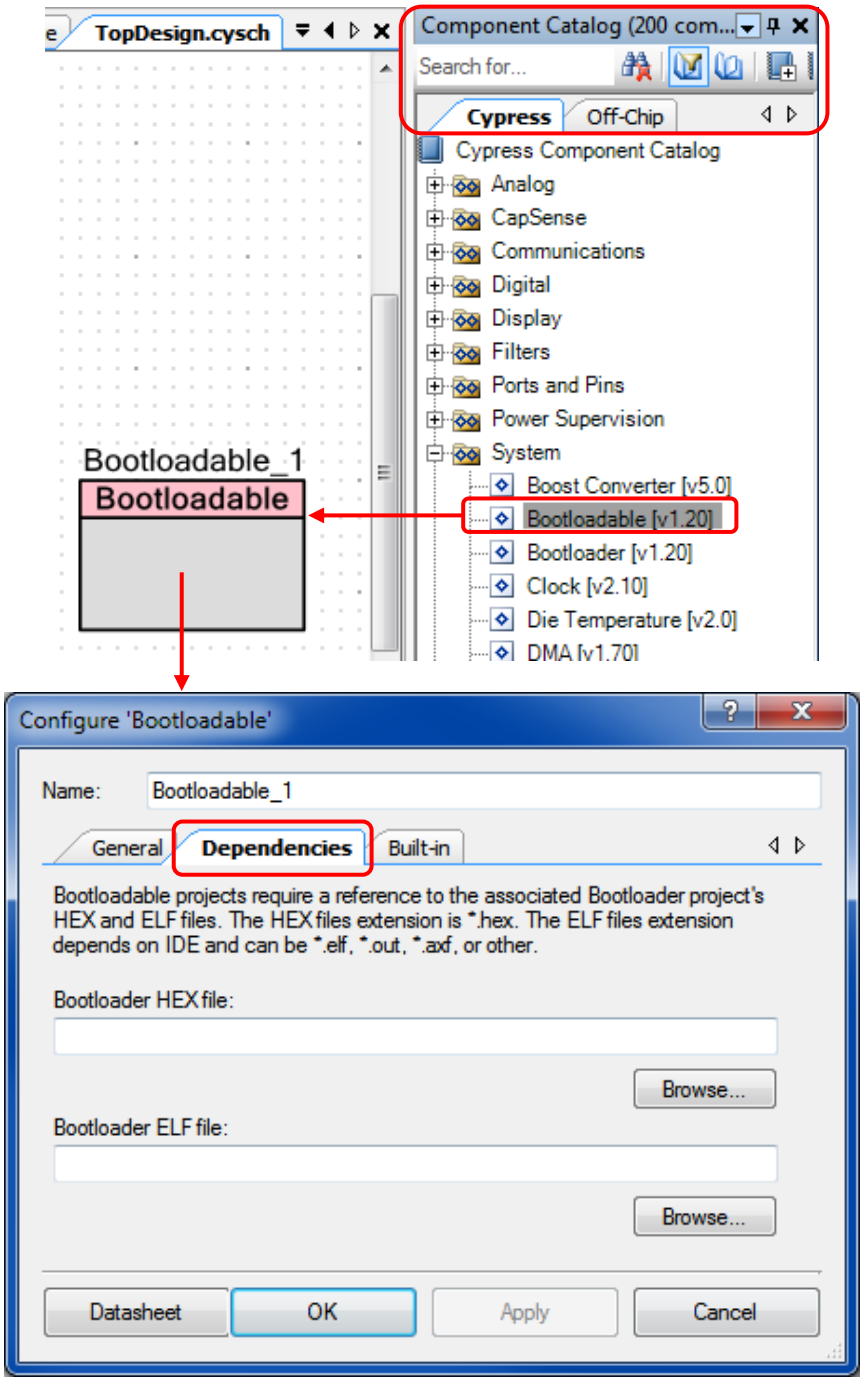
Figure 10. Create a Bootloadable Project



1. Select your design template (PSoC 3 or PSoC 5LP).
2. Open the **Advanced** options in the dialog.
3. Select the CSP device that you intend to use. If needed, click **<Launch Device Selector…>** in the pull-down menu, and use the package filter as Figure 9 on page 8 shows.
4. Select Application Type**Bootloadable**.

