设计指南：TIDM-1022
# 使用 *C2000™ MCU* 的谷底开关升压 *PFC* 参考设计

TEXAS INSTRUMENTS

说明

该参考设计说明了一种数字控制方法，可显著提高升压功率因数校正 (PFC) 转换器性能，例如难以满足效率和 THD 标准的轻负载条件下的效率和总谐波失真 (THD)。这是通过集成数字控制功能 (Type-4 PWM)（由 Piccolo™F280049 微控制器 (MCU) 提供）实现的。该设计支持不同负载和瞬时输入电压条件下的切相、谷底开关、谷底跳跃和零电压开关 (ZVS)。该 TI 设计非常适用于多种 应用，例如：电动汽车 (EV) 充电、服务器、电信整流器和工业电源。可供此参考设计使用的软件可缩短上市时间。

资源

TIDM-1022	设计文件夹
TMS320F280049	产品文件夹
TMDSCNCD280049C	工具文件夹
TMDSADAP180TO100	工具文件夹
C2000WARE-DIGITALPOWER-SDK	工具文件夹
TIDM-2PHILPFC	设计文件夹

Search Our E2E™ support forums
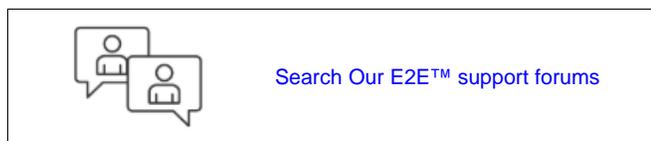
特性

- 交错式 750W 两相升压 PFC 级
- 全范围参数：95–260Vrms、47–63Hz、750W
- 可编程输出电压，标称值为 380V 直流输出
- 正常条件（负载 > 10%）下的开关频率为 200kHz
- 轻负载 (<10%) 下可实现 150–330kHz 的可变脉宽调制 (PWM) 开关
- 低 THD：5% 负载（低压线路）下为 6%；5% 负载（高压线路）下为 7%
- 5% 负载下具有大于 92% 的高效率
- C2000™ 提供 powerSUITE™支持，以使设计轻松适应用户要求
- 用于快速测量开环增益的软件频率响应分析器 (SFRA)
- 使用 TI Piccolo™ F280049C 控制器实现全面数字控制
- 针对输出过流和过压情况提供保护
- 可编程谷底开关和谷底跳跃

应用

- 电动汽车 (EV) 车载充电器
- 服务器和网络电源
- 电信整流器
- 工业电源
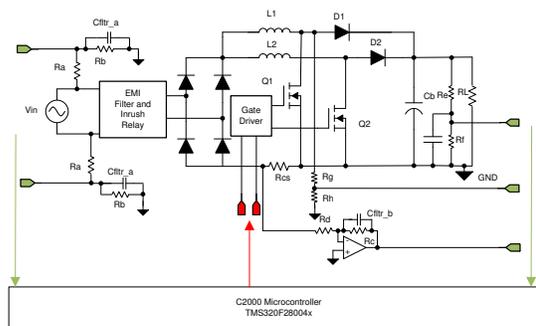
⚠ 该 TI 参考设计末尾的重要声明表述了授权使用、知识产权问题和其他重要的免责声明和信息。

1 System Description

As the EV charging market grows rapidly, increasing charging efficiency becomes an important consideration for researchers, as it can be applied to both trickle and fast charging. Compared to the heavy load performance, light load efficiency and THD are more difficult to improve.

Valley switching is a soft-switching technique that improves system efficiency of AC-DC and DC-DC converters. Significant efficiency improvements are achieved with valley switching mainly under low-load conditions where efficiency standards are difficult to meet. This document presents the implementation details of controlling a boost power stage with valley switching using F28004x MCU from TI's C2000 family of MCUs. A modified 2-phase interleaved power factor correction (ILPFC) kit is used to validate the software and system operation. A digitally controlled PFC converter for such applications is shown in 图 1.

1.1 Key System Specifications

表 1 lists the power specifications of the Valley Switching Boost PFC reference design.

表 1. Key System Specifications [Light Load Conditions (Load < 10%, 750 W)]

PARAMETER	SPECIFICATION	
INPUT CONDITIONS		
Input voltage (V_{INAC})	110 V AC (**MIN**) to 230 V AC (**MAX**)[1]	
Input current (I_{IN})	<1 A (light load)	
Input frequency (f_{LINE})	47–63 Hz	
OUTPUT CONDITIONS		
Output voltage (V_{OUT})	380 V DC	
Output current (I_{OUT})	<0.2 A (light load)	
Output power	<75 W (single phase)	
SYSTEM CHARACTERISTICS (Test Conditions: 5% load)		
	High Line (V_{INAC} = 220 V)	**Low Line (V_{INAC} = 120 V)**
Efficiency (η)	92%	92%
PWM switching frequency	150–330 kHz	
Current THD	7%	6%
Power factor	0.992	0.995

[1] The build examples in this document were based on these specifications, any deviance outside these specified ranges, may require additional modifications to the software.

2 System Overview

2.1 Block Diagram

图 1. Digital Interleaved PFC Converter
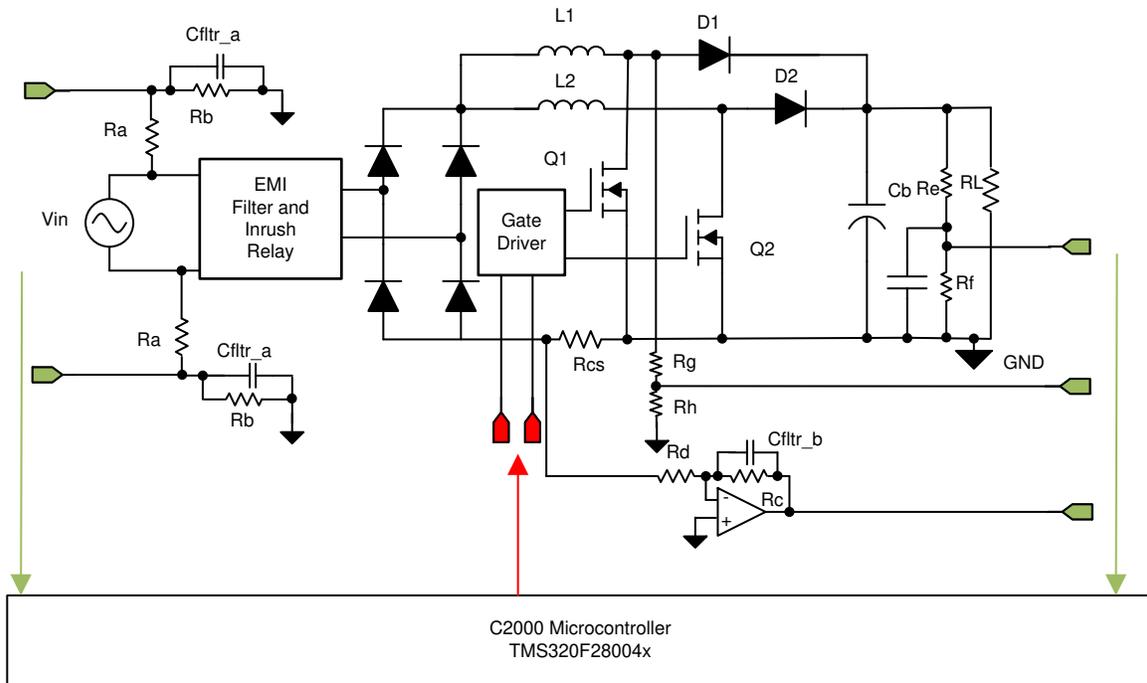
2.2 Design Considerations

图 1 illustrates a C2000 controller based interleaved PFC converter control system. The input AC voltage is applied to the PFC converter through the input EMI filter, followed by an inrush current limit and a bridge rectifier. The PFC stage consists of two boost converters, each operating at 200 kHz and phase shifted by 180 degrees under normal conditions (load > 10%). Once the load drops below 10%, the phase shedding and valley switching are enabled and the switching frequency is variable based on the captured valley point.

Inductor L1, MOSFET switch Q1 and diode D1 together form one of the boost stages while, L2, Q2, and D2 form the other boost stage. A capacitor Cb at the boost converter output acts as an energy reservoir and this, in conjunction with closed loop PFC control, provides a regulated DC voltage to the PFC load R_L.

图 2 indicates all the interface signals needed for full control of this PFC converter using a C2000 MCU. The MCU controls the hardware using five feedback signals and one or two PWM outputs of a single PWM module. The signals that are sensed and fed back to the MCU via the Analog-to-Digital Converter (ADC) include: the line and neutral input voltages (VL_sen and VN_sen), the PFC input current (PFC_Isen), the DC bus output voltage (Vbus_sen), and the drain-to-source voltage of MOSFET(Vds_sen). These sensed signals are used to implement the voltage and current control loops for the ILPFC converter with valley switching.

For valley switching, channel A of the EPWM1 Module maintains the output and channel B is disabled. In other words, phase shedding is enabled under light load conditions (<10% load). Also, valley skipping can be implemented using EPWM1. EPWM2 and EPWM8, respectively, are utilized to trigger 100- and 10-kHz Interrupt Service Routine (ISR) which is same as the normal ILPFC condition. EPWM3 and EPWM4 are used for oversampling to achieve better performance. EPWM7 is used to implement part of the equation-based ZVS algorithm, for example, the blanking window.

For more information on implementing valley switching and using the EPWM modules, see the Valley Switching Application Report.

图 **2. F28004x Peripheral Usage**
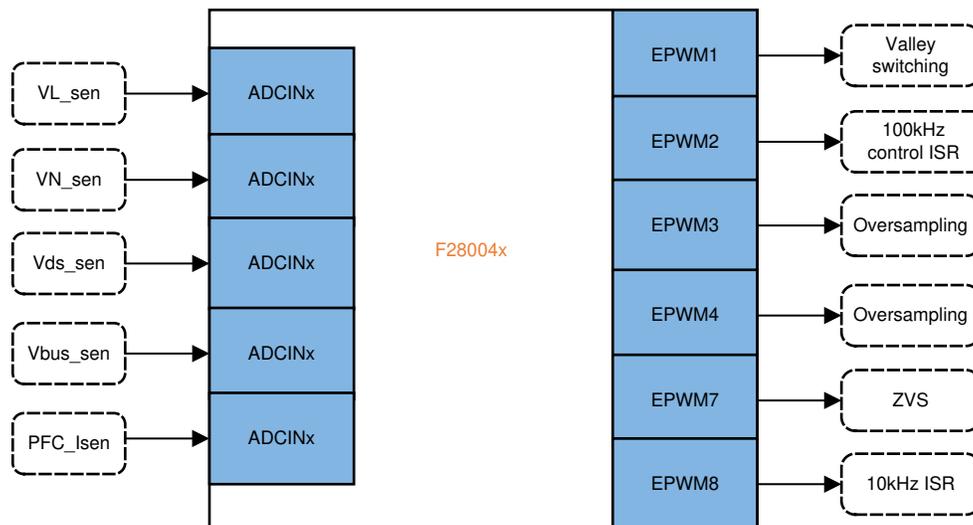
2.3 Highlighted Products

2.3.1 C2000™ MCU F28004x

C2000 MCUs are part of an optimized MCU family for real-time control applications. The fast and high-quality ADCs enable accurate measurement of the current and voltage signals, and an integrated Comparator SubSystem (CMPSS) provides protection for overcurrent and overvoltage without use of any external devices. The optimized CPU core enables fast execution of control loops. Trigonometric operations are accelerated using the on-chip Trigonometric Math Unit (TMU). The solution also provides an option, when using F28004x and F2837x devices, to use the Control Law Accelerator (CLA) coprocessor. The CLA can be used to alleviate CPU burden and enable faster-running loops or enable more functions on the C2000 MCU.

2.4 System Design Theory

2.4.1 Firmware Flow

The valley switching firmware flow diagram is shown in 图 3. The firmware project makes use of the C-background, C-ISR, and CLA-ISR framework. The project uses C-background code as the main supporting program for the application, which is responsible for all system management tasks, decision making, intelligence and host interaction. The C-ISR code consists of two components (Slow Control ISR and Fast Control ISR), which are executed inside time-critical ISRs and run all the critical control code.

This code includes ADC reading and PFC control calculations. The Slow Control ISR is executed at a fixed rate of 10 kHz using the PWM triggered ISR. The Fast Control ISR is triggered by PWM2 and is executed at a rate of 100 kHz. The CLA based PWM ISR executes at a variable rate determined by the PWM switching frequency and is triggered every time Vds reaches the valley point. The key valley switching control calculations and PWM updates are in the CLA based high-speed ISR.

