1 Description
This TI Design implements a digitally-controlled, 500-W, two-phase interleaved LLC resonant converter. The system is controlled by a single, C2000 microcontroller (MCU), TMS320F28379, which also generates pulse width modulation (PWM) waveforms for all power electronic switching devices under all operating modes. This design implements an innovative current sharing technique to accurately achieve current balancing between phases.

2 Resources
TIDM-1001 Design Folder
controlSUITE™ Tools Folder
TMS320F28379 Product Folder
UCC27524 Product Folder
OPA365 Product Folder
UCD7138 Product Folder

3 Features
• Digitally-Controlled Two-Phase Interleaved LLC Resonant DC-DC Converter
• 500-W (250 W per Phase) Full-Load Power or 42.5-A Full-Load Current
• Vin: 370-V to 410-V DC
• Vout: 12-V DC
• Resonant Frequency: 250 kHz, Switching Frequency Range: 200 kHz to 350 kHz
• Peak Efficiency: 94.5%, Efficiency > 90% for all Loads Above 10% of Rated Load
• Excellent Current Sharing Between Phases Without Any Additional Hardware
• Phase Shedding With Programmable Limits
• Fault Protection: Phase and Output Overcurrent, Output Overvoltage, and Input Undervoltage and Overvoltage
• Supported by Various C2000 powerSUITE Tools

4 Applications
• Server Power Supplies
• Telecom Rectifiers
• Automotive Charging
• Industrial Power Supplies

An IMPORTANT NOTICE at the end of this TI reference design addresses authorized use, intellectual property matters and other important disclaimers and information.
5 System Overview

5.1 System Description

Resonant converters are popular DC-DC converters frequently used in server, telecom, automotive, industrial, and other power supply applications. The converters are a good choice for medium- to high-power applications because of their adherence to improving industry standards, ever-increasing power-density goals, and high-performance (efficiency, power density, and so forth) standards.

These are variable-frequency converters where the PWM-switching frequency of operation frequently changes during runtime. For reliable operation the changing frequencies must not produce any glitches or irregular PWM behavior. For applications with high-output currents that require input-output isolation, it is a common practice to use synchronous rectification (SR) on the secondary of the isolation transformer. SR uses additional power electronic devices switching with changing frequencies. High-power applications may require use of multiphase interleaved converters. These interleaved converters have even more devices switching with variable frequencies and additionally require fixed-phase relationships between various phases under all operating frequencies. Guaranteeing correct PWM waveform generation with changing frequencies under all operating conditions is a big challenge for the controller. Furthermore, interleaving multiple phases of resonant converters presents current sharing challenges between phases. Inadequate or improperly implemented current sharing or incorrect PWM waveform generation can lead to converter failure, significant system or component damage, and, in the worst case, significant property damage or resultant bodily injury or loss of life.

In server and telecom power supply applications, these converters are used to work as the isolated DC-DC converter stage in the rectifier system. These converters provide high efficiency and power density through soft-switching, SR, and other techniques.

These converters are gaining popularity in automotive on-board charging applications. Additionally, these devices may be used as isolated, bi-directional converters in electric vehicles (EVs) and hybrid electric vehicle (HEVs).

This design implements a 500-W, two-phase, interleaved half-bridge (HB) LLC resonant converter with SR on the secondary. The system is controlled by a single C2000 MCU, TMS320F28379, which also generates correct PWM waveforms for all power electronic switching devices (MOSFETs) using the latest features on C2000 MCUs. An innovative current sharing technique is implemented by the C2000 MCU to accurately achieve current and phase balancing for multiphase interleaved converters.

The accompanying software allows programming the controller and experimenting with different control parameters to tune the control loop for good system performance. This design supports the use of C2000 powerSUITE tools like the compensation designer, the software frequency response analyzer, and the solution adapter. The software project allows users to evaluate the complete system with the help of these supported tools. This document provides the hardware and software design details along with the test results. This document also describes a structured step-by-step method to evaluate this solution by starting with a simple open-loop excitation and then working towards a complete well-tuned closed-loop system.
### 5.2 Key System Specifications

#### Table 1. TIDM-1001 Performance Specifications

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TEST CONDITIONS</th>
<th>MINIMUM</th>
<th>TYP</th>
<th>MAXIMUM</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INPUT CHARACTERISTICS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voltage range</td>
<td>—</td>
<td>370</td>
<td>390</td>
<td>410</td>
<td>VDC</td>
</tr>
<tr>
<td>Input undervoltage threshold</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>VDC</td>
</tr>
<tr>
<td>Input overvoltage threshold</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>410</td>
<td>VDC</td>
</tr>
<tr>
<td>Input current</td>
<td>Vin = 370 V&lt;sub&gt;DC&lt;/sub&gt;, full load = 42.5 A</td>
<td>—</td>
<td>—</td>
<td>1.55</td>
<td>A</td>
</tr>
<tr>
<td>Input current</td>
<td>Vin = 390 V&lt;sub&gt;DC&lt;/sub&gt;, full load = 42.5 A</td>
<td>—</td>
<td>—</td>
<td>1.5</td>
<td>A</td>
</tr>
<tr>
<td>Input current</td>
<td>Vin = 410 V&lt;sub&gt;DC&lt;/sub&gt;, full load = 42.5 A</td>
<td>—</td>
<td>—</td>
<td>1.5</td>
<td>A</td>
</tr>
<tr>
<td>Phase overcurrent threshold</td>
<td></td>
<td>—</td>
<td>—</td>
<td>3.2</td>
<td>A</td>
</tr>
<tr>
<td><strong>OUTPUT CHARACTERISTICS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output voltage, V&lt;sub&gt;out&lt;/sub&gt;</td>
<td>No load to full load</td>
<td>—</td>
<td>12</td>
<td>—</td>
<td>VDC</td>
</tr>
<tr>
<td>Output overvoltage threshold</td>
<td></td>
<td>—</td>
<td>13.5</td>
<td>—</td>
<td>VDC</td>
</tr>
<tr>
<td>Load regulation</td>
<td>V&lt;sub&gt;in&lt;/sub&gt; = 410 V&lt;sub&gt;DC&lt;/sub&gt;, I&lt;sub&gt;out&lt;/sub&gt; = 2.5 A to 42.5 A</td>
<td>—</td>
<td>50</td>
<td>—</td>
<td>mV</td>
</tr>
<tr>
<td>Line regulation</td>
<td>V&lt;sub&gt;in&lt;/sub&gt; = 370 V&lt;sub&gt;DC&lt;/sub&gt; to 410 V&lt;sub&gt;DC&lt;/sub&gt;, I&lt;sub&gt;out&lt;/sub&gt; = 10 A</td>
<td>—</td>
<td>&lt; 5</td>
<td>—</td>
<td>mV</td>
</tr>
<tr>
<td>Output load current, I&lt;sub&gt;out&lt;/sub&gt;</td>
<td></td>
<td>—</td>
<td>—</td>
<td>42.5</td>
<td>A</td>
</tr>
<tr>
<td>Output overcurrent threshold</td>
<td></td>
<td>—</td>
<td>—</td>
<td>55</td>
<td>A</td>
</tr>
<tr>
<td>Output power per phase</td>
<td></td>
<td>—</td>
<td>—</td>
<td>250</td>
<td>W</td>
</tr>
<tr>
<td><strong>SYSTEM FEATURES</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Peak efficiency</td>
<td>V&lt;sub&gt;in&lt;/sub&gt; = 390 V&lt;sub&gt;DC&lt;/sub&gt;, I&lt;sub&gt;out&lt;/sub&gt; = 10 A</td>
<td>—</td>
<td>94.5</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Efficiency</td>
<td>V&lt;sub&gt;in&lt;/sub&gt; = 370 V&lt;sub&gt;DC&lt;/sub&gt; to 410 V&lt;sub&gt;DC&lt;/sub&gt;, 4.2 A &lt; I&lt;sub&gt;out&lt;/sub&gt; &lt; 42.5 A</td>
<td>—</td>
<td>&gt; 90</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Full-load efficiency</td>
<td>V&lt;sub&gt;in&lt;/sub&gt; = 370 V&lt;sub&gt;DC&lt;/sub&gt;, I&lt;sub&gt;out&lt;/sub&gt; = 42.5 A</td>
<td>—</td>
<td>92</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Resonant frequency</td>
<td></td>
<td>—</td>
<td>—</td>
<td>250</td>
<td>kHz</td>
</tr>
<tr>
<td>Switching frequency</td>
<td></td>
<td>—</td>
<td>200</td>
<td>350</td>
<td>kHz</td>
</tr>
</tbody>
</table>

<sup>(1)</sup> These default overcurrent limits are programmable and may be adjusted to allow high-load operation under noisy test conditions or to limit these currents to lower levels.

<sup>(2)</sup> Note that the resonant tank inductors are the hottest components on the board. Use an external air cooling fan (CFM rating > 50) directed at the board when operating with loads (I<sub>out</sub>) of 30 A or higher.
5.3 Block Diagram

![Block Diagram Image]

Figure 1. TIDM-1001 System Block Diagram

5.4 Highlighted Products

5.4.1 TMS320F28379

The Delfino™ TMS320F28379xD is a powerful, 32-bit, floating-point MCU designed for advanced closed-loop control applications, such as industrial drives and servo motor control, solar inverters and converters, digital power, transportation, and power line communications.