When the project is created, add a Bootloadable Component to the design schematic, as Figure 11 shows. Double-click the Component to open its configuration dialog, and click the **Dependencies** tab.

Figure 11. Add a Bootloadable Component



Link to Bootloader

To link your bootloadable project to a bootloader, you must tell the Bootloadable Component where the bootloader's *.hex* and *.elf* files are. This allows PSoC Creator to relocate your project to the correct starting address in flash, i.e., just above the bootloader.

Normally, the *.hex* and *.elf* files are in the folder for a bootloader project that you create, however in this case the bootloader project has already been created for you. Its *.hex* and *.elf* files are available on the Cypress web site, at:

www.cypress.com/go/PSoC3datasheet

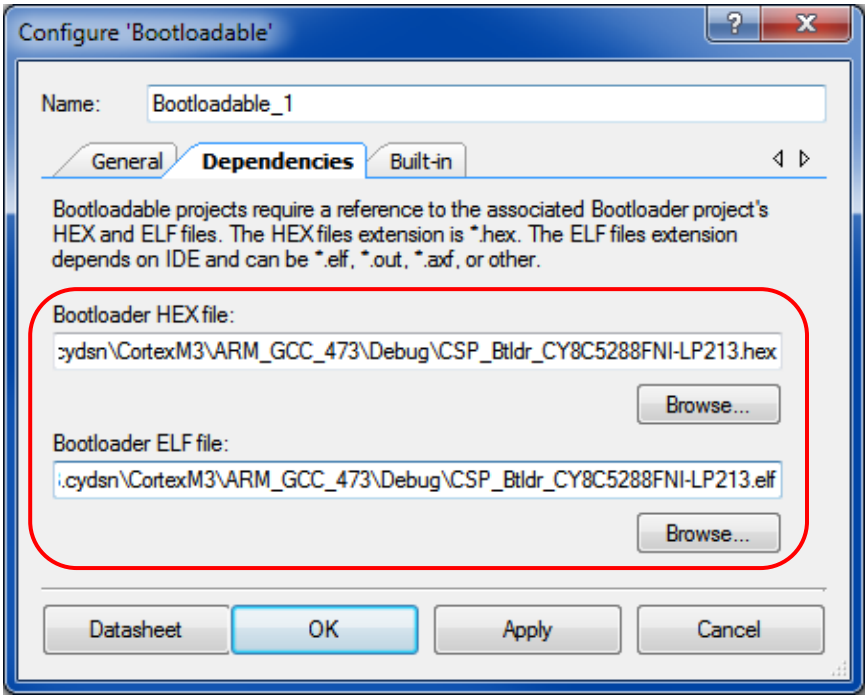
www.cypress.com/go/PSoC5LPdatasheet

for PSoC 3 and PSoC 5LP, respectively. (The PSoC Creator bootloader project is also available on the same web pages.)

Download the *.hex* and *.elf* files for the device that you intend to use, and save them in a convenient local folder. Then, in the Bootloadable Component configuration dialog, click the **Browse** button for **Bootloader HEX file**, as Figure 12 shows.

Navigate to the appropriate *.hex* file and select it. If the corresponding *.elf* file is in the same folder, it is also selected, automatically.

Figure 12. Link to Bootloader File



Click **OK** when the files are correctly identified. You can then complete the remainder of your design and build it, as is normally done.