图 3. Valley Switching Firmware Flow Diagram
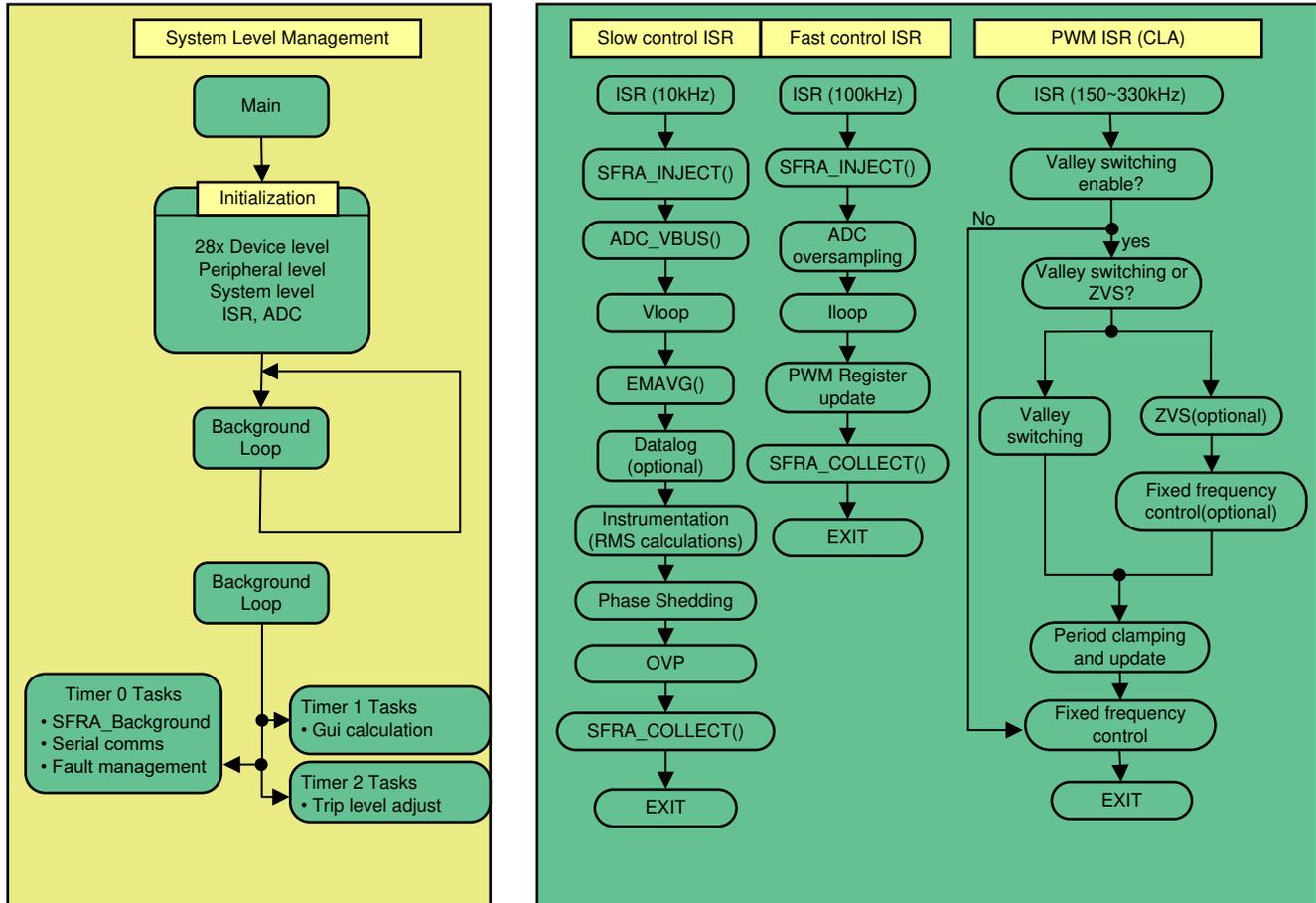
2.4.1.1 Background Loop (C-background)

A task state-machine has been implemented as part of the background code. Tasks are arranged into three groups (A1, A2, A3…, B1, B2, B3…, and C1, C2, C3...). The tasks within each of the three groups are executed according to three assigned CPU timers, which are configured with periods of 1 ms , 5 ms, and 10 ms, respectively. Within each group, each task is run in a round-robin manner. For example, group B is dedicated to a CPU timer that is configured with a period of 5 ms. If there are two tasks in group B (B1 and B2), then, the two group B tasks execute once every 10 ms.

2.4.1.2   Slow Control ISR (a Component of C-ISR)

The Slow Control ISR is responsible for running the voltage loop. The voltage loop uses a filtered version of the bus voltage feedback. A slew limit algorithm and software overvoltage protection algorithm are also implemented inside the Slow Control ISR. Part of the valley switching algorithm is executed in this ISR as well. The phase shedding algorithm under light load conditions (< 10%) is also implemented together with the switch to "turn on" and "turn off" the valley switching. Once the MCU makes the decision to disable the second phase, the valley switching is enabled at the same time.

2.4.1.3   Fast Control ISR (a Component of C-ISR)

The Fast Control ISR is responsible for current sensing, PFC calculation, SFRA, current loop, voltage loop (half speed of current loop), and duty cycle calculation when the valley switching is disabled. This is the same compared with the traditional Interleaved PFC.

2.4.1.4   PWM ISR (CLA-ISR)

This variable frequency ISR mainly contains the algorithm of valley switching and ZVS for different load and input conditions. This software solution makes use of the global reload feature and other valley switching-related features available on C2000 devices, like the Piccolo F280049C, with type-4 PWM modules. The global load feature enables the synchronization for different PWM modules when the valley point is captured. The PWM ISR is moved into CLA considering the CPU usage based on the driverlib when the frequency of PWM ISR can go up to 300 kHz under some circumstances.

Based on the instantaneous input voltage conditions, these are the algorithms implemented in the PWM ISR (CLA-ISR):

*   When instantaneous input voltage is greater than half of the output voltage, ZVS can be achieved by using equation based methodology.

*   When instantaneous input voltage is less than half of the output voltage, the valley switching is realized by setting up valley switching related registers such valley capture, edge filter, etc.

*   When the instantaneous input voltage is close to zero, the fixed frequency is applied to help achieve higher performance of THD and efficiency.

2.4.1.5   powerSUITE™ Tools

The TI SFRA library is designed to enable frequency response analysis on digitally-controlled power converters using software alone. This feature enables performing frequency response analysis of the power converter with relative ease as no external connections or equipment is required. The optimized library can be used in high-frequency power conversion applications to identify the plant and the open-loop characteristics of a closed-loop power converter, which can be used to get stability information such as bandwidth, gain margin, and phase margin to evaluate the control loop performance. For more information, see the C2000™ SFRA Library and Compensation Designer User's Guide.

In addition to SFRA, this kit supports the use of another powerSUITE tool called the Compensation Designer. The Compensation Designer tool allows the design of different styles of compensators to achieve the desired closed loop performance, which can be done using the measured power stage or plant data from the SFRA Tool. The coefficients that must be programmed on the device are generated by the Compensation Designer and can be copied into the code directly. These tools help users evaluate the complete system, adapt it for their end application, and tune it for improved performance.

2.4.2 Incremental Builds

This project is divided into three incremental builds, which makes learning and getting familiar with the board and software easier. This approach is also good for debugging and testing boards. 表 2 shows the incremental build options. To select a particular build option, select the corresponding INCR_BUILD option in the main.syscfg. Once the build option is selected, compile the complete project by selecting the rebuild-all compiler option. 节 3.1.2.4 provides more details to run each of the build options.

表 **2. Incremental Build Options**

BUILD OPTION	DESCRIPTION
INCR_BUILD = 1.1	Open-loop check (check PWM drive circuit, ADC drive circuit, sensing circuit, protection mechanisms, valley switching disabled)
INCR_BUILD = 1.2	Open-loop check (valley switching enabled under DC input)
INCR_BUILD = 2	Open voltage loop, closed current loop check
INCR_BUILD = 3	Closed voltage loop, closed current loop check

3   Hardware, Software, Testing Requirements, and Test Results

3.1   Required Hardware and Software

3.1.1   Hardware for Valley Switching

The design hardware consists of a modified TIDM-2PHILPFC base board and a C2000 F280049C controlCARD along with a 180- to100-pin DIMM adapter to connect the two.

3.1.1.1   Base Board Modification (see 图 5 and 图 6)

This implementation uses a modified 2-phase Interleaved Power Factor Correction (ILPFC) Kit (TIDM-2PHILPFC) to validate this system. 图 4 is a top view of the base board. Other than C2000 MCU, MOSFET driver UCC27524 and operational amplifier OPA365 are used in the base board.

Only one phase of the boost configuration is used with valley switching enabled. 图 5 provides the relevant schematic sections. Compared to the 2-phase interleaved power factor correction (ILPFC) kit, an extra voltage divider is needed to achieve valley switching. In the current implementation an op amp (U3) can be used for good signal integrity in the Vds sensing circuit. However, this reference design can achieve very good results without using the op amp (U3). The Vds sensing circuit allows experimenting without the op amp (U3) by directly connecting the voltage divider (R20 through R22) output to the comparator input (pins 15 and 69). It is also worth noting that the voltage divider needs to be in the Printed Circuit Board (PCB) to achieve the best performance specifications (see 表 1) since the valley switching control requires a high-quality Vds sensing signal.

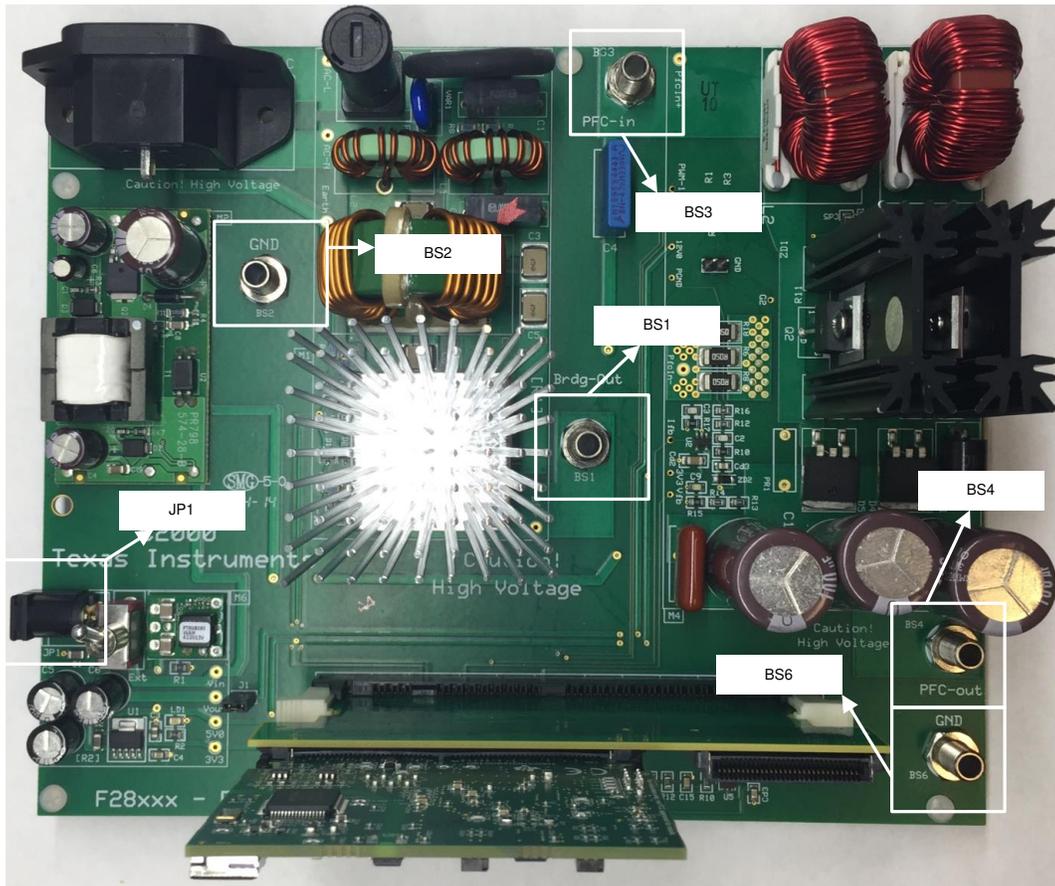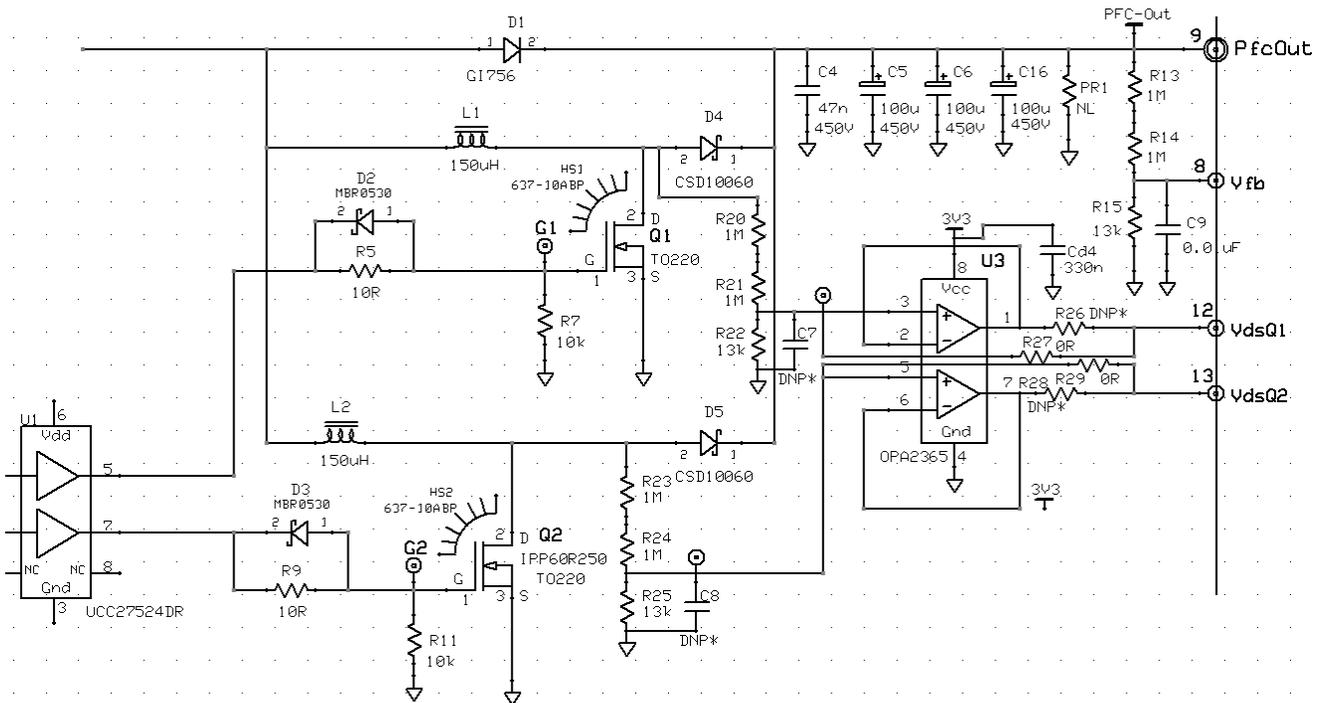图 **4. Top View of the Base Board**

图 5. Valley Switching Vds Sensing Schematic

In this reference design, the valley switching schematic controlCARD interface is shown in 图 6. The Vds feedback paths are added based on the original ILPFC kit. Users do not need to connect both VdsQ1 and VdsQ2 to the controlCARD. If the phase 1 is working when valley switching is enabled, then only VdsQ1 is needed to feed back to the controlCARD which means only one voltage divider is needed in the circuit.