These devices feature integrated performance analog and control peripherals along with various communication peripherals to enable system consolidation. Four independent 16-bit analog-to-digital converters (ADCs) provide precise and efficient management of multiple analog signals, which ultimately boosts system throughput. The comparator subsystem (CMPSS) with windowed comparators allows for protection of power stages when current limit conditions are exceeded or not met. Other analog and control peripherals include DACs, PWMs, eCAPs, eQEPs, and other peripherals. Latest features on the PWM, CMPSS, and ADC peripherals are extensively used in this TI Design.

5.4.2 UCD7138

The UCD7138 device is a high-performance, 4-A and 6-A, single-channel MOSFET driver with body-diode conduction sensing and reporting. In this design the device is used to achieve advanced SR control without a need for typically lossy and expensive current sensing circuit. The device contains a high-speed gate driver, a body-diode conduction-sensing circuit, and a turnon delay optimization circuit. The device is suitable for high-power, high-efficiency isolated converter applications where SR dead-time optimization is desired. The benefits of the chipset include maximizing system efficiency by minimizing body-diode conduction time, robust and fast negative-current protection, and a simple interface.
The UCD7138 device offers asymmetrical rail-to-rail, 4-A source and 6-A sink peak-current drive capability. The short propagation delay and fast rise and fall time allows efficient operation at high frequencies. The UCD7138 device is capable of sensing body-diode conduction time as low as 10 ns.

### 5.4.3 UCC27524

The UCC2752x family of devices are dual-channel, high-speed, low-side gate-driver devices capable of effectively driving MOSFET and IGBT power switches. Using a design that inherently minimizes shoot-through current, UCC2752x can deliver high-peak current pulses of up to 5-A source and 5-A sink into capacitive loads along with rail-to-rail drive capability and extremely small propagation delay (typically 13 ns). In addition, the drivers feature matched internal propagation delays between the two channels. These delays are very well suited for applications requiring dual-gate drives with critical timing, such as the high-side and low-side switches of a HB in this design. This delay also enables connecting two channels in parallel to effectively increase current-drive capability or driving two switches in parallel with one input signal. Wide hysteresis between the high and low thresholds offers excellent noise immunity.

### 5.5 Functional Description

Resonant power converters contain resonant L-C networks whose voltage and current waveforms vary in a sinusoidal pattern during one or more sub-intervals of each switching period. Figure 2 shows a generic block diagram of a HB LLC resonant converter system. The resonant tank circuit is formed by external inductor $L_r$, transformer leakage inductance $L_k$, transformer magnetizing inductance $L_m$, and capacitor $C_r$. In some cases it is possible to integrate $L_r$ into the transformer design.

![Figure 2. Block Diagram of HB LLC Converter](image)

HB switches Q1 and Q2 are driven with 50% (typical) duty cycle signals that are frequency modulated. As a result the voltage at Q2 drain is a square wave voltage with changing fundamental frequency. The tank circuit acts as a band-pass filter that filters out the high-frequency and low-frequency components from this voltage and leaves the dominant fundamental frequency component. This component is then rectified on the secondary with either diode rectification or SR. A higher tank circuit current at the fundamental frequency implies higher energy that is transferred to the secondary or load. Frequency modulation of the PWM signals driving the HB switches consequently changes the tank network impedance resulting in a higher or lower tank circuit current depending on the fundamental frequency. This allows output voltage to be regulated by simply modulating the PWM frequency. [1]
5.5.1 The Resonant Tank

There are two resonant frequencies associated with this converter. The lower frequency, \( f_1 \), can be calculated as Equation 1.

\[
f_1 = \frac{1}{2\pi \sqrt{(L_r + L_k + L_m)} C_r}
\]  

The second frequency, \( f_2 \), can be calculated as Equation 2.

\[
f_2 = \frac{1}{2\pi \sqrt{(L_r + L_k)} C_r}
\]  

This frequency, \( f_2 \), is often referred to as the resonant frequency of the converter. The ratio \( (L_r+L_k)/L_m \) is an important parameter for the designer [2].

As mentioned above, frequency modulation changes the tank network input impedance. For frequencies above \( f_2 \), this impedance is inductive, while for frequencies below \( f_1 \), it is capacitive. For frequencies between these two resonant frequencies the impedance is inductive above a critical frequency based on the load resistance.

The LLC resonant converter is normally operated in the region where the tank impedance is inductive. This means that the impedance increases with frequency. As a result a properly regulated converter operates at low frequencies under high loads and at higher frequencies under low loads.

A typical gain vs frequency curve for the LLC HB converter is shown in Figure 3. For the purposes of this discussion the combined inductance \( (L_r+L_k) \) will be referred to as \( L_r \). As seen in Figure 4, Figure 5, Figure 6, and Figure 7, the \( L_r/L_m \) ratio plays an important role in deciding the range of frequencies over which the converter should operate [3].

![Figure 3. Typical Gain Versus Frequency](image-url)
With a smaller $L_r/L_m$ ratio, this range is larger. However, the converter may operate at frequencies relatively farther away from the resonant frequency $f_{r2}$. Designs with a higher $L_r/L_m$ ratio provide a larger change in gain for a small change in frequency. As a result the converter operates over a relatively smaller range of frequencies and fairly close to the resonant frequency $f_{r2}$ under all operating conditions.

![Figure 4. $L_r/L_m = 0.1$](image1)

![Figure 5. $L_r/L_m = 0.35$](image2)

![Figure 6. $L_r/L_m = 0.6$](image3)

![Figure 7. $L_r/L_m = 0.85$](image4)
5.5.2 Interleaving Phases

Figure 8 shows a generic block diagram of a two-phase interleaved HB LLC resonant converter system. Complementary PWM signals drive the high-side and low-side switches for each HB. All PWM signals shown in Figure 8 operate at 50% duty cycle with some dead-time between the turn ON and turn OFF of PWM signals driving switches in the same HB.

Figure 8. Block Diagram of Two-Phase Interleaved HB LLC Converter
5.5.2.1 Current Sharing

When two or more identical HB LLC converters are interleaved, any differences in their tank circuits lead to unequal sharing of the load current between individual phases. Unequal load sharing is a major problem in interleaving resonant converters as it decreases system efficiency, reliability, and thermal stability and can lead to high-circulating currents and even converter failure with significant system damage. As even small differences due to component tolerances may lead to considerable imbalances between phases, all interleaved resonant converters must have a way of sharing and balancing current between individual phases to allow safe, reliable, and efficient operation.

One solution is to design the tank circuit with extremely tight tolerances; however, this design ends up increasing the system cost considerably. Moreover, as mentioned above, even the slightest of differences due to tolerances may still lead to imbalances between phases. Although this solution may work for some applications, this solution is impractical on its own for most systems.

Some solutions try to match the tank circuits in different phases by adding more reactive components to the tank circuits in all or some of the phases[4]. In addition to increasing system cost and size, this solution may also increase manufacturing costs if further trimming is required on the assembly line.

More recent proposals take this approach a step further by trying to match the tank circuits during operation. Matching circuits is done by adding more power electronic switches to switch additional inductance or capacitance in and out of the tank circuit for some or all of the phases[5], which provides a good way of precisely adjusting the tank circuits at the expense of increased system cost, size, complexity, and a possible reduction in system efficiency.

There are other proposals that add extra converters based on an additional secondary winding on some or all of the LLC transformers. The operation of this additional converter is then controlled in a way to compensate for extra current being carried by the other phases[6]. This method suffers the same drawbacks as those discussed above and additionally increases complexity in the transformer design.

This design uses a new current-sharing technique that is implemented in software without a need for any additional external components or circuits. This new design is made possible by the C2000 MCU’s highly-configurable PWM modules. In this implementation the PWM duty cycle for switches in the phase carrying a higher load current is decreased appropriately by a current-balancing loop in the software. The controller also adjusts PWM timing for corresponding synchronous rectifier (SR) switches in that phase.
### 5.5.2.2 Phase Shift Between Phases

When two identical converter stages are interleaved, the PWM signals driving switches in the second phase are phase shifted by 180° with respect to the signals driving the corresponding switches in the first phase. For a three-phase, interleaved converter the phase shift between signals driving corresponding switches in the three phases is 120°. For a four-phase interleaved converter this phase shift is 90° and so on. Phase shifting is a well-known method that provides reduced input and output current ripples, reduced bus capacitances, and better EMI and EMC performance. For two-phase, interleaved resonant converters, maintaining this phase shift between phases under all operational frequencies possesses a big challenge for the PWM controller.

When the duty cycle of PWM signals driving the switches in the two phases is identical, a 180° phase shift between the two phases can be achieved by maintaining 180° phase shift between any similar points in the PWM cycle of corresponding switches in the two phases, for example the low-to-high PWM transitions, high-to-low PWM transitions, the mid-point of the ON time, and so forth. Typically this shift is done between the mid-points of the ON time of corresponding switches in the two phases, which is shown in Figure 9. A PWM period of 4 µs with a 50% duty cycle is assumed. Although dead-time is ignored here (for simplicity), this discussion is valid for the practical case of non-zero dead-time.

![Figure 9. Identical Duty Cycles for Two Phases](image-url)
With the current sharing technique employed here, one of the two phases, such as phase two, operates at a lower duty cycle than the other. As a result of a lower duty cycle, maintaining a 180° phase shift between the mid-point of the ON time of corresponding switches does not provide a 180° phase shift between high-to-low PWM transitions, which is shown in Figure 10.