Update Design-Wide Resource (DWR) Settings

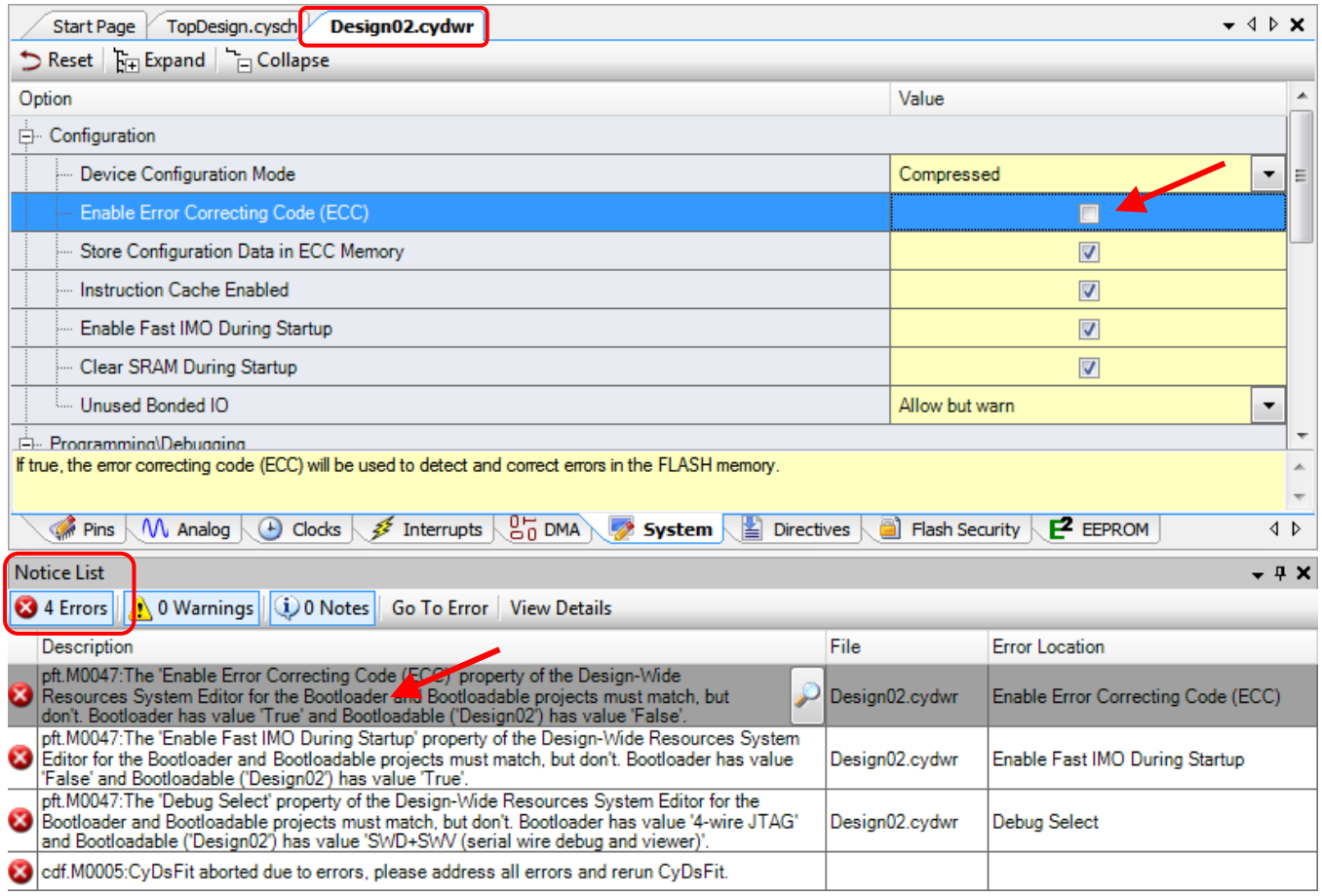
The factory-installed bootloader projects have non-volatile latch (NVL) settings that match the device datasheets. However, when you create a new PSoC Creator project, some of the project's default NVL settings do not match the device datasheet, as Table 1 shows:

Table 1. PSoC 3/5LP NVL Default Settings

NVL Parameter	PSoC 3/5LP Datasheet Default	PSoC Creator Project Default
Flash Error Correction Code (ECC)	Enabled	Disabled
Internal main oscillator (IMO) speed	Slow (12 MHz)	Fast (48 MHz)
Debug Pins Select	4-wire JTAG	SWD+SWV

The NVL settings for a bootloadable project must match those of the linked bootloader project. In this case, if you attempt to build a PSoC Creator bootloadable project, you will get 3 error notices. Double-click a notice to open the project's design-wide resources (DWR) file, and change the setting to match the bootloader, as Figure 13 shows:

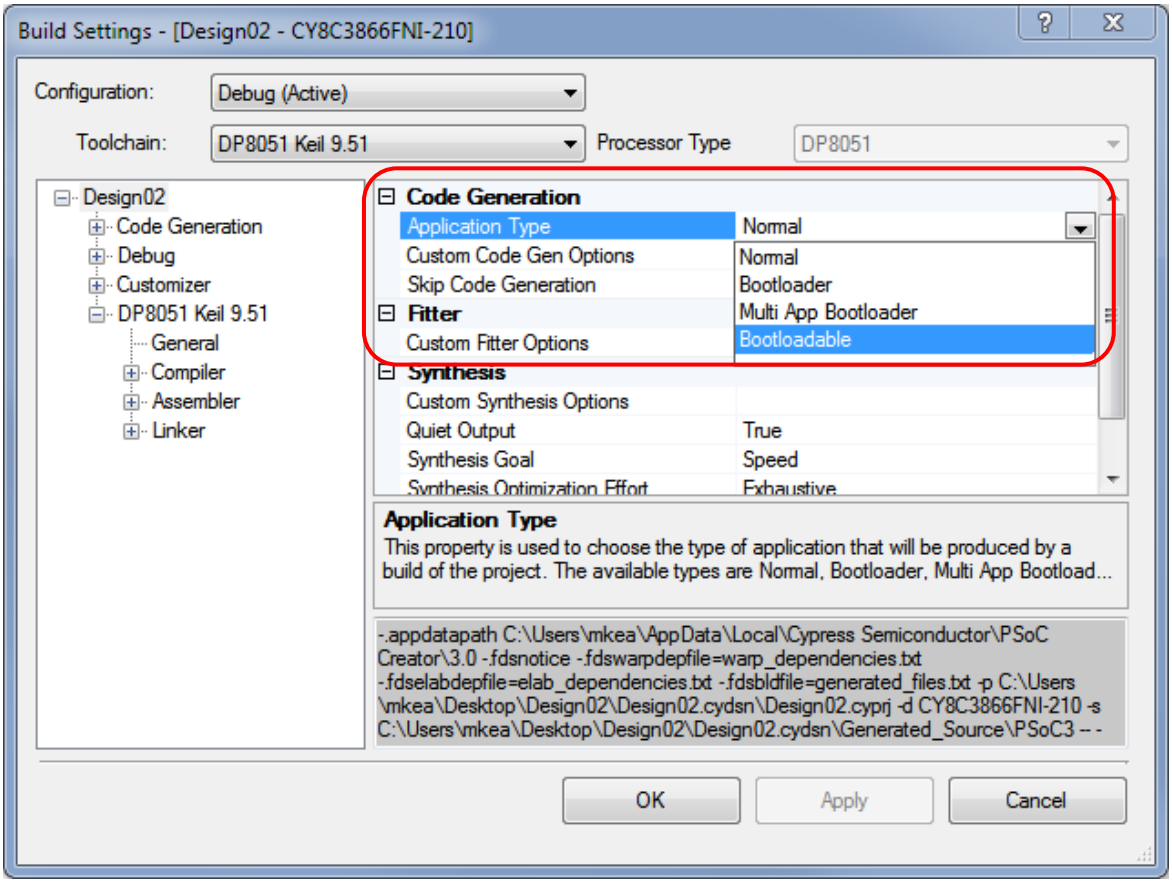
Figure 13. Adjust PSoC Creator Default DWR Settings



When all 3 errors are corrected, you should be able to successfully rebuild the bootloadable project.