图 6. Valley Switching Schematic controlCARD Interface
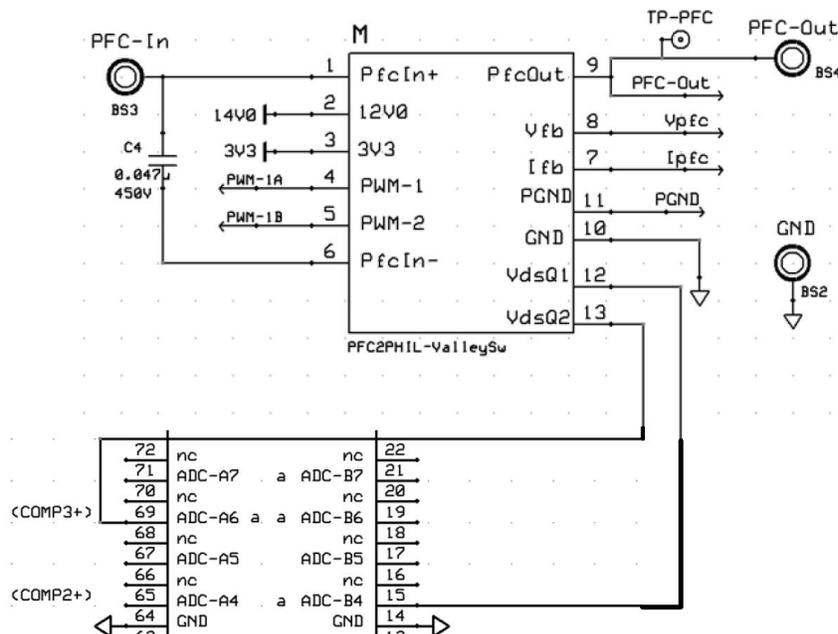
3.1.1.2   *F280049C controlCARD Modifications and Settings*

Certain modifications and settings on the device control card are required to control the base-board, communicate over the JTAG, and use the isolated UART port. The following steps are the required for running the valley switching based on the F280049C controlCARD. Users can also refer to the information sheet located inside the C2000Ware at <install_path>\c2000ware\boards\controlcards\TMDSCNCD280049C or alternatively, get the information sheet from the *Piccolo F280049C controlCARD Information Guide*.

1. Set S7: Turn off S7-4 and S7-5 (by moving the switches down) as displayed in 图 7 to disconnect the filter capacitor C37 and C38 to avoid Vds sensing signal distortion.

2. Remove the capacitor (C26) connected between the isolated grounds on the controlCARD, for the best performance of this reference design.

Connect the USB connector to the Piccolo F280049C controlCARD for emulation. Connect the 12-V bias supply output to JP1 and apply this bias voltage to the board by setting the switch (SW1) to position "Ext". By default, the F280049C controlCARD jumpers (see the *Piccolo F280049C controlCARD Information Guide*) are configured such that the device boots from Flash.

图 **7. F280049C controlCARD**
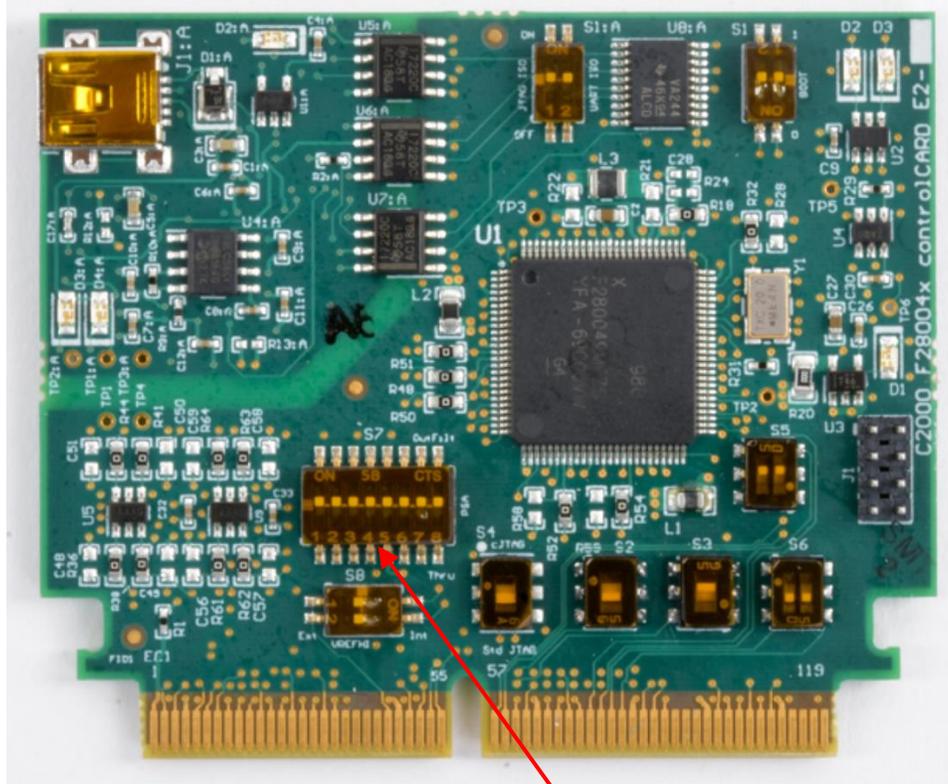
S7- PGA Filter Switches

3.1.2 Firmware

The firmware of this design is available inside the C2000Ware DigitalPower SDK and is supported inside the powerSUITE framework.

3.1.2.1 Opening the Project Inside Code Composer Studio™ (CCS) Software

To start:

1. Install the CCS software (version 8.2 or higher).

2. Open the CCS software. Go to View → CCS App Center. Under the Code Composer Studio Add-ons, make sure the GUI Composer Runtime v1.0 is installed. If the GUI is not installed, install GUI Composer Runtime v1.0.

3. Install the C2000Ware DigitalPower SDK at the *DigitalPower Software Development Kit (SDK)* for C2000 Microcontrollers tools folder. Note, the powerSUITE is installed with the C2000Ware DigitalPower SDK in the default install.

4. Close the CCS software, and open a new workspace. The CCS software automatically detects the powerSUITE. A restart of the CCS software may be required for the change to be effective.

5. Go to View → Resource Explorer.
   Under the TI Resource Explorer menu, go to Software → C2000Ware_DigitalPower_SDK_<version>.
   For example, "C2000Ware_DigitalPower_SDK_1_00_00_00".

> 注**:**   While doing these steps, ***do not*** make any modifications through the powerSUITE GUI inside the project.

To open the reference design software as is, do these steps:

1. Under the C2000Ware DigitalPower SDK, select Development Kits → Valley switching boost PFC TIDM-1022. After selected, the designs page is displayed.

   This page can be used to browse all the information on the reference design including this design guide, TI tools pages, test reports, hardware design files, and so forth.

2. Click Import <device.name> Project. This imports the project into the workspace environment, and a main.syscfg page with a GUI is displayed.

   If this GUI page does not appear, refer to the FAQ section under the powerSUITE in the C2000Ware DigitalPower SDK resource explorer.

If modifications to the GUI are needed, do these steps:

Open the reference design software for adaptation. Once the software is open, the user can modify the power-stage parameters, which are then used to create a model of the power stage in the Compensation Designer. The user can also modify scaling values for both voltages and currents.

1. Under the powerSUITE in the C2000Ware DigitalPower SDK resource explorer, click the Solution Adapter Tool.

2. From the list of solutions presented, select the valley switching boost PFC TIDM-1022.

3. On the next page, select the device this solution needs to run on (for example, TMS320F280049C).

4. Once the device for this solution is selected, a pop-up window shows asking for a location to create the project. The user can also save the project inside the workspace itself. After the location for the project is specified, a project is created, and a GUI page appears with the modifiable options for the solution (see 图 8).

5. This GUI can be used to change the parameters for an adapted solution, like power ratings, inductance, capacitance, sensing circuit parameters, and so forth.

6. If this GUI page does not appear, refer to the FAQ section under the powerSUITE in the C2000Ware DigitalPower SDK resource explorer.

图 **8. Valley Switching powerSUITE Page**
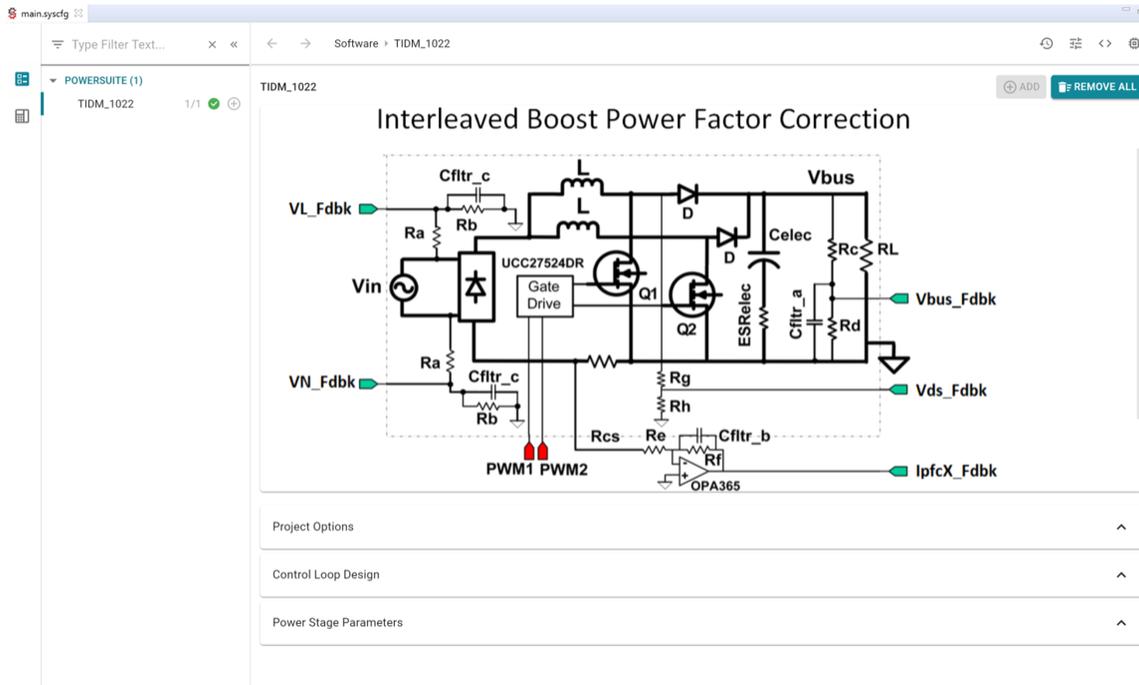
3.1.2.2   *Project Structure*

图 9 shows the general structure of the project.

The solution-specific and device-independent files are <solution >.c/h, where <solution > is "ILPFC_valley_switching". These files consist of the main.c file of the project and are responsible for the control structure of the solution.

The board-specific and device-specific files are <solution >_board.c/h and, again, <solution > is "ILPFC_valley_switching". These files consists of device-specific drivers to run the solution.

The <solution_clatask.cla> file includes the CLA Task definitions.

The powerSUITE page can be opened by clicking on the main.syscfg file, listed under the Project Explorer (see 图 9). The powerSUITE page generates the <solution >_settings.h file. This file is the only file used in the compile of the project that is generated by the powerSUITE page. The user *must not* modify the content above *user section* in this file manually, as the changes will be overwritten by the powerSUITE every time the project is saved. The user also need to rebuild the whole project to change the <solution >_settings.h file. For <solution >_user_ settings.h, user can open it directly and modify it.

The "Kit.xml" and "solution.js" files are used internally by the powerSUITE and *must also not* be modified by the user. Any changes to these files will result in the project not functioning properly.

图 **9. Project Structure**
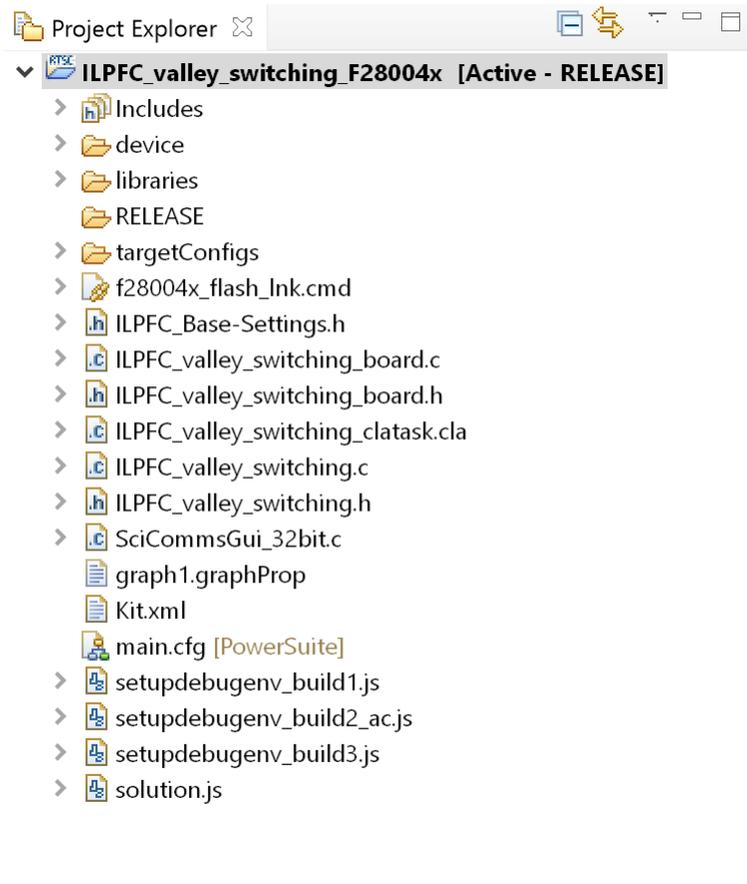
3.1.2.3   Using Control Law Accelerator (CLA) on C2000 MCU to Alleviate CPU Burden

The CLA is a coprocessor available on the C2000 MCU family of devices. This coprocessor enables offloading the control-ISR functions from the main C28x CPU core.

The control-ISR code is *not* duplicated in the CLA and a single source for the solution algorithm is maintained, even when code is run on the CLA or the C28x CPU core. This configuration enables flexible debugging of the solution.

The CLA features vary slightly from device-to-device. For example, on the F2837xD, F2837xS, and F2807x, the CLA can support only one task at a given time and there is no nesting capability, which means the task is *not* interruptible. Hence, realistically, only one ISR can be offloaded to the CLA. On the F28004x, the CLA supports a background task from which a regular CLA task can nest. This configuration enables offloading two ISR functions to the CLA.

F28004x device-specific information:

The CLA on the F28004x device, supports a background task from which a regular CLA task can nest. This configuration enables offloading two ISR functions to the CLA. However, for the valley switching reference design, only the fastest variable frequency ISR is ported to the CLA. On the F28004x, the CPU usage will be above 80%, in total, if all three ISR functions remain in the CPU. With the nested CLA option, the CPU burden can be significantly reduced since the PWMISR occupies close to 40% of the CPU usage.

3.1.2.4   Running the Project

This section details the necessary equipment, test setup, and procedural instructions for the design board and software testing along with validation.