![Figure 10. Different Duty Cycles for Two Phases](image)

LLC resonant converters are operated in the inductive region of the gain versus frequency curve of the tank circuit. As a result to maintain correct interleaving between phases with unequal duty cycles, a 180° phase shift should be maintained between the trailing edges (high-to-low transitions) of the PWM signals driving corresponding switches in the two phases. This phase shifting scheme is used in this design, which is shown in Figure 11. It is easy to infer that Figure 11 will look identical to Figure 9 when the same duty value is used for both phases.

![Figure 11. Correct PWM Waveforms for Two Phases Regardless of Individual Duty Cycles](image)
5.6 TIDM-1001 Implementation

Figure 12 shows a simplified block diagram of the circuit implemented on the TIDM-1001 two-phase, interleaved LLC resonant converter design. Components L1, T1, and C21 form the resonant tank in the first phase while L2, T3, and C38 form the resonant tank in the second phase. MOSFET switches Q3 and Q4 form the HB for the first phase. Q5 and Q6 are used as SR switches for this phase. MOSFET switches Q7 and Q8 form the HB for the second phase. Q9 and Q10 are used as SR switches for this phase.

Controlling this system in different operation modes requires generating complex PWM drive waveforms along with fast and efficient control-loop calculations. This is made possible on C2000 MCUs by advanced on-chip control peripherals like PWM modules, analog comparators with on-chip 12-bit DACs, and 12-bit high-speed ADCs coupled with an efficient 32-bit CPU.

Controller

This converter is controlled by a single C2000 MCU from Texas Instruments, TMS320F28379. The controller is referenced to the low voltage ground. Placing the controller on the output side of the isolation avoids any requirement for isolation in the output voltage feedback path and also possibly in the downstream communication (PMBUS, CAN, SPI, and so forth) with other modules or subsystems in the application. However, this choice necessitates isolation for the PWM signals driving the HB MOSFET switches on the high voltage side of the isolation. On this design, dual MOSFET drivers, UCC27524, from Texas Instruments along with gate drive transformers, 56PR3362, from Vitec are used for this purpose.
5.6.2 Hybrid Duty Control Mode

As this is a frequency controlled converter, the control algorithm controls the PWM switching frequency to regulate the output voltage. However, under certain operating conditions, the controller enters a new hybrid duty control mode where the duty cycle is also modulated. In this mode the switching frequency is still the controlled parameter for the voltage loop. The system enters this operating mode when it can no longer adequately regulate the output voltage with frequency control alone. This mode change is initiated when the software evaluates that the output voltage is higher than the desired voltage by a set limit, while operating at the maximum possible operating frequency.

5.6.3 Resonant Tank

The resonant tank is designed for a resonant frequency of 250 kHz. The external inductors L1 and L2 are 62.1 µH inductors, 75PR8126, from Vitec. Vitec transformers 75PR8125 are used as the resonant tank transformers T1 and T3. A 6.2 nF capacitor is used as the resonant tank capacitor. The transformer has a turns ratio, n, of 17 and is designed with a magnetizing inductance, \( L_m \), of 149.2 µH and a leakage inductance, \( L_k \), of 3.2 µH. These values provide a \( (L_r+L_k)/L_m \) ratio of approximately 0.44 and a resonant frequency, \( f_{r2} \), of approximately 250.13 kHz.

5.6.4 HB PWM Drivers

The C2000 MCU generates PWM waveforms for all four HB switches and the four SR switches. Complementary PWM signals drive the high-side and low-side switches for each HB. As explained in the previous section, when two phases are interleaved, PWM signals in the two phases operate 180° out of phase with respect to each other. All PWM signals operate at 50% duty cycle with some dead-time between the turn ON and turn OFF of PWM signals driving switches in the same HB. The only exceptions to this rule of 50% duty cycle operation are either when the current sharing algorithm forces one of the phases to operate at lower duty cycles or when the converter operates in hybrid duty control mode.

5.6.5 SR PWM Drivers

For reliable and efficient operation of this power converter, the SR turn-on and turn-off instants have to be accurately controlled relative to the PWM switching cycle and current in the transformer secondary winding. The C2000 MCU generates PWM signals for the SR switches relative to the signals driving the HB switches. A single-channel SR driver, UCD7138, from Texas Instruments detects body-diode conduction of the SR switch and uses this to turn the switch ON at the optimum point in time. The UCD7138 driver achieves this by accurately delaying the rising edge (turn ON) of the PWM signal, which comes from the C2000 MCU, to a point in time where the body-diode starts conducting.

Four of these drivers are used to drive the four SR switches. This approach brings a huge performance or cost benefit to the system as no other type of additional, and usually dissipative, current or voltage sensing is required to determine this accurate turn-on instant for the SR switches. The SR turnoff instant is controlled by the C2000 MCU.

5.6.6 Phase Shedding and Disabling SR Under Low-Load Operation

Multiphase converters use multiple identical converter phases to achieve a higher power rating. The resultant system has a significantly larger number of switching devices, which switch at high frequencies than a single converter phase. As a result the switching power losses for these converters are the sum total of switching losses in each of the converter phases. While all phases must contribute and share the load to support operations at high load, it may be possible for a few phases to handle operations at lower loads if the other converter phases are disabled. Disabling some of the converter phases is quite advantageous for improving converter efficiency as this eliminates switching losses in those disabled converter phases. As the system is operating under lower loads, any savings in power losses contributes considerably in increasing the overall system efficiency.

This design implements phase shedding by disabling one of the converter phases when the output current \( (I_{out}) \) falls below a programmable threshold (default approximately 10 A). This phase is re-enabled once the output current goes back up above a higher programmable threshold (default approximately 12 A). These two output current thresholds must provide some hysteresis (here 2 A) to avoid the system from constantly going in and out of phase shedding mode. When both phases are active and the controller software determines that the current has fallen down below the phase shedding threshold, it initiates a
gradual shut-down of one of the two phases. The system continues to operate normally during this time. As soon as the controller software determines that the output current has risen above the higher phase shedding threshold, it quickly re-enables the phase that is currently disabled. Unlike disabling of the converter phase when phase shedding mode is entered, the re-enabling of a phase when phase shedding mode is exited is not gradual. Both phases are active once the system exits phase shedding mode. The phase that was active during the previous phase shedding operation is marked to be shed (disabled) when the system enters phase shedding mode again. The phase that was shed during the previous phase shedding operation is active during the next phase shedding mode operation.

Under very low load operation SR switching losses can be greater than the power savings obtained by SR. In this case the SRs may be disabled and only their body diodes used for rectification. This mode is used when the output current falls below approximately 2 A. The system exits this mode when the output current increases above approximately 3 A.

5.6.7 Feedback Signals

For proper control of the power stage under all operating conditions, the controller needs feedback information about the output voltage \(V_{\text{out}}\), the resonant tank currents \(I_{\text{pri-1}}, I_{\text{pri-2}}\), and the output current \(I_{\text{out}}\).

\(V_{\text{out}}\), used for the voltage control loop, is fed back to the on-chip 12-bit ADC on the MCU by way of simple resistive voltage dividers.

A 0.5 m\(\Omega\) equivalent shunt resistor is used to sense \(I_{\text{out}}\). This signal is amplified by a low-noise, high CMRR, op-amp, OPA365, from Texas Instruments and sent to the ADC and an on-chip comparator on the MCU. \(I_{\text{out}}\) information is used for determining different operating modes and for output over-current protection.

The resonant tank currents, which are also the transformer primary currents, are each sensed using a 25 turn current sense transformer, WCM 603-7, from West Coast Magnetics and sent to the ADC and on-chip comparators. The difference between the two resonant tank currents is compared against a programmable set threshold that determines how well the currents in the two phases are shared. This threshold is user programmable. \(I_{\text{pri-1}}\) and \(I_{\text{pri-2}}\) are also used for overcurrent protection.

5.6.8 Fault Protection

At this stage, it is appropriate to introduce the shutdown mechanism used with this project. Here overcurrent protection is implemented for the transformer high voltage winding current for each phase using on-chip analog comparator subsystems CMPSS1 and CMPSS2. Output overcurrent protection is implemented for the transformer high voltage winding current for each phase using on-chip analog comparator sub-system CMPSS5. The reference trip levels are set using the corresponding internal 12-bit DACs, which are fed to the inverting terminals of these comparators. The comparator outputs are configured to generate a one-shot trip action on ePWM1, ePWM2, ePWM4, and ePWM5 whenever the sensed current is greater than the set limit. Output overvoltage protection is implemented in the software inside the state-machine task A1. Whenever an output overvoltage is detected, a one-shot trip action is initiated on ePWM1, ePWM2, ePWM4, and ePWM5.

The flexibility of the trip mechanism on C2000 devices provides the possibilities for taking different actions on different trip events. In this project ePWM1A, ePWM1B, ePWM2A, ePWM2B, ePWM4A, ePWM4B, ePWM5A, and ePWM5B outputs are driven low immediately to protect the power stage. These outputs are held in this state until a device reset is executed. The software also sets the system fault LED under these fault conditions.