Note For debugging, one best practice recommendation is to first create and build your project as Application Type **Normal** (Figure 10), and debug it using an alternate device or hardware, for example, a Cypress PSoC kit. Then, convert the project to type **Bootloadable**, as Figure 14 shows (choose **Project** > **Build Settings**). Change the device if necessary (Figure 9), add and configure the Bootloadable Component (Figure 11), and rebuild.

Figure 14. Changing the Application Type



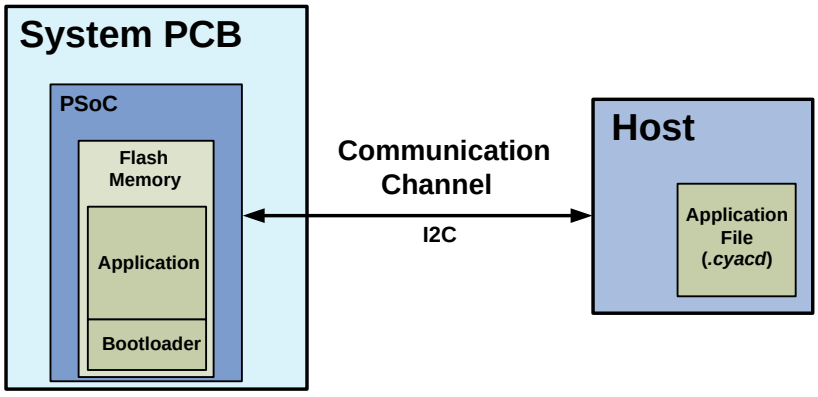
Install the Bootloadable

When you build a bootloadable project, PSoC Creator creates two output files: *.hex* and *.cyacd*.

- The *.hex* file contains code for both the bootloadable and the associated bootloader, and is typically used for factory programming.
- The *.cyacd* file is intended for field use, where it is downloaded to a target device in a bootload operation.

How the *.cyacd* file is installed and used depends on how your PSoC target device is used in your system. Figure 15 shows one example, with an external host. The host is the I²C master.

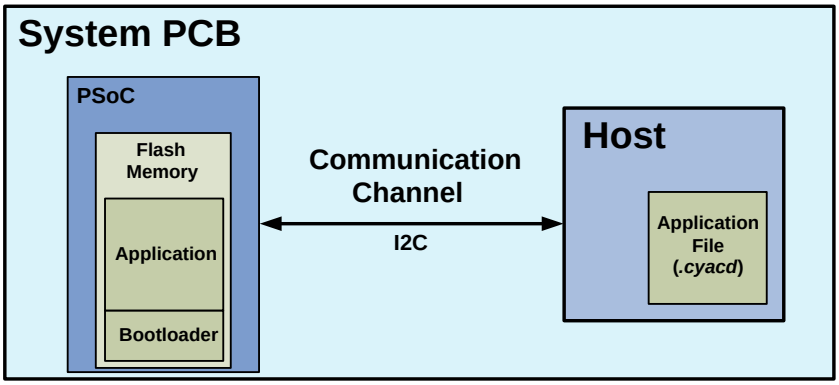
Figure 15. Bootload System with External Host



This system requires that a connector (three contacts minimum) or test points be mounted on the PCB to enable a connection to the host. There is no industry standard for I²C connectors; you can select the connector hardware to minimize the cost or PCB footprint.

Figure 16 shows another example in which a second MCU is mounted on the same board as the PSoC.

Figure 16. Bootload System with Onboard Host



The second MCU acts as the host and I²C master. Additionally, the on-board I²C bus can be used to communicate between other devices on the PCB for other tasks besides bootloading.

Note The system shown in Figure 16 assumes that a connector exists to allow communication with the host and downloading of the *.cyacd* file to the host.