1. For input, the power supply source (VIN) must have a range from 100- to 260-V AC. Set the input current limit of AC input source to 12 A for full-power tests, and 1 A for valley switching tests. Start with a lower-current limit during initial board bring-up. For the output, use an electronic-variable load or a variable-resistive load, which must be rated for ≥400 V.
2. Test Equipment Required for Hardware Validation:
    *   Isolated AC source
    *   Single-phase power analyzer
    *   Digital oscilloscope
    *   Multimeters
    *   Electronic or resistive load (variable)
3. Test Setup (mainly for Build 1)
    a. Connect the (180- to 100-pin) adapter to the C2000 F280049C controlCARD
    b. Connect the adapter to the ILPFC board.

    注: ***Do not*** "turn on" the power sources, until indicated.

    ---

    c. Connect an external 12-V DC bench power supply to the ILPFC board (JP1)
    d. Connect a DC power supply between the PFC-In (BS3) and GND (BS6) input terminals.
    e. For the initial test, set and maintain a load of about 100 mA (4 kΩ) at a 380-V DC output.
    f. Connect a voltmeter, oscilloscope probes, and other measurement equipment to probe or analyze various signals and parameters, as desired. Only use appropriately-rated equipment and follow proper isolation and safety practices.
    g. With the power supplies still "turned off", connect a USB B-to-A cable between the PC and the C2000 controlCARD.

4. Download and install in the default folder the C2000WARE-DIGITALPOWER-SDK software package. The software project should now reside inside the C2000Ware DigitalPower SDK folder at <install_location>\solutions\tidm_1022.

Do these steps to build and run this code with different incremental builds. Note that the builds need to be done one by one.

3.1.2.4.1   INCR_BUILD 1.1: Open Loop, DC

The objective of this build is to:

- Evaluate the open loop operation of the system
- Verify the PWM and ADC driver modules
- Verify the FET driver circuit and sensing circuit on the board
- Become familiar with the operation of the CCS software

The control software diagram of build 1.1 and 1.2 is shown in 图 10.

图 **10. Build Level 1 Control Software Diagram: Open Loop Project**
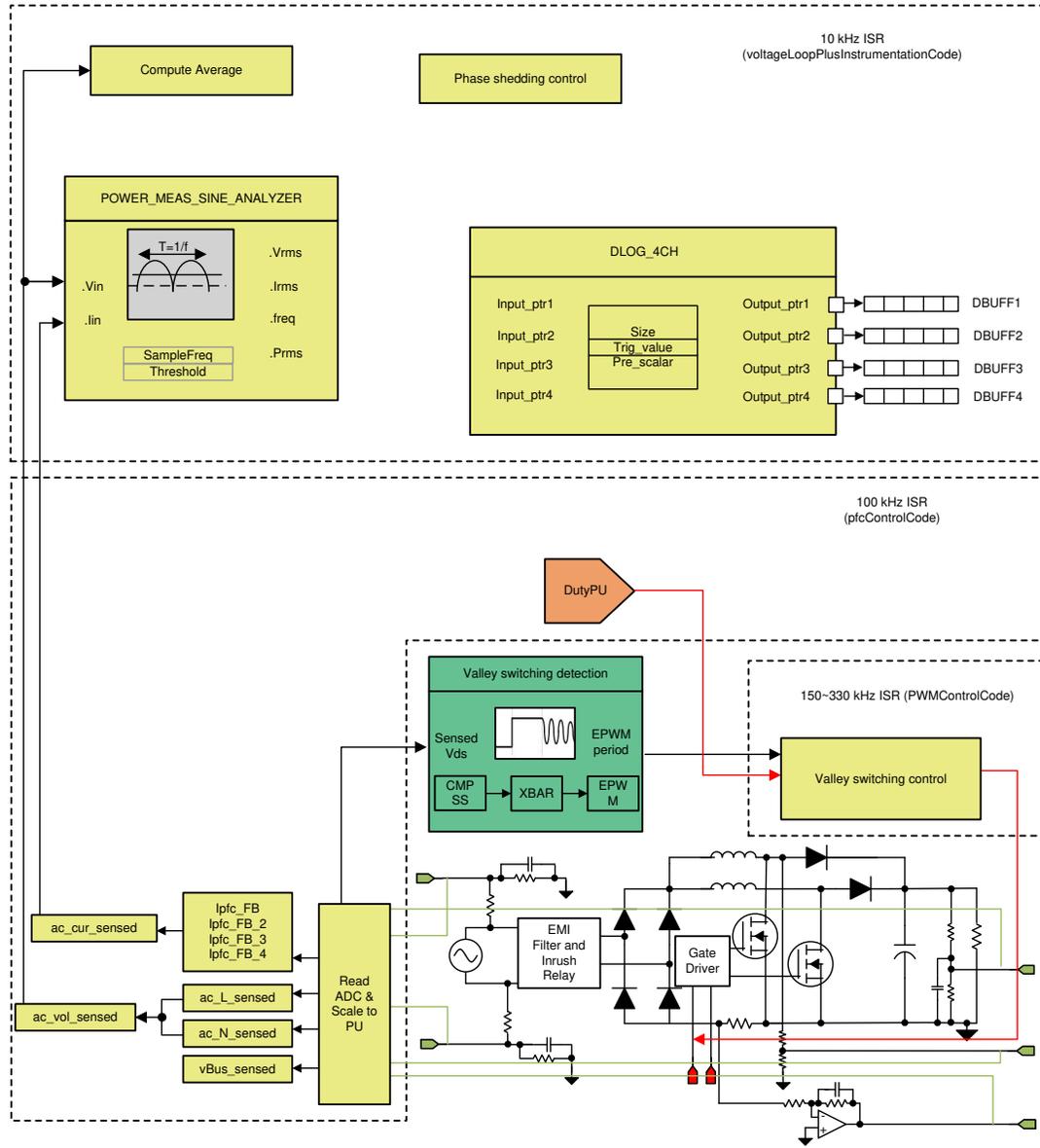
注: Since this system is running an open-loop, the ADC measured values are only used for instrumentation purposes. Only the DC input is used in this build.

3.1.2.4.1.1 *Start the CCS Software and Open a Project*

To quickly execute this build:

1. Connect an isolated DC power supply capable of providing at least 100 V at up to 1 A to the input terminals (PFC-In [BS3] and GND [BS6]) of the modified reference board. Set the power supply to output 0 V with a current limit of 1 A. ***Do not*** "turn on" the power supply at this time.

2. Connect a 400-V resistive load of about 4 kΩ at the output terminal (PFC-Out [BS4]).

3. Open the CCS software (v8.2 or higher). Maximize the program to fill the screen.
   Close the welcome screen, if it pops up.

注**:**   A project contains all the files and build options needed to develop an executable output file (.out), which can be run on the MCU hardware.

4. On the menu bar, click *Project → Import Existing CCS/CCE Eclipse* project.
   Under the Select root directory to navigate to the C2000Ware DigitalPower SDK folder and select *<install_location>\solutions\tidm_1022*.
   Click *Finish*.

   The *CCS/CCE Eclipse Project* invokes all the necessary tools (compiler, assembler, linker, and so forth) to build the project.

3.1.2.4.1.2 Device Initialization, Main, and ISR Files

1. In the Project window, on the left, click the plus sign (+) to the left of "Project".
2. Open the *main.syscfg* file by double-clicking on the filename in the Project window.
3. Under *Project Options*, select *Open Loop* as the *INCR_BUILD* option (see 图 11).
4. Check the project options to make sure that *2-phase interleaved PFC* is selected and that the *valley switching* option is automatically set to *valley switching disabled*.
   This helps verify that the ADC and PWM driver modules are under normal interleaved PFC.
   Save the *main.syscfg* file. Rebuild the whole project to make the change valid if any changes are needed.
5. Open and inspect the *ILPFC_valley_switching.c* file.
   Notice the call made to the *setupDevice()* function and other initialization functions like *InitPwms()*, *setupADC()*, and so on. Also notice code for the ISRs and the background loop.
   Close the file, **do not** Save.
6. Open and inspect *ILPFC_valley_switching.h* file.
   Notice the code for different incremental build options inside the *pfcControlCode()* specifically, Build 1.
   Close the file, **do not** Save.

图 **11. powerSUITE Page Under Build 1 Without Valley Switching**
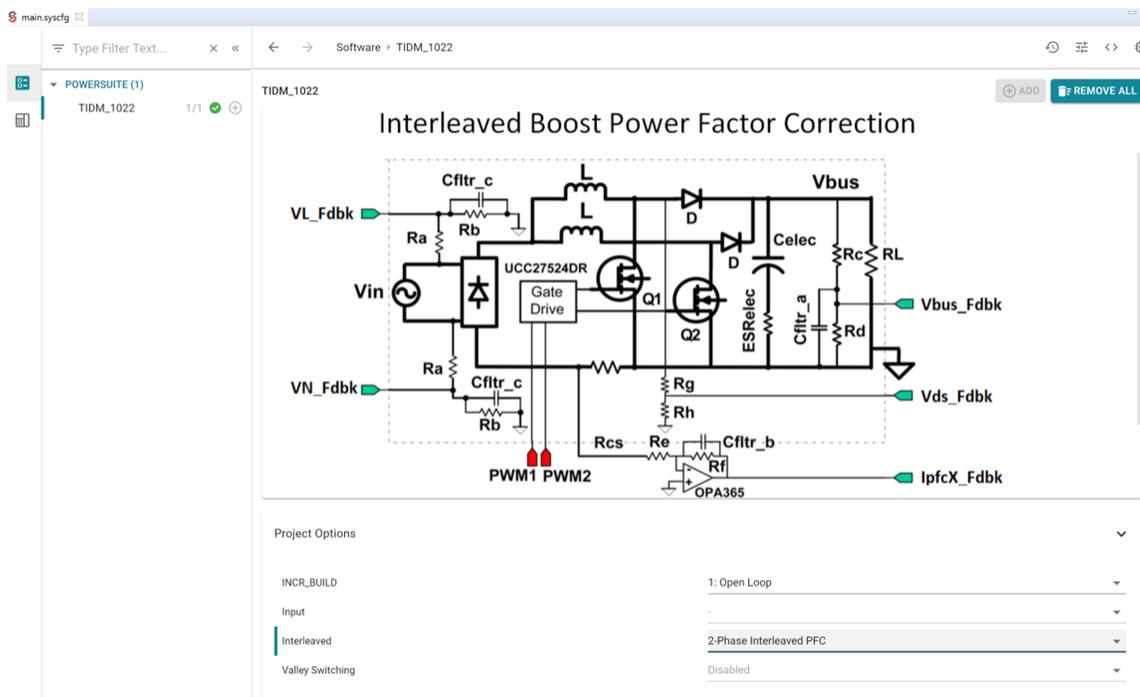
### 3.1.2.4.1.3 Build and Load the Project

1. Open the ILPFC_settings.h file. Notice that the incremental build (*INCR_BUILD*) option is set to "1" (Build 1) and the *INTERLEAVE* option is set to "2" (valley switching control is disabled).

2. If another build option was built previously, right click on that project name and click *Clean Project*. Click *Project → Build All* button, and watch the tools run in the build window.

3. "Turn on" the 12-V DC bench power supply. Click the *Debug* button *or* click *Run → Debug*. When completed, the Build 1 code should compile and load.

4. Notice the CCS Debug icon in the upper right-hand corner, indicating that the user is now in the Debug Perspective view. The program should be stopped at the start of *main()*.

3.1.2.4.1.4 Debug Environment Windows

It is standard debug practice to watch local and global variables while debugging code. There are various methods for doing this in the CCS software, such as memory views and watch views. Additionally, the CCS software has the capability to make time (and frequency) domain plots. This capability to make time/frequency domain plots allows the user to view waveforms using graph windows.

- To add the variables in the Expressions window, click *View → Scripting* console to open the scripting console dialog box. In the upper-right corner of this console, click *Open*, then browse to the *setupdebugenv_build1.js* script file located inside the Project folder. This script file populates the Expressions window with the appropriate variables needed to debug the system. Click the *Continuous Refresh* button (), in the Expressions window, to enable continuous updates of values from the controller. The Expressions window will appear as shown in 图 12.

图 12. Expressions Window View
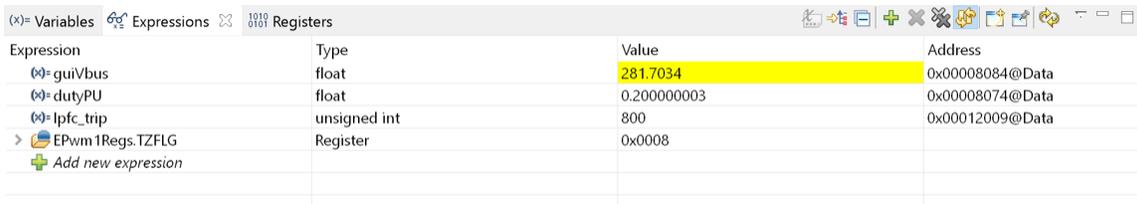
- The current and voltage measurements can be verified by viewing the data in the graph window. The current and voltage measurement values are logged in the slower 10-kHz routine. Go to *Tools → Graph → DualTime*. The user can enter the values as shown in the 图 13. Once the entries are verified, click *OK*. Two graphs will appear in the CCS software. Click on the *Continuous Refresh* button on each graph. A second set of graphs can also be added by importing the *graph2.GraphProp* file.