Input undervoltage and overvoltage protection are implemented in the software inside the slower state-machine task C2. These conditions disable LLC operation and all PWMs are held in low state. LLC operation and the PWMs are re-enabled once the input comes back within programmed limits and stays good for a programmable time interval.
6 Getting Started Hardware and Software

6.1 Hardware and Resources Guide

Figure 13 shows some of the key components on the actual hardware.

Figure 13. TIDM-1001 – Two-Phase Interleaved LLC Resonant Converter Board
Table 2 shows the key signal connections between the TMS320F28379D controlCARD and the TIDM-1001 base board. For reference relevant portions of the schematic are also provided in Figure 14 to Figure 18.

Table 2. Key Signal Connections

<table>
<thead>
<tr>
<th>SIGNAL NAME</th>
<th>DESCRIPTION</th>
<th>CONNECTION TO controlCARD</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPWM-1A</td>
<td>High-side drive signal for phase one HB (Ph1_PWMHS)</td>
<td>GPIO-00</td>
</tr>
<tr>
<td>EPWM-1B</td>
<td>Low-side drive signal for phase one HB (Ph1_PWMLS)</td>
<td>GPIO-01</td>
</tr>
<tr>
<td>EPWM-2A</td>
<td>Drive signal for phase one SRA (Ph1_PWMSRA)</td>
<td>GPIO-02</td>
</tr>
<tr>
<td>EPWM-2B</td>
<td>Drive signal for phase one SRB (Ph1_PWMSRB)</td>
<td>GPIO-03</td>
</tr>
<tr>
<td>EPWM-4A</td>
<td>High-side drive signal for phase two HB (Ph2_PWMSRB)</td>
<td>GPIO-06</td>
</tr>
<tr>
<td>EPWM-4B</td>
<td>Low-side drive signal for phase two HB (Ph2_PWMLS)</td>
<td>GPIO-07</td>
</tr>
<tr>
<td>EPWM-5A</td>
<td>Drive signal for phase two SRA (Ph2_PWMSRA)</td>
<td>GPIO-08</td>
</tr>
<tr>
<td>EPWM-5B</td>
<td>Drive signal for phase two SRB (Ph2_PWMSRB)</td>
<td>GPIO-09</td>
</tr>
<tr>
<td>Fault</td>
<td>Drive signal for system fault LED</td>
<td>GPIO-16</td>
</tr>
<tr>
<td>System_Good</td>
<td>Drive signal for system good LED</td>
<td>GPIO-17</td>
</tr>
<tr>
<td>V_{out fb}</td>
<td>Output voltage feedback</td>
<td>ADC-C2</td>
</tr>
<tr>
<td>I_{out fb}</td>
<td>Output current feedback</td>
<td>ADC-C4/CMPSS5</td>
</tr>
<tr>
<td>I_{out filt}</td>
<td>Heavily filtered output current feedback</td>
<td>ADC-B0</td>
</tr>
<tr>
<td>I_{pri1 fb}</td>
<td>Phase one primary and resonant current feedback</td>
<td>ADC-A2/CMPSS1</td>
</tr>
<tr>
<td>I_{pri1 filt}</td>
<td>Heavily filtered phase one primary and resonant current feedback</td>
<td>ADC-A3</td>
</tr>
<tr>
<td>I_{pri2 fb}</td>
<td>Phase two primary and resonant current feedback</td>
<td>ADC-A4/CMPSS2</td>
</tr>
<tr>
<td>I_{pri2 filt}</td>
<td>Heavily filtered phase two primary and resonant current feedback</td>
<td>ADC-A5</td>
</tr>
<tr>
<td>V_{in}</td>
<td>Input voltage feedback</td>
<td>ADC-C0</td>
</tr>
</tbody>
</table>

This design uses a lot of jumper options for experimentation but there are some jumpers that must be populated for proper operation of the board. The jumpers that must be populated are listed below:

- J6
- J7
- J8
- J14 (positions 1 to 2, 3 to 4, 5 to 6, 7 to 8, 11 to 12, 13 to 14, 15 to 16, 17 to 18)
- J15 (for stand-alone or DEMO operation)
Figure 14. TIDM-1001 – HB PWM Drive Circuit
**Figure 15. TIDM-1001 – Phase-One SR PWM Driver Circuit**
Figure 16. TIDM-1001 – Phase-Two SR PWM Driver Circuit
Figure 17. TiDM-1001 – \( I_{\text{out}} \) and \( I_{\text{pri1}} \) Feedback Circuit
Figure 18. TIDM-1001 – Ipri2 and Vout Feedback Circuit
### 6.2 Software Overview

#### 6.2.1 Software Flow

The software project makes use of the C-background and C-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, both executed inside time critical interrupt service routines (ISRs), and runs all the critical control code. This code includes ADC reading, control calculations, and PWM updates. The Control ISR portion of the C-ISR is executed at a fixed rate of 50 kHz using a spare PWM module timer. The Fast Update ISR portion is triggered by the master switching PWM (phase one HB driver PWM), which is only executed when there is a change in frequency or when a system command requires a fast update. Figure 19 shows the general software flow for this project. Note the references to software frequency response analyzer (SFRA) library functions.

<table>
<thead>
<tr>
<th>C Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System Level Control/Management</strong></td>
</tr>
<tr>
<td><strong>Main</strong></td>
</tr>
<tr>
<td><strong>Initialization</strong></td>
</tr>
<tr>
<td>28x Device level</td>
</tr>
<tr>
<td>Peripheral level</td>
</tr>
<tr>
<td>System level</td>
</tr>
<tr>
<td>SFRA Init, ISR, ADC</td>
</tr>
<tr>
<td><strong>Background Loop</strong></td>
</tr>
<tr>
<td><strong>Background Loop</strong></td>
</tr>
</tbody>
</table>

| Control ISR |
| 50 kHz |
| **ISR** |
| SFRA_INJECT() |
| ADCDRV_1ch() |
| CNTL2P2Z() |
| **Current Sharing Control** |
| **Phase Shedding Control** |
| **Hybrid Duty Control** |
| **PWM Register Update** |
| SFRA_COLLECT() |
| **DLOG_1ch()** |

| Fast Update ISR |
| 200 kHz to 350 kHz |
| **PWM ISR** |
| **Global Synchronous Update** |
| **Fast Phase/SR/LLC Enable/Disable** |
| **Exit** |

**Figure 19. TIDM-1001 - Software Flow**
Texas Instruments’ SFRA library is designed to enable frequency response analysis on digitally controlled power converters using software alone, which 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, refer to the SFRA library documentation.

In addition to SFRA, this kit supports the use of other powerSUITE tools including the Compensation Designer and the Solution Adapter. These tools help users evaluate the complete system, adapt it for their end application, and tune it for improved performance. Figure 20 shows the typical process flow for designing and tuning such a system using the powerSUITE tools.

![Figure 20. Design Flow With powerSUITE](image)

The Solution Adapter tool allows users to adapt existing code examples from TI digital power application library and configure them to run on their custom digital power supply board that uses the same topology and similar resources. The GUI steps the user through the process of selecting the solution to adapt, selecting the relevant options for that solution, and customizing those options to adapt the software solution to the user's custom hardware design.

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. Note that the default software project uses a two-pole, two-zero control law and does not support three-pole, three-zero compensators.

The key framework C file used in this project is HV2PHILLLC-Main.c. This file is used to initialize, run, and manage the application. This file is the brains behind the application and contains all-time critical control type code in the ISR.

The Digital Power Library functions (modules) are called from this framework.
Table 3 shows the library modules used in the project.

### Table 3. Software Library Modules

<table>
<thead>
<tr>
<th>C CONFIGURE FUNCTION</th>
<th>C INITIALIZATION FUNCTION</th>
<th>C RUN-TIME FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>PWM_HB_LLC_2PHIL_SR</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>ADC_SOC_Cnf</td>
<td>—</td>
<td>ADCDRV_1ch_F_C</td>
</tr>
<tr>
<td>PWM_1ch_CNf</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>—</td>
<td>CNTL_2P2Z_F_C_COEFF_INIT</td>
<td>—</td>
</tr>
<tr>
<td>CNTL_2P2Z_F_C_VAR_INIT</td>
<td>CNTL_2P2Z_F_C</td>
<td>—</td>
</tr>
</tbody>
</table>

Although a two-pole, two-zero controller is used, the controller could very well be a PI, PID, a three-pole three-zero, or any other controller that can be suitably implemented for this application. This modular library structure makes it convenient to visualize and understand the complete system software flow. The structure also allows for easy use and for additions and deletions of various functionalities. This fact is amply demonstrated in this project by implementing an incremental build approach, which is discussed in more detail in the Section 6.2.2.

### 6.2.2 Incremental Builds

This project is divided into two incremental builds, which makes learning and getting familiar with the board and software easier. This approach is also good for debugging and testing boards.

Table 4 shows the incremental build options. To select a particular build option, select the corresponding project option in `main.cfg` as shown below. Once the build option is selected, compile the complete project by selecting `rebuild-all` compiler option. Section 6.2.3 provides more details to run each of the build options.

### Table 4. Incremental Build Options

<table>
<thead>
<tr>
<th>INCREMENTAL BUILD OPTIONS</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INCR_BUILD = 1</td>
<td>Open-loop check (check PWM drive circuit and sensing circuit) with SFRA</td>
</tr>
<tr>
<td>INCR_BUILD = 2</td>
<td>Closed voltage loop (check full system functionality) with SFRA</td>
</tr>
</tbody>
</table>
6.2.3 Procedure for Running the Incremental Builds

The main source files and the project file for C framework to bring up the system are located in the project directory. The software project included with this software is targeted for Code Composer Studio™ (CCS) version 6.1 and higher.