An example of the system shown in Figure 16 is described in AN60317, I²C Bootloader. In this example, the host is a PSoC 5LP and the target is a PSoC 3. Two *.cyacd* files from two different projects are embedded in the PSoC 5LP firmware as arrays of strings. Depending on user input, one of the files is bootloaded to the target PSoC 3, which changes the PSoC 3 functions.

Summary

This application note presented an overview of guidelines and best practices for CSPs for PSoC 3 and PSoC 5LP. It also explained how to use the I^2C bootloader that is factory installed in these devices. For more information on both of these topics, see Related Resources.

Related Resources

AN69061 – Design, Manufacturing, and Handling Guidelines for Cypress Wafer-Level Chip Scale Packages (WLCSP)

AN34359 – PCB Layout Guidelines for West Bridge™ Generation A Peripheral Controllers in Wafer Level Chip Scale Package Although this application note is for non-PSoC Cypress products, which have different size packages, it provides general helpful PCB layout tips.

Amkor also provides several application notes, as well as a wealth of other information, for WLCSP.

Other relevant documents:

AN61290 – PSoC 3 and PSoC 5LP Hardware Design Considerations

AN73854 – PSoC 3 and PSoC 5LP – Introduction to Bootloaders

AN60317 – PSoC 3 and PSoC 5LP I^2C Bootloader

AN73054 – PSoC 3 and PSoC 5LP Programming Using an External Microcontroller (HSSP)

About the Author

Name:	Mark Ainsworth
Title:	Applications Engineer Principal
Background:	Mark Ainsworth has a BS in Computer Engineering from Syracuse University and an MSEE from the University of Washington, as well as many years of experience designing and building embedded systems.

Appendix A –Schematics and PCB Footprints

Cypress offers an Altium designer library for PSoC 3 and PSoC 5LP. The following CSP schematics and footprints are included:

Figure 17. 72-WLCSP Schematics, for Pin Group and PSoC Function

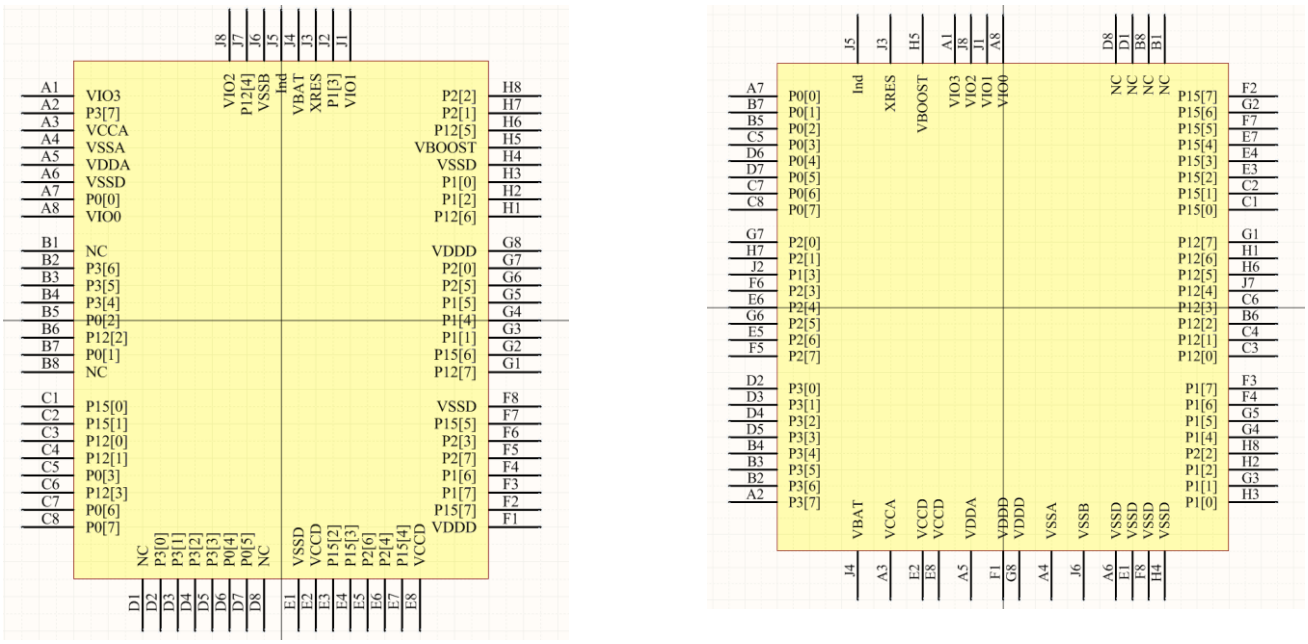


Figure 18. 72-WLCSP Footprint

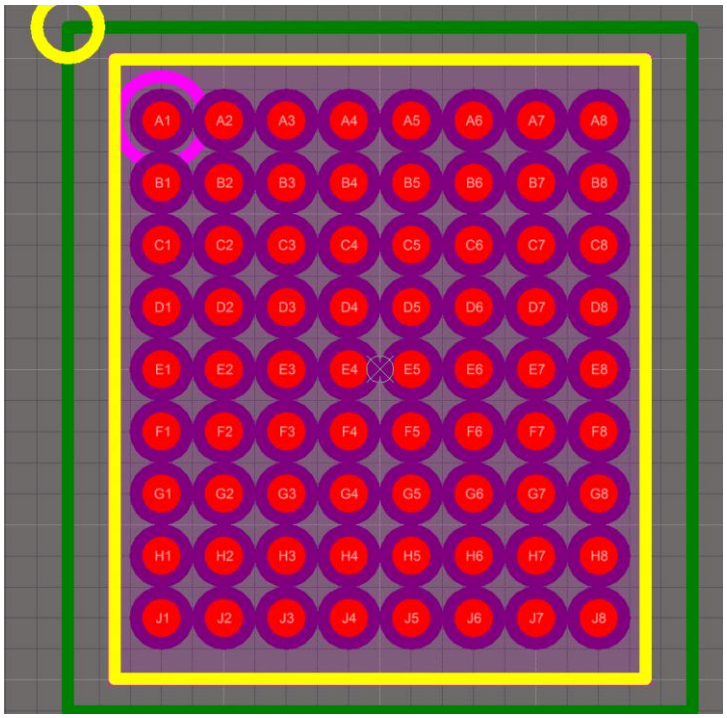


Figure 19. 99-WLCSP Schematics, for Pin Group and PSoC Function

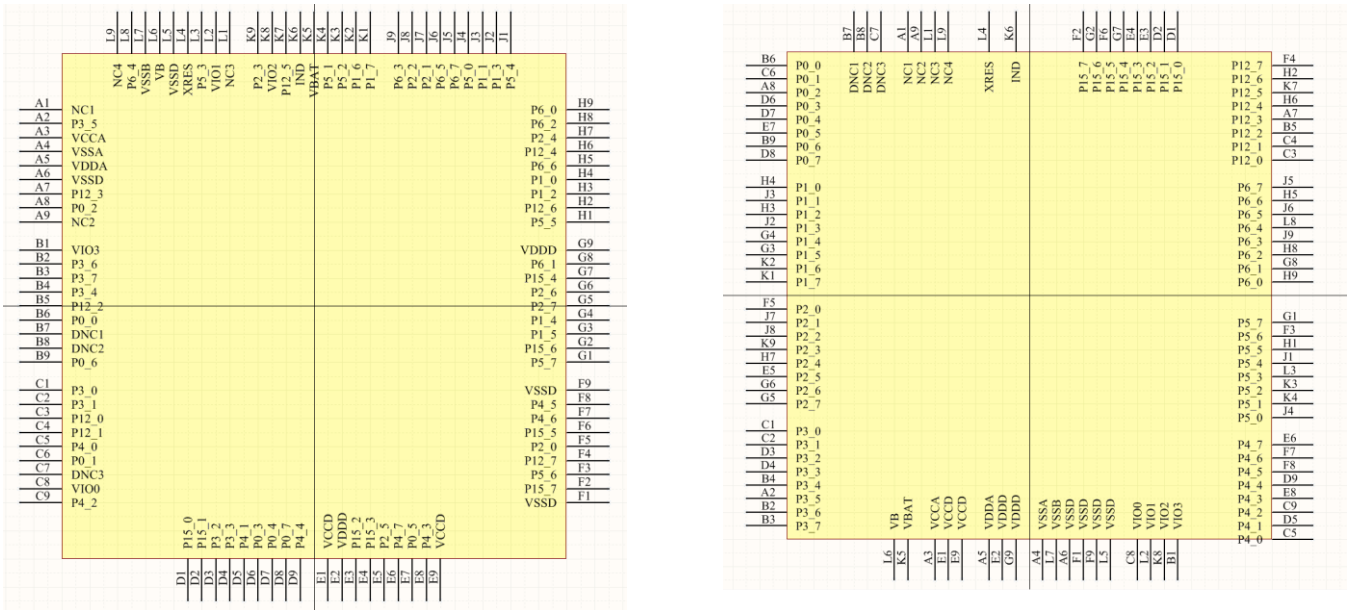
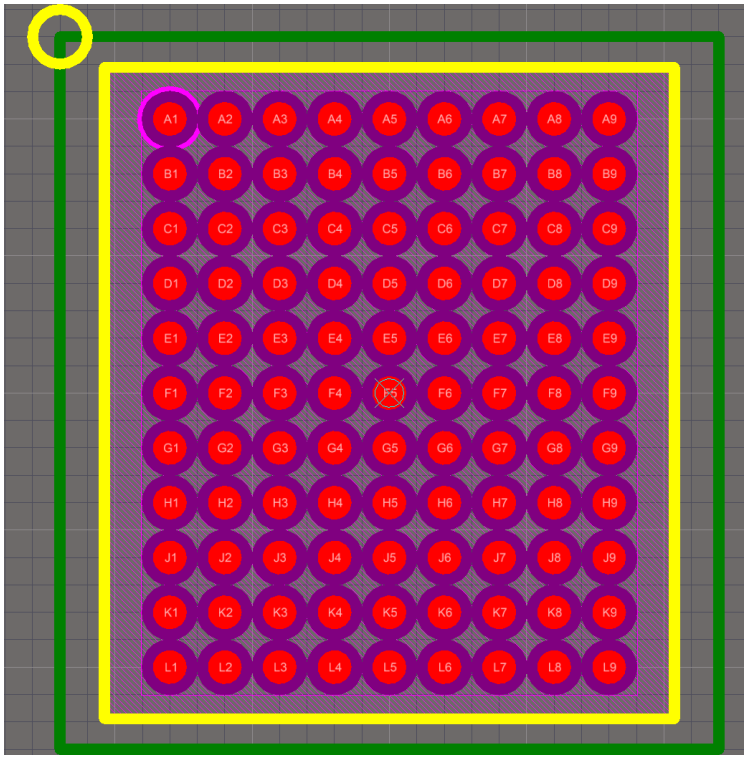


Figure 20. 99-WLCSP Footprint



Document History

Document Title:AN89611 - PSoC® 3 and PSoC 5LP – Getting Started with Chip Scale Packages (CSP)

Document Number:001-89611

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	4220233	MKEA	12/13/2013	New application note
*A	4256695	MKEA	01/21/2014	Updated link to web page
*B	4292957	MKEA	02/26/2014	Added Appendix A for PCB and schematic images. Added link to Altium library.
*C	4507268	MKEA	09/18/2014	Added section Update Design-Wide Resource (DWR) Settings
*D	4625892	MKEA	01/23/2015	Changed section Related Application Notes to Related Resources. Added PCB Layout section. Added references to AN34359.
*E	5702106	AESATMP9	04/19/2017	Updated logo and copyright.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

©Cypress Semiconductor Corporation, 2013-2017. 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.