图 **13. Graph settings**
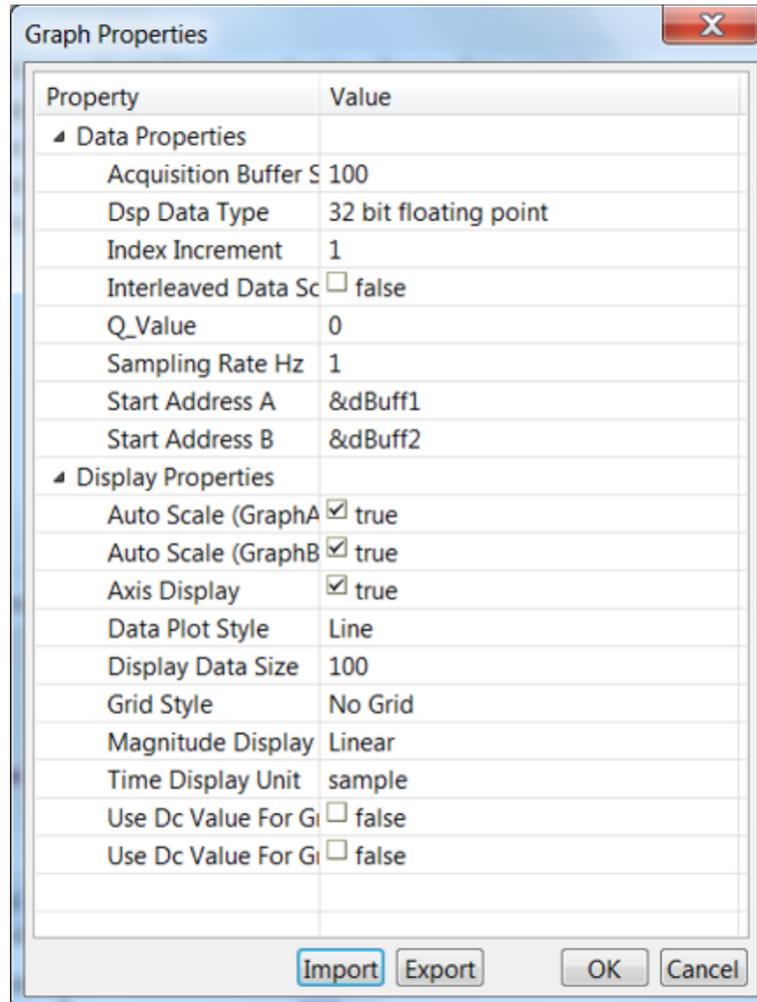
3.1.2.4.1.5 Using Real-Time Emulation

Real-time emulation is a special emulation feature that allows the windows within the CCS software to be updated at up to a 10-Hz rate while the MCU is running. This emulation not only allows watch and graphs views to update, but also allows the user to change values in the watch or memory windows and have those changes affect the MCU behavior. This is very useful when tuning control law parameters in real-time, for example.

1. Enable real-time mode by hovering the mouse on the buttons on the horizontal toolbar and clicking the *Enable Silicon Real-time Mode* button (🔊).

2. A message box may appear. If so, select *YES* to enable debug events. Enabling the debug events, will set the Status Register 1 (ST1) Debug Enable Mask bit (DGBM) [bit 1] to "0". When the DGBM bit is set to "0", the memory and register values can be passed to the Host processor for updating all the debugger windows.

3. As bandwidth over the emulation link is limited, when a large number of windows are open, updating too many windows and variables in continuous refresh can cause the refresh frequency to become slow.

   Right click the down arrow button in the Expressions window and select the *Continuous Refresh Interval* button. The user can slow down the refresh rate for the Expressions window variables by changing the continuous refresh interval value (in milliseconds). A rate of 1000 ms is usually enough for these debug exercises.

4. Click the *Continuous Refresh* button for the Expressions view.

3.1.2.4.1.6 Run the Code

1. Run the code by using the <F8> key, *or* by using the *Run* button (▶▼) on the toolbar.

2. The dutyPU parameter should be set to 0.2 initially. Check the PWM switching frequency which should be set to 200 kHz as shown in 图 14. And verify channel A and channel B of PWM1 module maintain the output.

3. "Turn on" the power supply and gradually increase the voltage to 100 V.

4. The *guiVbus* and *ac_cur_sensed* fields should correctly reflect the output parameters. If this is the case, proceed to the next step. If this is not the case, an option is to debug the sensing circuit for the parameters that did not update correctly.

5. With a 4-kΩ load, the output voltage should go up to approximately 280 V (see 图 15).

6. Slowly increase the dutyPU parameter in increments of 0.02. The output voltage should change accordingly. Make sure the output voltage does not go above 400 V.

7. The user can also experiment with different load values and different input voltage values. Make sure that the board is never operated out of the specifications (for the system specifications, see 表 1).

8. The user can also probe the drain-to-source voltages for the phase 1 using appropriately-rated voltage probes and oscilloscopes. 图 16 provides the Vds waveforms under normal PFC conditions.

9. Set the *guiVbusMax* parameter below the current output voltage. The PWM modules should shut down. When this shutdown happens, first "turn off" the DC power source, then, take the MCU out of real-time mode and then reset the device by clicking the reset button (🔄).

10. Repeat step 1 through step 8, only this time, set the *Ipfc_trip* parameter below the input current level that is being drawn. The PWM modules should again shut down and the one shot bit of EPWM1regs.TZFLG register should be set to "1" in the Expressions window.
    This validates the Build 1 of the software, along with all the sensing circuit, and protection mechanisms, and so forth.

11. To fully halt the MCU when in real-time mode, is a two-step process. First "turn off" the 100-V supply and wait a few seconds.
    Halt the processor by using the *Suspend* button (⏸), or by using *Target →Halt*. Then, click the *Enable Silicon Real-time Mode* button again to take the MCU out of the real-time mode. Reset the MCU.

12. Either leave the CCS software running for the next exercise *or* close the CCS software.
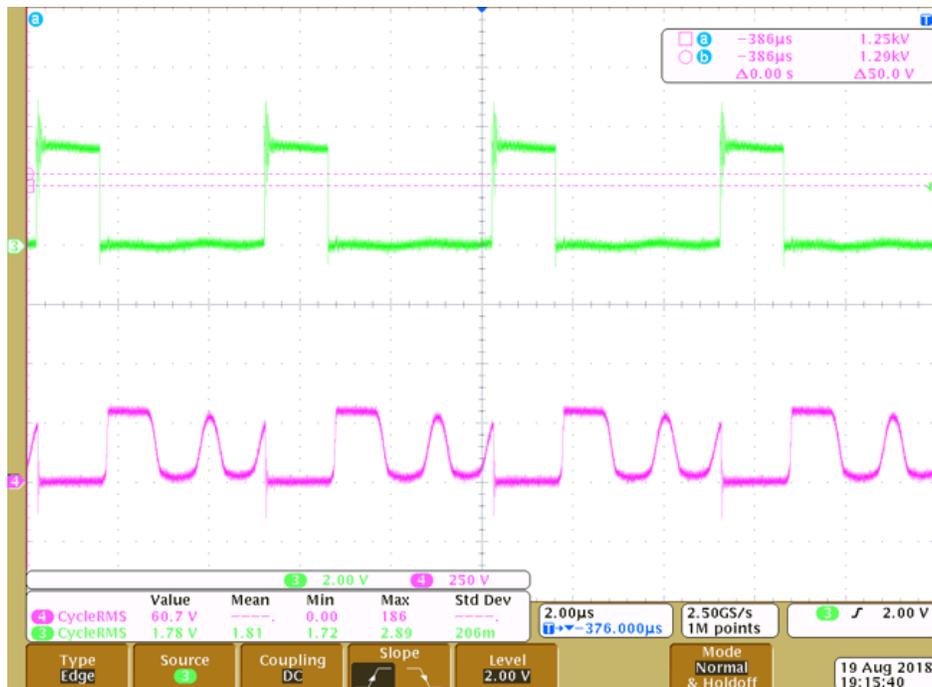
图 **14. Build 1 PWM Output**

图 **15. Build 1 Expressions Window View**

图 **16. Build 1 Vds(lower) and PWM Output(upper) Without Valley Switching**

3.1.2.4.2 *INCR_BUILD 1.2: Open Loop, DC (Valley Switching Validation Under Open Loop)*

The objective of this exercise is to evaluate the valley switching under open loop with DC input.

1. Open the *main.syscfg* file by double-clicking on the filename in the Project window. Under *Project Options*, select *Open Loop* as the *INCR_BUILD* option (see 图 17). Select the *Single-phase PFC* and the *valley switching* options should be automatically set to *valley switching enabled*. This enables valley switching under Build 1. Rebuild the project if there are any changes in the solution_settings.h.

2. Open the *solution_settings.h* file. Notice that the incremental build (*INCR_BUILD*) option is set to "1" (Build 1) and the *INTERLEAVE* option is set to "1" (valley switching control is enabled). If not, go back to step 1 to make sure the setting is correct.

3. "Turn on" the 12-V DC bench power supply. Click the *Debug* button *or* click *Run → Debug* from the menu. Once complete, the Build 1 code should compile and load.

4. Repeat step 1 through step 4 in 节 3.1.2.4.1.5 for real-time emulation and to continuously refresh.

5. Probe the drain-to-source voltages for the phase 1 using appropriately-rated voltage probes and oscilloscopes.

6. Set the resistive load to 4 kΩ.

7. Run the code by using the <F8> key *or* by using the *Run* button ( ) on the toolbar.

8. The dutyPU parameter should be automatically set to 0.2 initially. Verify channel A of the PWM1 module maintains the output and that channel B is disabled.

9. "Turn on" the power supply and gradually increase the voltage to 100 V. This will increase the output to approximately 220 V.

10. Using the oscilloscope, observe the Vds voltage waveform . The MOSFET (Q1) should be "turned on" at the valley point of Vds (see 图 18).

11. Inspect the values in the valley switching-related registers (that is, DCFCTL, VCAPCTL, and VCNTCFG). For more information on these registers and the associated values, refer to the Valley Switching Application Report and TMS320F28004x Piccolo Microcontrollers Technical Reference Manual .

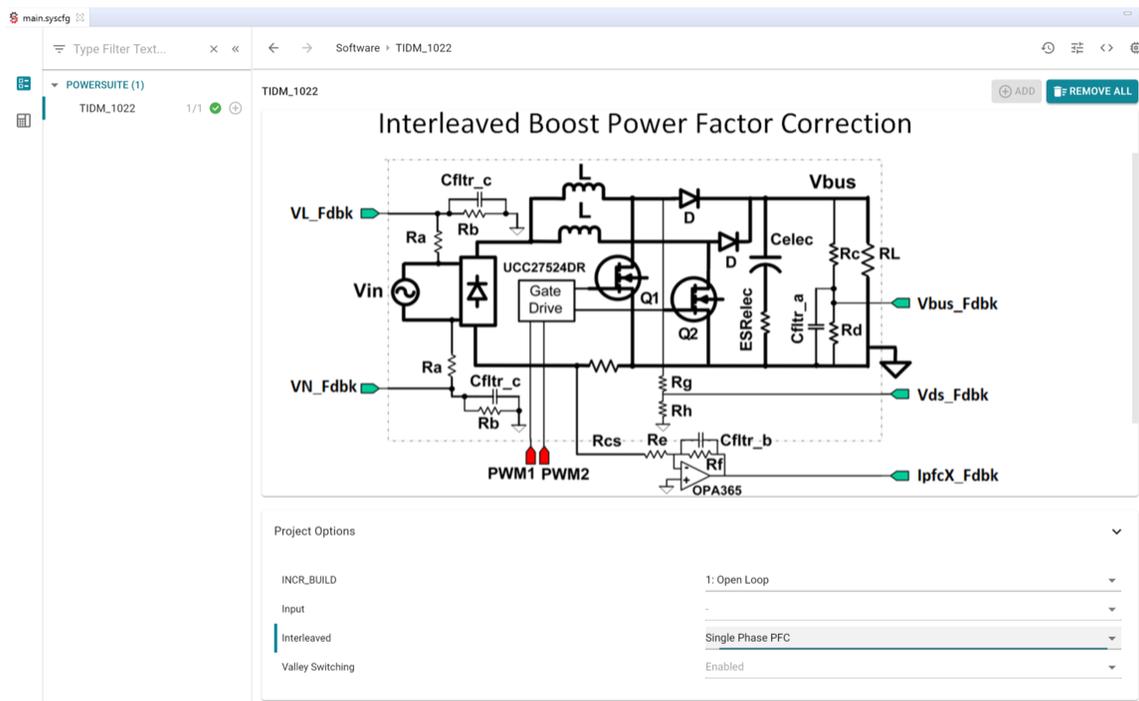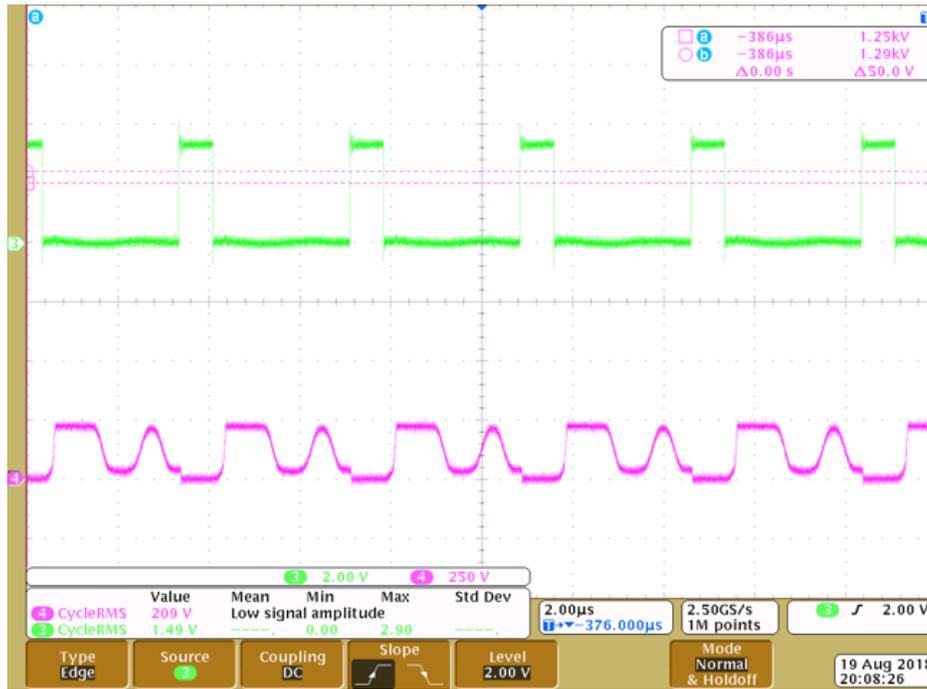**图 17. powerSUITE Page Under Build 1 With Valley Switching**

图 **18. Build 1 Vds(lower) and PWM Output(upper) With Valley Switching**

3.1.2.4.3 *INCR_BUILD 2: Current Closed Loop, AC*

The objective of this build is to evaluate the closed current loop operation of the system. The AC input is applied for this build. If this is the first time this board is powered, start with Build 1.

The control software diagram of build 2 is shown in 图 19.