Follow the steps in the following subsections to build and run the example software.

**WARNING**

There are high voltages present on the board, which should only be handled by experienced power supply professionals in a lab environment. To safely evaluate this board, an appropriate isolated and current limited high voltage DC source should be used. Before DC power is applied to the board, a voltmeter and an appropriate resistive or electronic load must be connected to the output. The unit should never be handled when power is applied to it.

6.2.3.1 Build One: Open-Loop Control With SFRA

The objective of this build is to get familiar with the TIDM-1001 hardware and control the output voltage using direct PWM period adjustments without feedback. Because this system is running open loop, the ADC measured values are only used for instrumentation purposes in this build. The PWM period is adjusted using the Expressions Window. Optionally, SFRA GUI can be used during run time to get frequency response of the plant.

6.2.3.1.1 Overview

The software has been configured to adjust the period of the PWM outputs for the modules selected in main.cfg (for the TIDM-1001 board these are PWMs 1, 2, 4, and 5). Variable Period is used to correctly update PWM registers inside the function UpdateRegs. The PWM period command can be adjusted from the Expressions Window by using the variable Period_Set.

On-chip analog comparators (selected in main.cfg file) and corresponding DAC mechanisms are used to provide overcurrent protection (for the TIDM-1001 board these are CMPSSs 1, 2 and 5). The reference trip level for the comparators can be set using the Gui_Ipri1tripSet, Gui_Ipri2tripSet, and Gui_IoutripSet variables. The comparator output is configured to generate a one-shot trip action on the PWM modules, selected in main.cfg, whenever the sensed current is greater than the set limit. The flexibility of the trip mechanism on C2000 devices provides the possibility for taking different actions on different trip events. In this project all PWM outputs are driven low immediately on a comparator event to protect the power stage.

The converter is driven at a PWM switching frequency between 200 kHz and 350 kHz according to the Period_Set command. The DPL_ISR_wFRA ISR is triggered by the master PWM module (PWM1). This ISR is where the control code and digital power library modules are executed.

A task state-machine has been implemented as part of the background code. Tasks are arranged in groups (A1, A2, A3…, B1, B2, B3…, C1, C2, C3…). Each group is executed according to three CPU timers, which are configured with periods of 1 ms, 20 ms, and 50 ms respectively. Within each group (for example, B) each task is run in a round-robin manner. For example, group B executes every 20 ms, and there are three tasks in group B. Therefore, B1, B2, and B3 execute once every 60 ms.
6.2.3.1.2 **Procedure**

6.2.3.1.2.1 **Hardware Setup**

1. To get started with the TIDM-1001 board, the user must have a TMS320F28379D controlCARD with a mini-USB cable, a 12-V DC bench power supply, and a computer with CCS version 6.1 or newer, the latest version of controlSUITE, and CCS GUI composer installed.

2. Make sure that all jumpers on the TIDM-1001 board are correctly installed as listed in the Section 6.1.

3. Insert the controlCARD in the 180-pin connectors J3 and J4. Connect a current limited 12-V DC bench power supply between TP25 and TP26 with correct polarity. Connect an isolated, 400-V programmable DC power source to the 400-V input connector (J9 to J12) and 12-V load to the 12-V output connector (J10 and J11). Make sure that this load does not exceed the board ratings. Connect a USB B-to-A cable between the PC and the controlCARD. Do not turn ON any of the power supplies at this time.

   **NOTE:** While this system can operate under **No Load** condition, it is recommended to use a load \( I_{\text{out}} \) of at least 5 A for the open-loop tests.

   Use an external air cooling fan (CFM rating > 50) directed at the board when operating with loads \( I_{\text{out}} \) of 30 A and higher

4. Make sure position two of the controlCARD switch A:SW1 is in ON position in the SFRA setup step.

6.2.3.1.2.2 **Software Setup**

1. Open CCS (version 6.1 or newer). Maximize CCS to fill the screen. Close the welcome screen if it opens up.

2. A project contains all the files and build options required to develop an executable output file (.out), which can be run on the MCU hardware. On the menu bar click **Project → Import CCS Project**. Under the root directory, navigate to and select `..\controlSUITE\development_kits\TIDM_1001\v1_00_00_00\f2837x\HV2PHILLLC` directory. Make sure that under the **Projects** tab, HV2PHILLLC is checked. Click **Finish**.

3. HV2PHILLLC project should now appear in the CCS Project Explorer window. This project will invoke all the necessary tools (compiler, assembler, and linker) to build the project. A project contains all the files and build options required to develop an executable output file (.out), which can be run on the MCU hardware.
4. In the project window on the left, click the arrow sign to the left of the project name. The project window will look like Figure 21.

![Figure 21. Project Window](image)

5. Open `main.cfg` file by double-clicking on the file name in the project window.
6. Under *Project Options* select *Open Loop* as shown in Figure 22. Save *main.cfg* file. Note that the default software does not allow phase shedding in this build regardless of this selection in the *main.cfg*.

![Figure 22. Build One: main.cfg](image)

6.2.3.1.2.3 *Build and Load the Project*

1. Turn ON the 12-V auxiliary supply.
2. 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.
3. Click on the *Debug* button ( ) or click *Run → Debug*. The build one code should compile and load.
4. Notice the *CCS Debug* icon in the upper-right corner, which indicates the *Debug Perspective* view. The program should be stopped at the start of *main()*.
6.2.3.1.2.4 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 CCS, such as memory windows and watch windows. Additionally, CCS has the ability to make time (and frequency) domain plots, which allows the user to view waveforms using graph windows.

1. Populate the Expressions Window entries by clicking on View → Scripting console on the menu bar and then opening the AddWatchWindowVars.js file from the project directory using the scripting console Open File ( ) command. The Expressions Window should look like Figure 23.

![Figure 23. Build One: Expressions Window at Reset](image_url)
# Table 5. Description of Expressions Window Entries

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period</td>
<td>Period input to the PWM driver. This is the controlled parameter.</td>
</tr>
<tr>
<td></td>
<td>• In build one this is a slewed value based on Period_Set. If</td>
</tr>
<tr>
<td></td>
<td>used, SFRA signal is injected here.</td>
</tr>
<tr>
<td></td>
<td>• In build two this is the output of the voltage control loop.</td>
</tr>
<tr>
<td>Period_Set</td>
<td>Only used in build one.</td>
</tr>
<tr>
<td></td>
<td>This value is set by user in build one.</td>
</tr>
<tr>
<td>Gui_Vin</td>
<td>Input voltage in V. Only valid if jumper J15 is populated on the TIDM-1001 board.</td>
</tr>
<tr>
<td>Gui_Vset</td>
<td>Not used in build one.</td>
</tr>
<tr>
<td></td>
<td>Output voltage command in V (set by user).</td>
</tr>
<tr>
<td>Gui_Vout</td>
<td>Output voltage in V.</td>
</tr>
<tr>
<td>Gui_Iout</td>
<td>Output current in A.</td>
</tr>
<tr>
<td>Gui_Ipri1</td>
<td>Phase one resonant tank current in A.</td>
</tr>
<tr>
<td>Gui_Ipri2</td>
<td>Phase two resonant tank current in A.</td>
</tr>
<tr>
<td>Gui_Ipri1tripSet</td>
<td>Phase one resonant tank overcurrent shut down level in A.</td>
</tr>
<tr>
<td>Gui_Ipri2tripSet</td>
<td>Phase two resonant tank over-current shut down level in A.</td>
</tr>
<tr>
<td>Gui_louttripSet</td>
<td>Output overcurrent shut down level in A.</td>
</tr>
<tr>
<td>FaultFlg</td>
<td>Overcurrent and output over-voltage PWM shut down flag</td>
</tr>
<tr>
<td>start_flag</td>
<td>Not used in build one.</td>
</tr>
<tr>
<td></td>
<td>Used to start converter operation when the system is not in DEMO mode.</td>
</tr>
<tr>
<td></td>
<td>Software clears this flag when converter operation begins.</td>
</tr>
<tr>
<td>LLC_Enable</td>
<td>Not used in build one.</td>
</tr>
<tr>
<td></td>
<td>In DEMO mode, this flag is set (by software algorithm) and converter</td>
</tr>
<tr>
<td></td>
<td>operation is started when the 400V input stays within specified limits</td>
</tr>
<tr>
<td></td>
<td>of operation for a programmable time interval.</td>
</tr>
<tr>
<td></td>
<td>When the system is not in DEMO mode, this flag is set (by software</td>
</tr>
<tr>
<td></td>
<td>algorithm) and converter operation is started when the start_flag is</td>
</tr>
<tr>
<td></td>
<td>Set by the user during runtime.</td>
</tr>
<tr>
<td>Auto_Balancing</td>
<td>Enable or disable the current sharing feature</td>
</tr>
<tr>
<td>Shed_Phase</td>
<td>• When set to 1, this indicates that the system is currently</td>
</tr>
<tr>
<td></td>
<td>operating in phase shedding mode (that is, a phase is disabled).</td>
</tr>
<tr>
<td></td>
<td>• When set to 0, this indicates both phases are active.</td>
</tr>
<tr>
<td>Phase_Pointer</td>
<td>Points to the phase that was shed in the last phase shedding</td>
</tr>
<tr>
<td></td>
<td>operation. The other phase is disabled the next time the system enters</td>
</tr>
<tr>
<td></td>
<td>phase shedding mode.</td>
</tr>
<tr>
<td>auto_DB</td>
<td>When set, dead-band between high-side and low-side HB switches is</td>
</tr>
<tr>
<td></td>
<td>adjusted by the software algorithm based on operating point.</td>
</tr>
</tbody>
</table>
6.2.3.1.2.5 Using Real-Time Emulation

Real-time emulation is a special emulation feature that allows windows within CCS to be updated while the MCU is running. This feature not only allows graphs and watch windows to update, but also allows the user to change variables or memory location values and have those changes affect the MCU behavior. This feature is very useful when tuning control law parameters on-the-fly, for example.

1. Enable real-time mode by hovering your mouse on the buttons on the horizontal toolbar and clicking button.

2. A message box may appear. If so, select YES to enable debug events. This will set bit one (DGBM bit) of status register one (ST1) to 0. The DGBM is the debug enable mask bit. When the DGBM bit is set to 0, memory and register values can be passed to the host processor for updating the debugger windows.

3. When a large number of windows are open, as bandwidth over the emulation link is limited, updating too many windows and variables in continuous refresh can cause the refresh frequency to slow down. Right-click on the button in the Expressions Window and select Continuous Refresh Interval.... The refresh rate can be slowed for the Expressions Window variables by changing the Continuous refresh interval (milliseconds) value. A rate of 1000 ms is usually enough for these exercises. Click on Continuous Refresh buttons ( ) for the watch view.
6.2.3.1.2.6 Run the Code

1. Run the code by using the <F8> key or the Run button on the toolbar.
2. As this is an open-loop test (no voltage loop), care must be taken not to use too small a load to avoid accidental high output voltages. Use a load of 5A at the 12-V output.
3. Set the 400-V DC supply to output 300 V (this value is lower than the 370-V, low-end limit for the reason mentioned in the previous step). Set the power supply current limit to an appropriate level for this test. Now turn ON this 300-V power supply.
4. The converter operation should start with the default Period_Set value resulting in approximately 250-kHz switching frequency (resonant frequency).
5. The Period_Set value may be changed between 0.0 and 0.95. As this value is increased the switching frequency decreases, which results in a higher energy delivered to the load. In open loop this results in an increase in output voltage, which should not be allowed to exceed board capabilities. Period_Set has been restricted to a maximum value of 0.95 in this build.
6. Figure 24 shows the Expressions Window that corresponds to the operation of the system when Period_Set is 0.8.

![Figure 24. Build One: Expressions Window With Period_Set = 0.8](image)

7. If desired, the SFRA tool may now be used to look at the frequency response of the plant.
8. To do this click on main.cfg file and open SFRA. Click on setup connection and select the appropriate COM port. Make sure that the baud rate is set to 57600 and that Boot on Connect is unchecked. Click OK.

![Figure 25. Build One: SFRA Connections Setup](image)

9. Select Floating Point math.
10. Click Connect on the SFRA GUI. Once the GUI is connected, click on Start Sweep. SFRA will start applying different frequencies and collecting the response for frequency analysis.
11. Once the frequency sweep is complete, the response will be displayed on the SFRA GUI, as shown in Figure 26.

![SFRA GUI](image)

**Figure 26. Build One: SFRA Results**

12. Close the SFRA GUI.
13. Turn OFF the 300-V DC power supply.
14. Fully halting the MCU when in real-time mode is a two-step process. First, halt the processor by using the **Suspend** button ( ) on the toolbar or by using **Run → Suspend**. Click **button again to take the MCU out of real-time mode and then reset the MCU (**Run → Reset → CPU Reset**).
15. CCS may be left running for the next build or optionally close CCS.

End of exercise.
6.2.3.2 **Build Two: Closed-Loop Control With SFRA**

The objective of this build is to regulate the output voltage of the converter using closed-loop feedback control realized in the form of a software-coded control loop. A compensator designed using the Compensation Designer GUI is used to achieve desired closed loop-performance. SFRA GUI can be used during run time to capture the frequency response of the system.

### 6.2.3.2.1 Overview

The software has been configured to provide closed loop voltage control for the two-phase, interleaved LLC resonant converter power stage. A two-pole, two-zero controller module (CNTL_2P2Z) is used to implement the control law. The output voltage feedback \( Adc\_Vout \) is an input to this block. The reference input to the \( CNTL\_2P2Z \) block comes from the slewed output voltage command \( Vout\_ref\_wInj \). If used, the SFRA signal is applied here. The controller output Period is used to update PWM registers similar to build one. The output voltage command can be adjusted from the Expressions Window using the variable \( Gui\_VSet \).

Similar to build one, on-chip analog comparators (selected in main.cfg file) and corresponding DAC mechanisms are used to provide overcurrent protection (for the TIDM-1001 board these are CMPSSs 1, 2, and 5). The reference trip level for the comparators can be set using the \( Gui\_lpri1tripSet \), \( Gui\_lpri2tripSet \), and \( Gui\_louttripSet \) variables. The comparator output is configured to generate a one-shot trip action on the PWM modules, selected in main.cfg, whenever the sensed current is greater than the set limit. The flexibility of the trip mechanism on C2000 devices provides the possibility for taking different actions on different trip events. In this project all PWM outputs are driven low immediately on a comparator event to protect the power stage.

The converter is driven at a PWM switching frequency between 200 kHz and 350 kHz. The \( DPL\_ISR\_wFRA \) ISR is triggered by the master PWM module (PWM1). This ISR is where the control code and digital power library modules are executed.

Similar to build one, a task state-machine has been implemented as part of the background code. Tasks are arranged in groups (A1, A2, A3..., B1, B2, B3..., C1, C2, C3...). Each group is executed according to three CPU timers, which are configured with periods of 1 ms, 20 ms, and 50 ms respectively. Within each group (for example, \( B \)) each task is run in a round-robin manner. For example, group B executes every 20 ms, and there are three tasks in group B. Therefore, B1, B2, and B3 execute once every 60 ms.

### 6.2.3.2.2 Procedure

#### 6.2.3.2.2.1 Hardware Setup

1. Follow the steps in Section 6.2.3.1.2.1 for build one procedure.

   **NOTE:** While this system can operate under No Load condition, it is recommended to use a load \( (I_{out}) \) of at least 2 A for the closed-loop tests.

   Use an external air cooling fan (CFM rating > 50) directed at the board when operating with loads \( (I_{out}) \) of 30 A and higher.

2. Navigate to and select the following project from the controlSUITE directory `..\controlSUITE\development_kits\TIDM_1001\v1_00_00_00\f2837x\HV2PHILLLC` directory. Make sure that under the Projects tab, HV2PHILLLC is checked. Click Finish.

3. HV2PHILLLC project should now appear in the CCS Project Explorer window. This project will invoke all the necessary tools (compiler, assembler, and linker) to build the project. 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. In the project window on the left, click the arrow sign to the left of the project name.
5. Open main.cfg file by double-clicking on the file name in the project window.

6. Under Project Options, select Closed Voltage Loop as shown in Figure 27.

7. Without changing any of the default parameters, click on Compensation Designer.

8. The Compensation Designer GUI uses the SFRA data from the previous run of SFRA on this system (default data for this design is in the project folder) to plot the expected frequency response for the set of compensator coefficients selected in the main.cfg file (here COMP3). The Compensation Designer GUI allows the design of different compensators to achieve desired closed-loop performance. Compensation Designer GUI can also use power stage model data, if it exists, based on the data entered in the main.cfg file. For this project this option (modelled plant) is not available.

9. The default compensator (COMP3) is a two-pole, two-zero compensator, which has been provided by TI. Notice that the magnitude and phase plots, for the plant, open loop, and the compensator update whenever a new compensator is selected from the list of possible compensators in main.cfg. These plots also update when the selected compensator’s parameters (pole and zero locations and gains) are changed in the Compensation Designer GUI. Note that the default software supports two-pole, two-zero compensator.

10. Do not change any of the parameters in the Compensation Designer GUI, and leave the default set of compensator coefficients (COMP3) in main.cfg. These parameters can be changed and experimented with at a later time. Close the Compensation Designer GUI.

11. The default project options will be used. Save the main.cfg file. This project is ready to be build and loaded.
6.2.3.2.2.3 Build and Load the Project

1. Turn ON the 12-V auxiliary supply.
2. 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.
3. Click on the Debug button ( ) or click Run → Debug. The build one code should compile and load.
4. Notice the CCS Debug icon in the upper-right corner, which indicates the Debug Perspective view. The program should be stopped at the start of main().

6.2.3.2.2.4 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 CCs, such as memory windows and watch windows. Additionally, CCS has the ability to make time (and frequency) domain plots, which allows the user to view waveforms using graph windows.

1. Populate the Expressions Window entries by clicking on View → Scripting console on the menu bar and then opening the AddWatchWindowVars.js file from the project directory using the scripting console Open File ( ) command. The Expressions Window should look like Figure 28.

![Figure 28. Build Two: Expressions Window at Reset](image)

2. Refer to build one procedure for a detailed description for each of the parameters in the Expressions Window.
3. Open a graph window by navigating to Tools → Graph and clicking on Single Time. This creates a single-time graph window to plot the data log buffer DBUFF1. Set the Graph Properties as shown in Figure 29.