图 **19. Build Level 2 Control Software Diagram: Closed Current Loop Project**
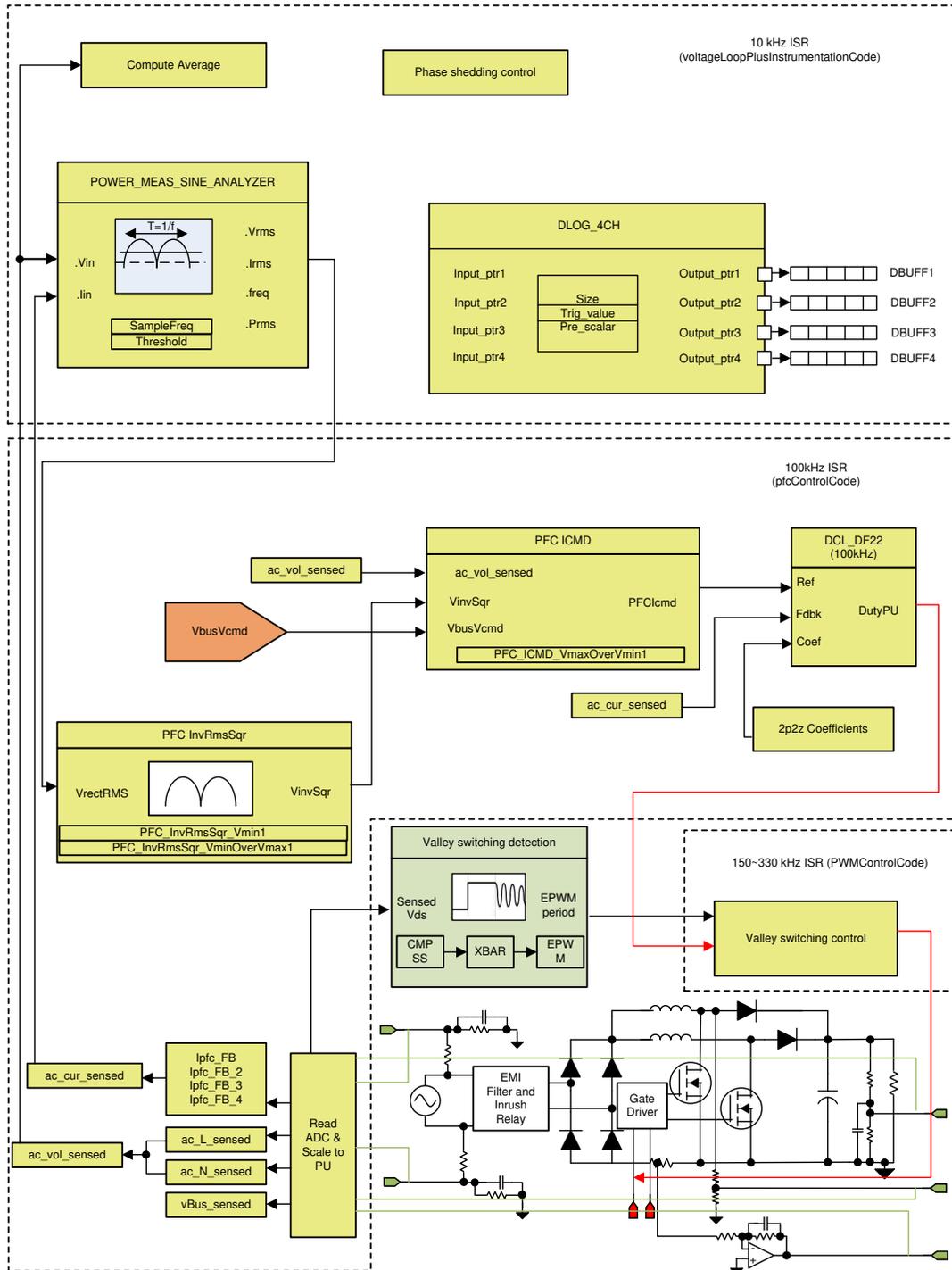
3.1.2.4.3.1 Start the CCS Software and Open a Project

To quickly execute this build:

1. Connect the "bridge out" (BS1) to the "PFC-in" (BS3) input terminal of the modified ILPFC board.

2. Connect a programmable, isolated AC power supply capable of providing a universal AC input up to 750 kW to the input terminal (P1) of the modified reference base board. Set the power supply to output 0 V with a current limit to 1 A and output frequency to 60 Hz. *Do not* "turn on" the power supply at this time.

3. Connect a 400-V resistive load of about 4 kΩ at the output terminal (PFC-Out [BS4]).

4. Repeat step 3 and step 4 of 节 3.1.2.4.1.1.

3.1.2.4.3.2 Build and Load the Project

1. Open *main.syscfg* file by double-clicking on the filename in the Project window. Under the *Project Options*, select *Closed Current Loop* as the *INCR_BUILD* option. Select *Single-phase PFC* and the *valley switching* option should be automatically set to *valley switching enabled*. Save the *main.syscfg* file. Rebuild the whole project to make the change happen. This is shown in 图 20.

2. Open *solution_settings.h* file. Notice that the incremental build (*INCR_BUILD*) option is set to "1" (Build 2) and *INTERLEAVE* option is set to "1" (valley switching control is enabled).

3. If another build option was built previously, right click on the project name and click on *Clean Project*. Click *Project → Build All* button and watch the tools run in the build window.

4. "Turn on" the 12-V DC bench power supply. Click the *Debug* button or click *Run → Debug* from the menu. Once complete, the Build 2 code should compile and load.

5. Notice the CCS Debug icon in the upper right corner, indicating that the user in now in the Debug Perspective view. The program should be stopped at the start of *main()*.

6. Repeat step 1 through step 4 in 节 3.1.2.4.1.5 for real-time emulation and to contiguously refresh.

图 20. powerSUITE Page Under Build 2 With Valley Switching
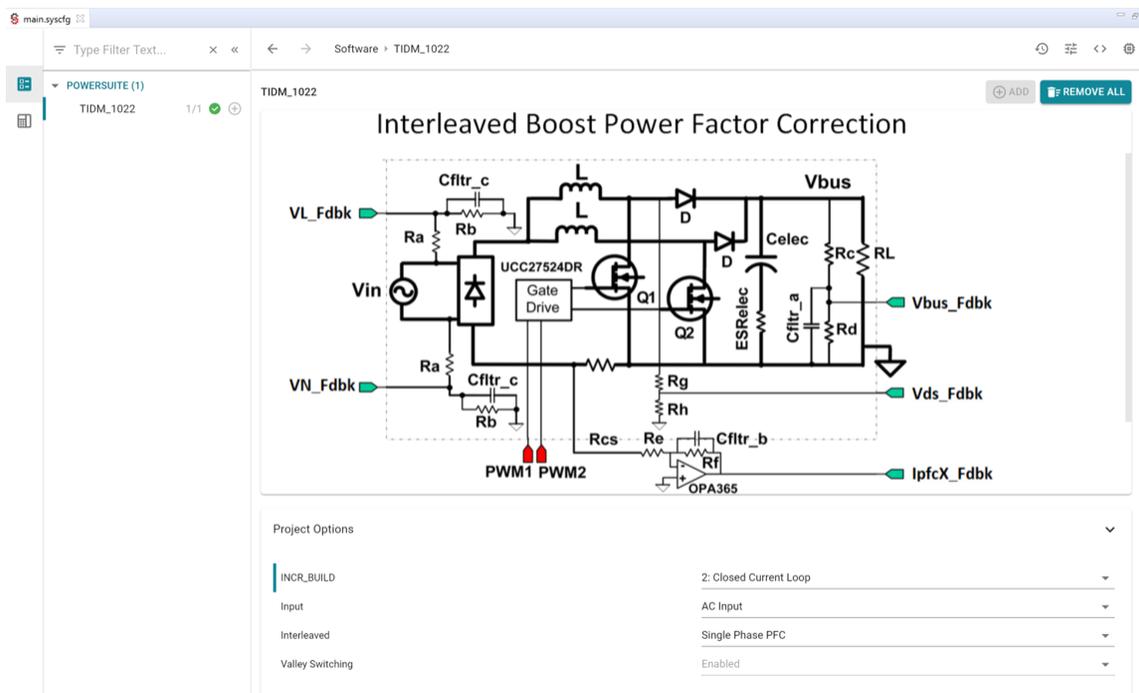
3.1.2.4.3.3 Run the Code

1. Probe the drain-to-source voltages for the phase 1 using appropriately-rated voltage probes and oscilloscopes.

2. Follow the step 1 in 3.1.2.4.1.4 to open setupdebugenv_build2_ac.js script file located inside the Project folder. This script file populates the Expressions window with the appropriate variables needed to debug the system for build 2.

3. Run the code by using the <F8> key, *or* by using the *Run* button ( ) on the toolbar.

4. With the AC source set to output 0 V at 60 Hz, "turn on" the AC power supply.

5. Set *VbusVcmd* to 0.025.

6. Gradually increase the input voltage from 0-V to 120-V AC.

7. *Gui_Freq_Vin*, *Gui_IrectRMS*, *Gui_VrectRMS*, and *guiVbus* should correctly reflect the input and output parameters, respectively.

8. With a 4-kΩ load, the output voltage should go boost up to about 260 V (see 图 21). 图 22 shows the input current, PWM output, and Vds waveforms under this condition.

9. Zoom in on the Vds waveform to verify the valley switching in each cycle under 120 V input conditions. The Vds waveform should be similar to 图 23.

10. Keep increasing the *VbusVcmd* in increments of 0.002 until the *VbusVcmd* is increased to a value of 0.045. The output voltage should increase to approximately 380 V (see 图 24). The *ac_cur_sensed* value may change rapidly and is not the same as the *ac_cur_ref* value. This change is because the *ac_cur_sensed* field is the instantaneous input current value while the *ac_cur_ref* is the amplitude reference for the current command.

11. The SFRA is integrated in the software of this build (as well as integrated in the software of Build 3) to verify the designed compensator provides enough gain and phase margin by measuring the hardware. To run the current loop SFRA, the user needs to select *current* option in main.syscfg(project rebuilt needed to change the settings). To run the SFRA, keep the project running, and from the main.syscfg page, click the SFRA icon (see 图 25). The SFRA GUI appears.

12. Select the options for the device on the SFRA GUI. For example, for F280049C select *floating point*. click the Setup Connection. From the pop-up window, uncheck the *boot on connect* option, and select an appropriate COM port. Click *OK*. Return to the SFRA GUI, and click *Connect*.

13. The SFRA GUI connects to the device. An SFRA sweep can now be started by clicking *Start Sweep*. The complete SFRA sweep takes a few minutes to finish. Activity can be monitored by viewing the progress bar on the SFRA GUI and also checking the flashing of blue LED on the back on the controlCARD that indicates UART activity. Once complete, a graph with the open loop plot appears, see 图 26.

14. Optionally, the user can use the measured frequency response of the plant to design the current compensator by clicking the *Compensation Designer* button again from the main.cfg page and choose the *SFRA Data* for *plant* option on the GUI. This uses the measured plant information to design the compensator. This option can be used to fine tune the compensation. By default, the compensation designer points to the latest SFRA run. If a previous SFRA run plant information is needed, the user can select the *SFRAData.csv* file by clicking the *Browse SFRA csv Data.*

图 **21. Build 2 Expressions Window View**

图 **22. Input Current Waveform(Channel 1) Under Build 2 With 0.025 dutyPU**

图 **23. Vds Waveform(Channel 4) Under Build 2 With Valley Switching**

图 **24. Input Current Waveform Under Build 2 With 0.045 dutyPU**

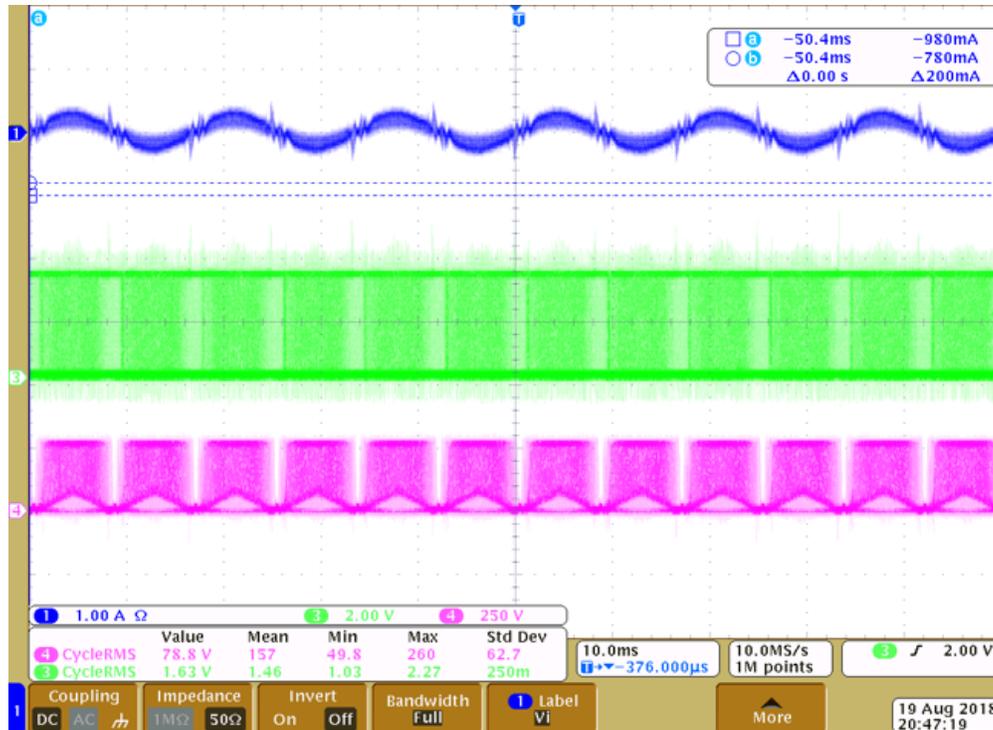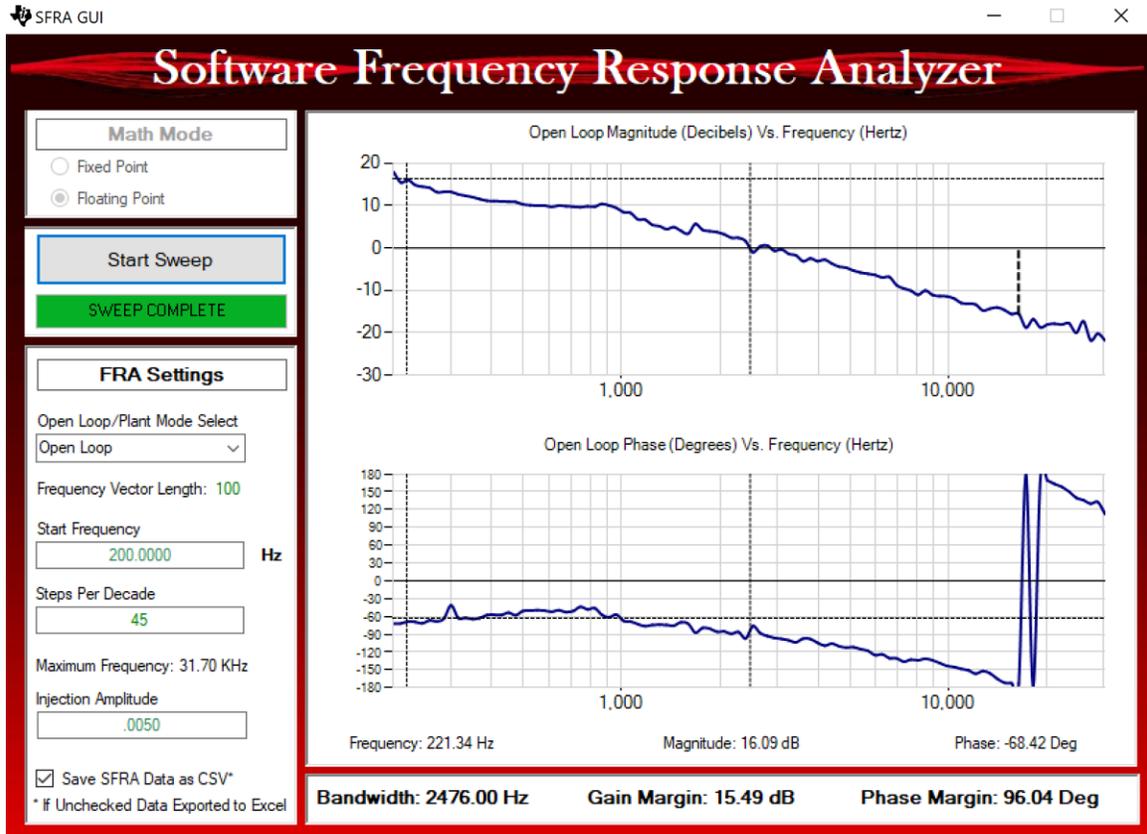图 **25. powerSUITE Page to Select and Open SFRA**

图 **26. Current Loop SFRA Result**

3.1.2.4.4   *INCR_BUILD 3: Voltage and Current Closed Loop, AC*

The objective of this build is to verify the operation of the complete valley switching PFC project from the CCS software environment with voltage and current closed. If this is the first time this board is powered, start with Build 1.