![Graph Properties](image)

**Figure 29. Build Two: Graph Properties**

4. Click OK.
6.2.3.2.2.5 Using Real-Time Emulation

Real-time emulation is a special emulation feature that allows windows within CCS to be updated while the MCU is running. This feature not only allows graphs and watch windows to update, but also allows the user to change variables or memory location values and have those changes affect the MCU behavior. This feature is very useful when tuning control law parameters on-the-fly, for example.

1. Enable real-time mode by hovering your mouse on the buttons on the horizontal toolbar and clicking button.

A message box may appear. If so, select YES to enable debug events. This will set bit one (DGBM bit) of status register one (ST1) to 0. The DGBM is the debug enable mask bit. When the DGBM bit is set to 0, memory and register values can be passed to the host processor for updating the debugger windows.

3. When a large number of windows are open, as bandwidth over the emulation link is limited, updating too many windows and variables in continuous refresh can cause the refresh frequency to slow down. Right-click on the button in the Expressions Window, and select Continuous Refresh Interval…. The refresh rate can be slowed for the Expressions Window variables by changing the Continuous refresh interval (milliseconds) value. A rate of 1000 ms is usually enough for these exercises. Click on Continuous Refresh buttons ( ) for the Expressions Window.

6.2.3.2.2.6 Run the Code

1. Run the code by using the <F8> key or the Run button on the toolbar.

2. Please use a load of at least 2 A or higher (within board specifications) at the 12-V output. If this is the first time of running closed loop test on this board, select a load between 5 A and 15 A.

3. Set the 400-V DC supply to output 390 V. Set the power supply current limit to an appropriate level for this test. Now turn ON this 390-V power supply.

4. At this point the output voltage should still be zero as the converter start command has not been initiated.

NOTE: When operating in stand-alone (DEMO mode) the converter operation should begin at this point, and the output voltage should ramp-up to approximately 12 V. DEMO mode can be selected by programming #define DEMO_MODE 1 (default is 0) at the top of the HV2PHILLLC-Main.c file.

5. Now set the start_flag to 1 in the Expressions Window.

6. The converter operation should start and the output should ramp-up to approximately 12 V.

NOTE: If output voltage does not ramp up to approximately 12 V or the FaultFlag is Set, turn OFF the 390-V DC supply immediately. Verify build one operation first as described in Section 6.2.3.1. The user may also be required to re-verify the board components and debug hardware issues (components do not match the bill of materials (BOM), PCB fabrication issue, and so forth) before this board can be tested again.
7. **Figure 30** shows a watch window that corresponds to the operation of the system with 12 V at the output, an input voltage of 390 V, and a load of approximately 20 A.

![Variables Expressions Registers](image)

**Figure 30. Build Two: Expressions Window With Iout = 20 A**

8. Observe the effect of varying load on the output voltage and input current. There should be virtually no effect on the output voltage. Similarly observe the effect of varying the input voltage. Again there should be virtually no effect on the output voltage. (1)

9. Different waveforms, like the PWM gate drive signals, input voltage, and current and output voltage may also be probed using an oscilloscope. Appropriate safety precautions should be taken and appropriate grounding requirements should be considered while probing these high voltages and high currents for this isolated DC-DC converter.

---

(1) Make sure that these changes are made within the abilities of the board as listed in Table 1.
10. Now click on main.cfg file and open SFRA. Click on setup connection, and select the appropriate COM port. Make sure that the baud rate is set to 57600 and that Boot on Connect is unchecked. Click OK.

11. Select Floating Point math.
12. Click Connect on the SFRA GUI. Once the GUI is connected, click on Start Sweep. SFRA will start applying different frequencies and collecting the response for frequency analysis. The effect of this process may be seen on the output voltage in the graph window as shown in Figure 32. The high-frequency signals riding on the output voltage indicate an active SFRA run.

![SFRA GUI and Output Voltage](image)

**Figure 32.** SFRA Window and Output Voltage During SFRA Run

13. Once the frequency sweep is complete, the response will be displayed on the SFRA GUI. The bandwidth, gain margin, and phase margin should be similar to the values noted on the Compensation Designer GUI. These values may differ to varying degrees based on how closely the parameters match the actual power stage values.
14. Close the SFRA GUI.
15. Turn OFF the 390-V DC power supply.
16. Fully halting the MCU when in real-time mode is a two-step process. First, halt the processor by using the Suspend button ( ) on the toolbar or Run → Suspend. Click button again to take the MCU out of real-time mode and then reset the MCU (Run → Reset → CPU Reset).
17. Close CCS.
End of exercise.
7 Testing and Results

7.1 Test Setup

Figure 34 shows the test setup used to validate this design. A current limited 400-V DC power supply (Agilent™ N5772A) was used to power the input. A 12-V, 600-W electronic load (Chroma 63106A) was used at the output. Simco® Aerostat XC ionizing air blower was used for air cooling when operating at loads of 30 A or higher. The following sections provide some results obtained using this board.

![Figure 34. Test Setup](image-url)
### 7.2 Test Results

#### 7.2.1 Efficiency

![Graph](image-url)

**Figure 35. Efficiency With and Without Phase Shedding at** $V_{in} = 390 \, V_{dc}$

![Graph](image-url)

**Figure 36. Efficiency With and Without Phase Shedding at** $V_{in} = 370 \, V_{dc}$

![Graph](image-url)

**Figure 37. Efficiency With and Without Phase Shedding at** $V_{in} = 410 \, V_{dc}$

---

**NOTE:** For these graphs, auxiliary power is not included in efficiency calculations.
7.2.2 Load Regulation

![Figure 38. Load Regulation](image)

7.2.3 Current Sharing

Table 6 lists the average $I_{tank1}$ and $I_{tank2}$ values observed by the controller with and without using current sharing under different output loads. The current imbalance was calculated as the difference between resonant current in the two phases as a percentage of the resonant current in phase one.

**Table 6. Current Sharing Between Phases**

<table>
<thead>
<tr>
<th>Pout (W)</th>
<th>WITHOUT CURRENT SHARING</th>
<th>WITH CURRENT SHARING (DEFAULT THRESHOLDS)</th>
<th>Imbalance%</th>
<th>Imbalance%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$I_{tank1}$ (A)</td>
<td>$I_{tank2}$ (A)</td>
<td>$I_{tank1}$ (A)</td>
<td>$I_{tank2}$ (A)</td>
</tr>
<tr>
<td>26</td>
<td>0.61</td>
<td>0.63</td>
<td>-3.28</td>
<td>0.62</td>
</tr>
<tr>
<td>53</td>
<td>0.66</td>
<td>0.66</td>
<td>-4.76</td>
<td>0.646</td>
</tr>
<tr>
<td>79</td>
<td>0.66</td>
<td>0.70</td>
<td>-6.06</td>
<td>0.67</td>
</tr>
<tr>
<td>105</td>
<td>0.69</td>
<td>0.72</td>
<td>-4.35</td>
<td>0.71</td>
</tr>
<tr>
<td>131</td>
<td>0.75</td>
<td>0.76</td>
<td>-1.33</td>
<td>0.76</td>
</tr>
<tr>
<td>158</td>
<td>0.80</td>
<td>0.80</td>
<td>0.00</td>
<td>0.81</td>
</tr>
<tr>
<td>184</td>
<td>0.87</td>
<td>0.85</td>
<td>2.30</td>
<td>0.87</td>
</tr>
<tr>
<td>210</td>
<td>0.95</td>
<td>0.91</td>
<td>4.21</td>
<td>0.92</td>
</tr>
<tr>
<td>236</td>
<td>1.01</td>
<td>0.94</td>
<td>6.93</td>
<td>0.98</td>
</tr>
<tr>
<td>263</td>
<td>1.14</td>
<td>0.97</td>
<td>14.91</td>
<td>1.06</td>
</tr>
<tr>
<td>289</td>
<td>1.22</td>
<td>1.02</td>
<td>16.39</td>
<td>1.14</td>
</tr>
<tr>
<td>315</td>
<td>1.3</td>
<td>1.08</td>
<td>16.92</td>
<td>1.21</td>
</tr>
<tr>
<td>341</td>
<td>1.41</td>
<td>1.15</td>
<td>18.44</td>
<td>1.28</td>
</tr>
<tr>
<td>367</td>
<td>1.48</td>
<td>1.20</td>
<td>18.92</td>
<td>1.35</td>
</tr>
<tr>
<td>394</td>
<td>1.54</td>
<td>1.27</td>
<td>17.53</td>
<td>1.41</td>
</tr>
<tr>
<td>420</td>
<td>1.61</td>
<td>1.33</td>
<td>17.39</td>
<td>1.47</td>
</tr>
</tbody>
</table>

(1) Table 6 lists the average $I_{tank1}$ and $I_{tank2}$ values observed by the controller with and without using current sharing under different output loads. The current imbalance was calculated as the difference between resonant current in the two phases as a percentage of the resonant current in phase one.
Figure 39 and Figure 40 show scope shots of current sharing between phases at $I_{\text{out}} = 15$ A and $V_{\text{in}} = 390$ V\text{DC}.