The control software diagram of build 3 is shown in 图 27.

图 27. Build Level 3 Control Software Diagram: Closed current and voltage Loop Project
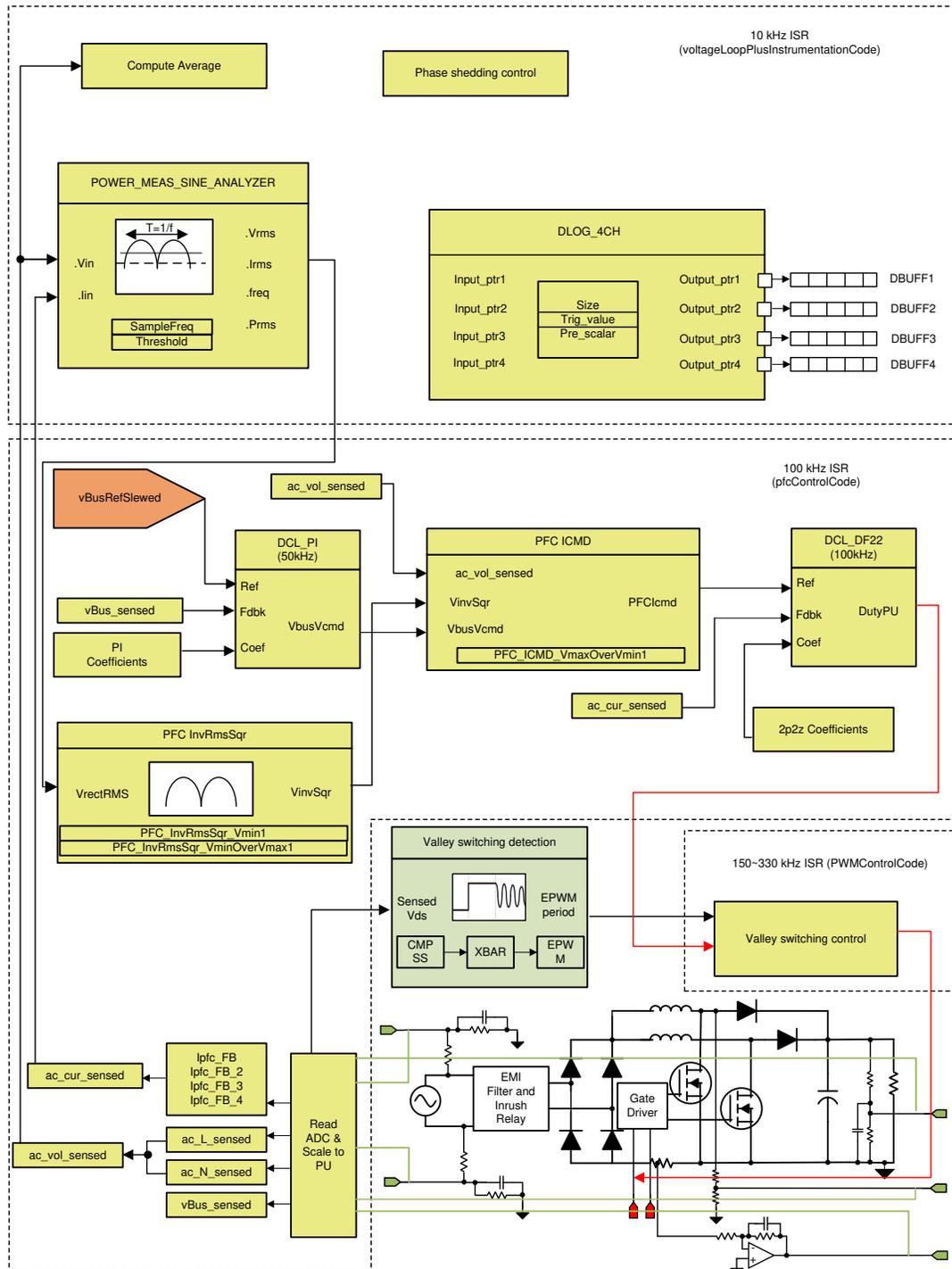
3.1.2.4.4.1   Start the CCS Software and Open a Project

To quickly execute this build:

1. Connect a programmable, isolated AC power supply capable of providing universal AC input up to 750 kW to the input terminals of the board. Set the power supply current limit to 1 A and the output frequency to 60 Hz. Do not turn the power supply on at this time.

2. Connect a 400-V resistive load of about 4 kΩ at the output.

3. Repeat step 3 and step 4 of 节 3.1.2.4.1.1.

3.1.2.4.4.2   Build and Load the Project

1. Open main.cfg file by double-clicking the filename in the Project window. Under *Project Options*, select *Closed Current Loop & Closed Voltage Loop* as the INCR_BUILD option. Select the *Single phase PFC* and the valley switching options should be automatically set to *valley switching enabled* as shown in 图 28. Save the main.cfg file. Rebuild the project if there are any changes in solution_settings.h.

2. Open solution_settings.h file. Notice that the incremental build (INCR_BUILD) option is set to "3" and the *INTERLEAVE* option is set to "1" (valley switching control is enabled).

3. If another build option was built previously, right click the project name and then click Clean Project. Click Project → Build All button and watch the tools run in the build window.

4. Turn on the 12-V DC bench power supply. Click the Debug button or click Run → Debug. The build 3 code should compile and load.

5. Notice the CCS Debug icon in the upper right-hand corner, which indicates the Debug Perspective view. The program should be stopped at the start of main().

6. Repeat step1 through step4 in 节 3.1.2.4.1.5.

图 **28. powerSUITE Page Under Build 3 With Valley Switching**
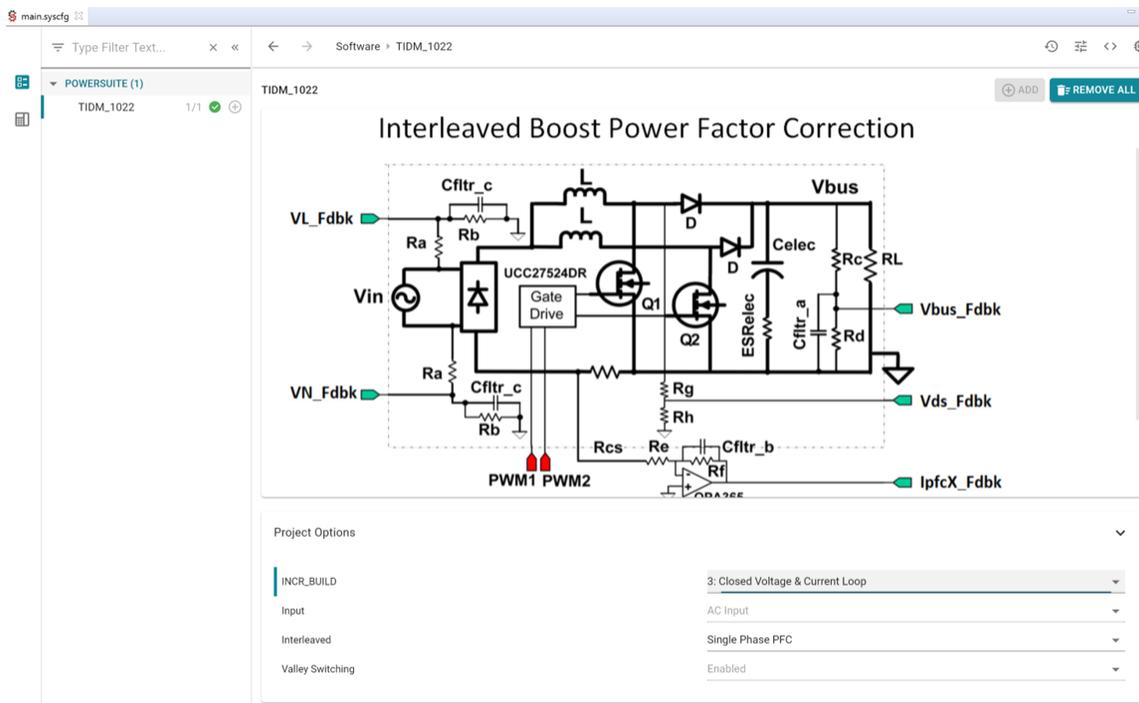
3.1.2.4.4.3 *Run the Code*

1. Probe the drain-to-source voltages for the phase 1 using appropriately-rated voltage probes and oscilloscopes.

2. Follow the step 1 in 3.1.2.4.1.4 to open setupdebugenv_build3.js script file located inside the Project folder. This script file populates the Expressions window with the appropriate variables needed to debug the system for build 3.

3. Run the code by using the <F8> key,

4. *or* by using the *Run* button on the toolbar.

5. With the AC source set to output 0 V at 60 Hz, turn on the AC power supply.

6. Slowly increase the input voltage from 0-V to 120-V AC with a 4-kΩ load. Once the output voltage reaches 160 V, the PFC starts working under the closed current loop and closed voltage loop.

7. The output voltage should boost up from 160 V to about 380 V automatically.

8. If the input voltage is increased or decreased, the output should be maintained to around 380 V.

9. Zoom in to the Vds waveform to verify the valley switching, ZVS and fixed freq in each cycle under 120-V input and 220-V input conditions.

10. GPIO 15 is originally selected in the code to monitor the valley switching and ZVS transition which should be shown like 图 29 and 图 30.

11. Resistive load can be changed at this time to test valley switching under different load condition(<10%).

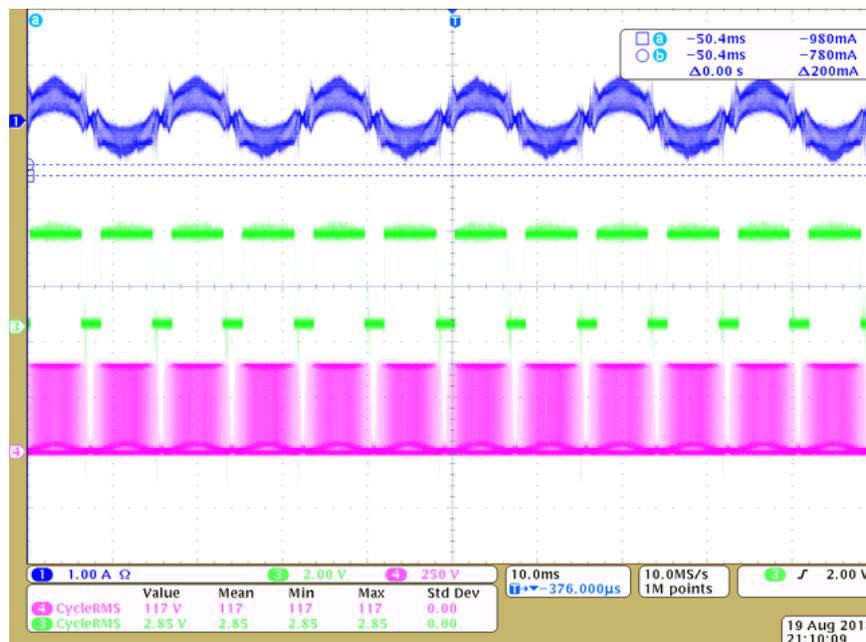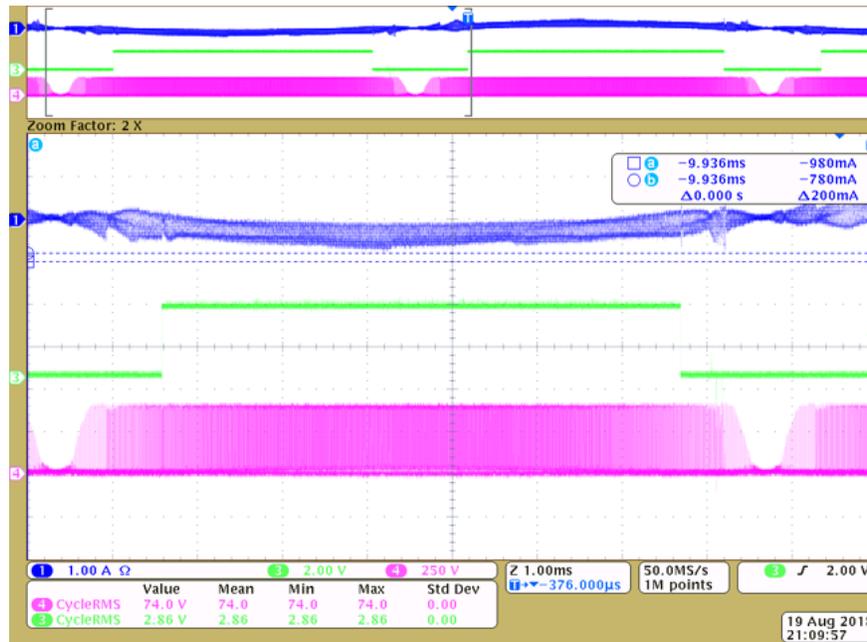图 **29. Valley Switching and ZVS Transition (Channel 1: Iin, Channel 3: GPIO)**

图 **30. Valley Switching and ZVS Transition Zoom In (Channel 1: Iin, Channel 3: GPIO)**

3.2 Testing and Results

图 31 and 图 32 show the current wavefrom with valley switching control only and with valley switching + ZVS + fixed freq control. 图 33 is the zoom in input current waveform near zero crossing to show the control logic. For the detailed control algorithm, please refer to the Valley Switching Application Report.

图 **31. High Current Distortion Near Zero Crossing Point**
Without Optimized ZVS and Fixed Frequency Control
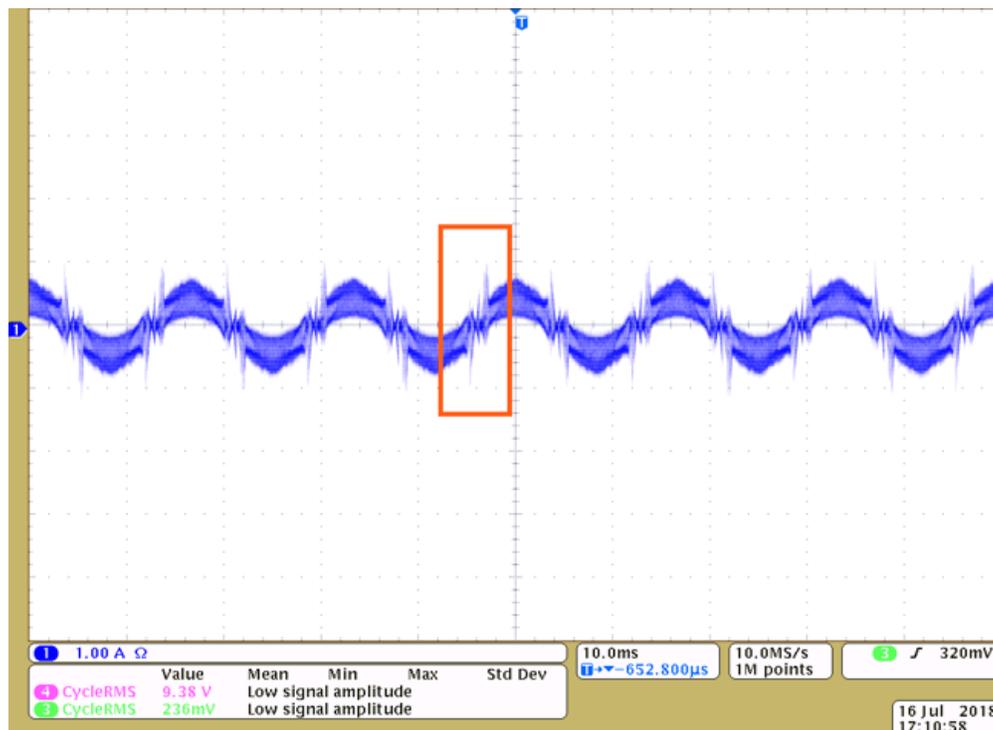
图 **32. Low Current Distortion Near Zero Crossing Point
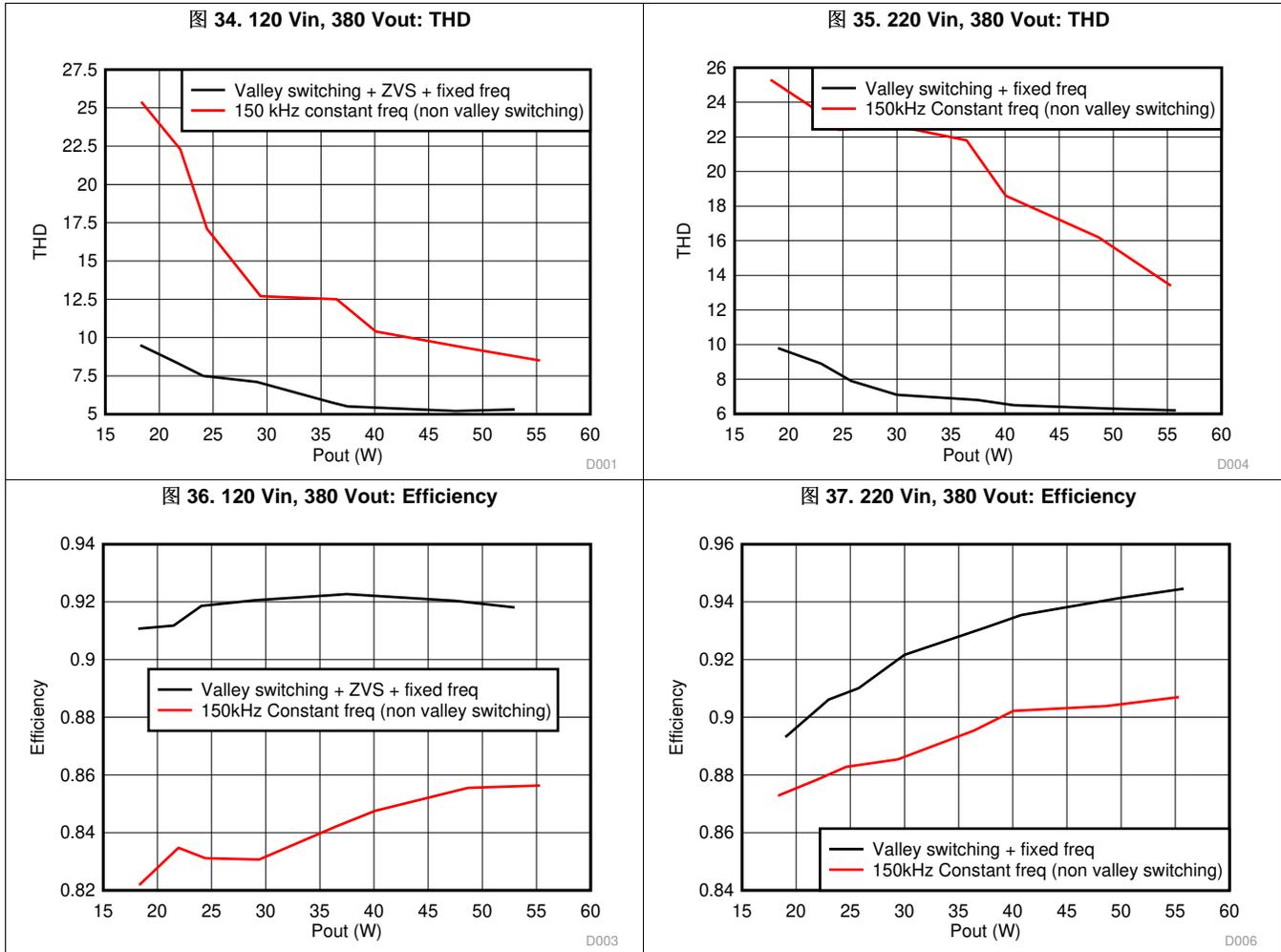With Optimized ZVS and Fixed Frequency Control**
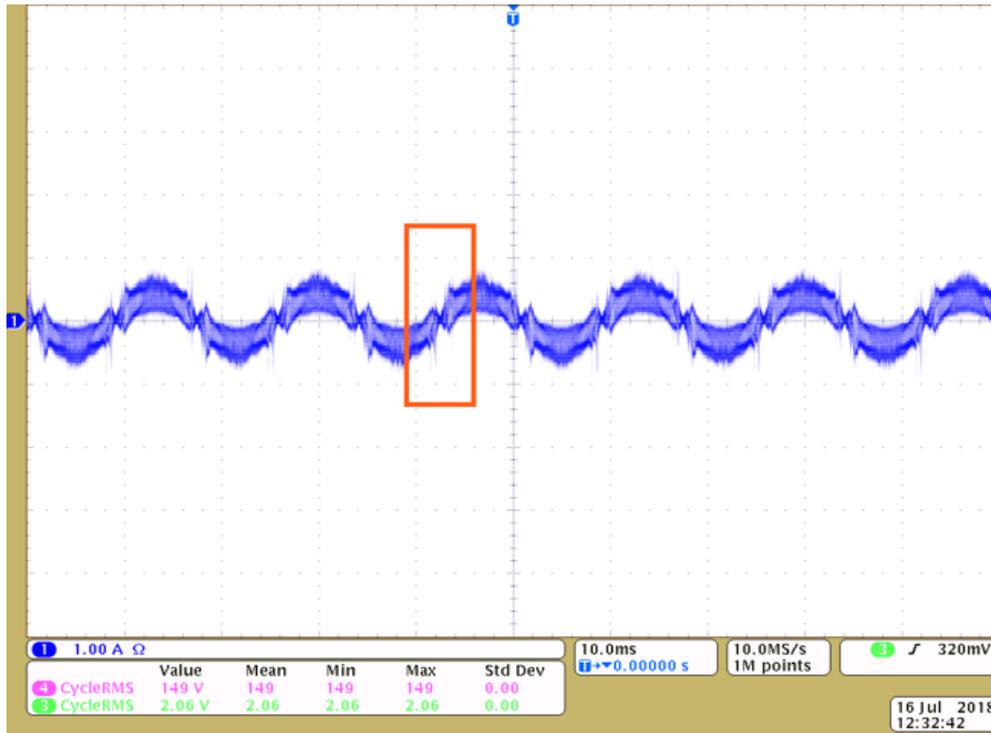
图 **33. Input Current (Channel 1) Near Zero Crossing to Show the Control Logic Zoom In**
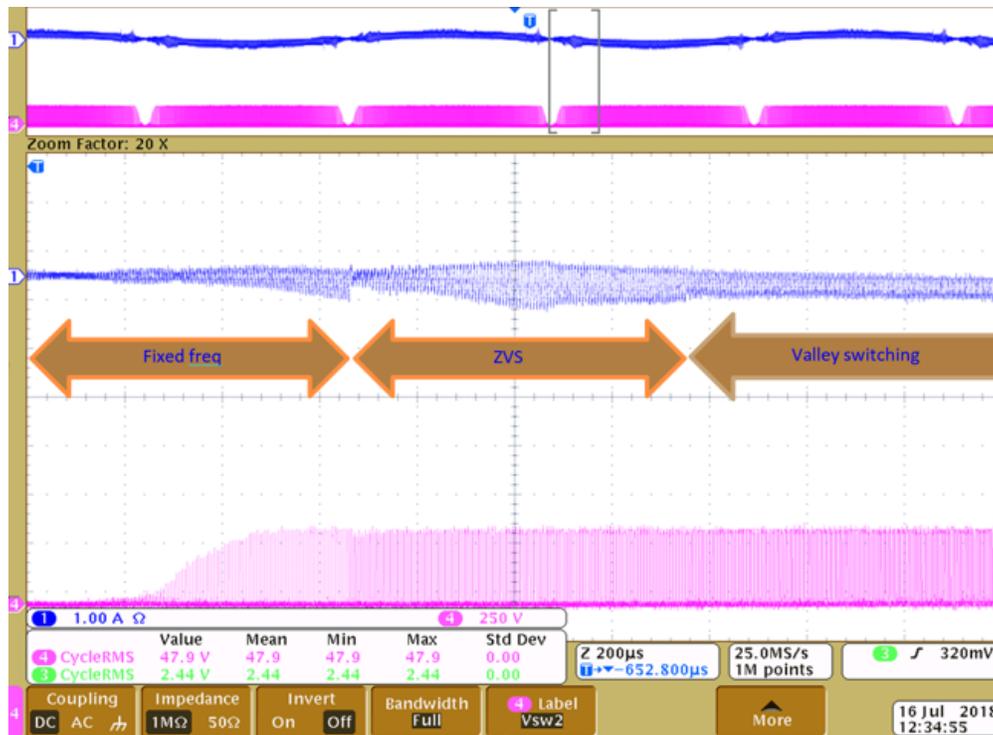
In 图 34 through 图 37, some test results are shown to verify the performance improvement after the enhanced valley switching control solution is implemented. 图 34 and 图 36 show the THD and efficiency under low line condition with traditional constant frequency control (150 kHz) and presented enhanced valley switching control solution. Based on the results, the THD at some point is 15% lower compared to the constant frequency control by adding optimized valley switching (valley switching + ZVS + fixed freq). The THD is close to 5 under 5% load (37.5 W/750 W). The efficiency and THD are improved for both high line and low line conditions based on the 图 34 through 图 37.

图 **34. 120 Vin, 380 Vout: THD**

图 **35. 220 Vin, 380 Vout: THD**

图 **36. 120 Vin, 380 Vout: Efficiency**

图 **37. 220 Vin, 380 Vout: Efficiency**

4 Design Files

This TI hardware design is based on modifications of the released reference design: C2000 2-phase Interleaved Power Factor Correction (ILPFC) Kit (TIDM-2PHILPFC). For more information on the hardware reference design, see C2000 2-Phase Interleaved Power Factor Correction (ILPFC) Kit With Integrated Power Metering (TIDM-2PHILPFC).

5 Software

To download the software files, see the C2000WARE-DIGITALPOWER-SDK.

6 Related Documentation

1. Texas Instruments, *TMS320F28004x Piccolo™ Microcontrollers Data Manual*
2. Texas Instruments, *Light Load THD and Efficiency Optimization of Digitally Controlled PFC Converter with Integrated Valley Switching Control Application Report*
3. Texas Instruments, *C2000™ Software Frequency Response Analyzer (SFRA) Library and Compensation Designer User's Guide*
4. Texas Instruments, *Piccolo F280049C controlCARD Information Guide*
5. Texas Instruments, *TMS320F28004x Piccolo Microcontrollers Technical Reference Manual*
6. Texas Instruments, *Two-Phase Interleaved PFC Converter w/ Power Metering Test Results Design Guide*

6.1 商标

Piccolo, E2E, C2000, 提供 powerSUITE, Code Composer Studio are trademarks of Texas Instruments. All other trademarks are the property of their respective owners.

7 About the Author

Chen Jiang is a Systems Application Engineer with C2000 Microcontrollers System Solutions Group at Texas Instruments, where he is responsible for developing reference design solutions for digital power applications. Before joining TI in 2017, Chen received his PhD in Electrical and Computer Engineering from Georgia Institute of Technology, Atlanta in 2017 and his Bachelor of Engineering from Zhejiang University, China in 2012.

修订历史记录

注：之前版本的页码可能与当前版本有所不同。

重要声明和免责声明

TI"按原样"提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做出任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 TI 的销售条款 或 ti.com 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。