- Channel one = Phase one SRA
- Channel two = Phase one $I_l$ ($I_{\text{pri}-1}$ or $I_{\text{tank}1}$)
- Channel three = Phase two $I_l$ ($I_{\text{pri}-2}$ or $I_{\text{tank}2}$)
- Channel four = Phase two $V_r$

![Scope shot of current sharing between phases at $I_{\text{out}} = 15$ A and $V_{\text{in}} = 390$ V\text{DC}](image)

Figure 39. $I_{\text{tank}1}(\text{RMS}) = 1.12$ A, $I_{\text{tank}2}(\text{RMS}) = 1.03$ A
(Without Current Sharing)

Figure 40. $I_{\text{tank}1}(\text{RMS}) = 1.09$ A, $I_{\text{tank}2}(\text{RMS}) = 1.09$ A
(With Current Sharing)

Figure 41 and Figure 42 show scope shots of current sharing between phases at $I_{\text{out}} = 25$ A and $V_{\text{in}} = 390$ V\text{DC}.

- Channel one = Phase one SRA
- Channel two = Phase one $I_l$ ($I_{\text{pri}-1}$ or $I_{\text{tank}1}$)
- Channel three = Phase two $I_l$ ($I_{\text{pri}-2}$ or $I_{\text{tank}2}$)
- Channel four = Phase two $V_r$

![Scope shot of current sharing between phases at $I_{\text{out}} = 25$ A and $V_{\text{in}} = 390$ V\text{DC}](image)

Figure 41. $I_{\text{tank}1}(\text{RMS}) = 1.52$ A, $I_{\text{tank}2}(\text{RMS}) = 1.22$ A
(Without Current Sharing)

Figure 42. $I_{\text{tank}1}(\text{RMS}) = 1.42$ A, $I_{\text{tank}2}(\text{RMS}) = 1.41$ A
(With Current Sharing)
Figure 43 and Figure 44 show scope shots of current sharing between phases at $I_{out} = 30 \text{ A}$ and $V_{in} = 390 \text{ V}_{DC}$.

- Channel one = Phase one SRA
- Channel two = Phase one $I_{pri-1}$ or $I_{tank1}$
- Channel three = Phase two $I_{pri-2}$ or $I_{tank2}$
- Channel four = Phase two $V_{r}$

Figure 43. $I_{tank1}(\text{RMS}) = 1.77 \text{ A}$, $I_{tank2}(\text{RMS}) = 1.35 \text{ A}$
(Without Current Sharing)

Figure 44. $I_{tank1}(\text{RMS}) = 1.63 \text{ A}$, $I_{tank2}(\text{RMS}) = 1.64 \text{ A}$
(With Current Sharing)

Figure 45 and Figure 46 show scope shots of current sharing between phases at $I_{out} = 35 \text{ A}$ and $V_{in} = 390 \text{ V}_{DC}$.

- Channel one = Phase one SRA
- Channel two = Phase one $I_{pri-1}$ or $I_{tank1}$
- Channel three = Phase two $I_{pri-2}$ or $I_{tank2}$
- Channel four = Phase two $V_{r}$

Figure 45. $I_{tank1}(\text{RMS}) = 1.99 \text{ A}$, $I_{tank2}(\text{RMS}) = 1.53 \text{ A}$
(Without Current Sharing)

Figure 46. $I_{tank1}(\text{RMS}) = 1.76 \text{ A}$, $I_{tank2}(\text{RMS}) = 1.78 \text{ A}$
(With Current Sharing)
The two resonant tank inductors (L1 and L2) are the hottest components on the board. Without any current sharing between phases, the inductor in the phase that carries more current heats up more. As a result recording thermal temperatures of these inductors under different operating conditions provides an indication of the effectiveness of the current sharing scheme. In these tests phase one seemed to contribute more to the load current than phase two. As a result phase one inductor (L1) recorded a higher temperature than L2 when current sharing was not implemented. With current sharing these inductors recorded very similar temperatures. This is clear in the following few thermal images of the board looking from the 12-V output and looking towards the 390-V input. Following thermal images were captured with the two inductors sitting on top of the corresponding transformer. No external cooling was used.

Figure 47 and Figure 48 show thermal images at \( I_{\text{out}} = 25 \, \text{A} \) and \( V_{\text{in}} = 390 \, \text{V}_{\text{DC}} \).

![Figure 47. Without Current Sharing](image1)

![Figure 48. With Current Sharing](image2)

Figure 49 and Figure 50 show thermal images at \( I_{\text{out}} = 30 \, \text{A} \) and \( V_{\text{in}} = 390 \, \text{V}_{\text{DC}} \).

![Figure 49. Without Current Sharing](image3)

![Figure 50. With Current Sharing](image4)
7.2.4 Output Ramp-Up (Soft-Start)

Figure 51 through Figure 54 show the output ramp-up (soft-start) at approximately 920 ms (programmable) at $V_{in} = 390$ VDC.

- **Figure 51.** $I_{out} = 0$ A (No Load)
- **Figure 52.** $I_{out} = 10$ A
- **Figure 53.** $I_{out} = 25$ A
- **Figure 54.** $I_{out} = 42.5$ A
7.2.5 Zero Voltage Switching (ZVS)

Figure 55 through Figure 58 show ZVS on primary switches at \(V_{in} = 390\text{V}_\text{DC}\) where channel one = low-side switch gate to source and channel two = low-side switch drain to source.

Figure 55. Phase One Low-Side Switch at \(I_{out} = 5\text{ A}\)

Figure 56. Phase Two Low-Side Switch at \(I_{out} = 5\text{ A}\)

Figure 57. Phase One Low-Side Switch at \(I_{out} = 35\text{ A}\)

Figure 58. Phase Two Low-Side Switch at \(I_{out} = 35\text{ A}\)
7.2.6 Load Transients

Figure 59 and Figure 60 show load transients at $V_{in} = 390 \, V_{DC}$ with phase shedding enabled where channel four = output voltage ($V_{out}$) and channel three = output current ($I_{out}$)/2.

Figure 59. $I_{out}$ = 0-A to 12-A Transition

Figure 60. $I_{out}$ = 12-A to 0-A Transition

Figure 61 and Figure 62 show load transients at $V_{in} = 390 \, V_{DC}$ with phase shedding enabled where channel four = output voltage ($V_{out}$) and channel three = output current ($I_{out}$)/2.

Figure 61. $I_{out}$ = 5-A to 25-A Transition

Figure 62. $I_{out}$ = 25-A to 5-A Transition

Figure 63 and Figure 64 show load transients at $V_{in} = 390 \, V_{DC}$ with phase shedding enabled where channel four = output voltage ($V_{out}$) and channel three = output current ($I_{out}$)/2.

Figure 63. $I_{out}$ = 5-A to 35-A Transition

Figure 64. $I_{out}$ = 35-A to 5-A Transition
7.2.7 Phase Shedding During Load Transients

Figure 65 and Figure 66 show phase one disabled and re-enabled during load transients at $V_{in} = 390\ V_{DC}$ where channel one = phase one high-side PWM, channel two = phase two high-side PWM, channel three = output current ($I_{out}/2$), and channel four = output voltage ($V_{out}$).

Figure 65. Phase One Disabled During $I_{out} = 15$-A to 5-A Transition

Figure 66. Phase One Re-enabled During $I_{out} = 5$-A to 15-A Transition

Figure 67 and Figure 68 show phase two disabled and re-enabled during load transients at $V_{in} = 390\ V_{DC}$ where channel one = phase one high-side PWM, channel two = phase two high-side PWM, channel three = output current ($I_{out}/2$), and channel four = output voltage ($V_{out}$).

Figure 67. Phase Two Disabled During $I_{out} = 25$-A to 5-A Transition

Figure 68. Phase Two Re-enabled During $I_{out} = 5$-A to 25-A Transition
7.2.8 SR Enable and Disable During Load Transients

Figure 69 and Figure 70 show SR disabled and re-enabled during load transients at $V_{in} = 390$ V$_{DC}$ where channel one = phase one SRA PWM, channel two = phase two SRA PWM, channel three = output current ($I_{out}$)/2, and channel four = output voltage ($V_{out}$).

Figure 69. SR Disabled During $I_{out}$ = 5-A to 1-A Transition

Figure 70. SR Re-enabled During $I_{out}$ = 1-A to 5-A Transition
8 Adapting This TI Design

Once this TI Design has been evaluated, use the Solution Adapter tool to adapt this solution to run on a custom digital power supply that uses the same topology and similar resources.

1. To do this start with a new CCS workspace. Open Resource Explorer and navigate to controlSUITE → English → powerSUITE → Solution Adapter. Click on LLC 2PH INTERLEAVED (TIDM-1001).

Figure 71. Resource Explorer
2. On the next screen, click on the Interleaved LLC DC-DC : F28379D button shown in Figure 72.

![Interleaved LLC DC-DC: F28379D](image)

**Figure 72. Solution Adapter Button in Resource Explorer**

3. Specify a destination location for the project, and click OK. The main.cfg file should open. Now the resource mapping can be changed for different ADC inputs to match a custom design. The voltage and current scaling and other power stage parameters can also be changed to match a custom design.

![Solution Adapter: main.cfg](image)

**Figure 73. Solution Adapter: main.cfg**

4. This allows quickly adapting the TI software without having to write code. The steps outlined in the previous sections may be used to incrementally test a custom design.

5. If further code customization is required than what is possible with main.cfg, different source files (HV2PHILLLC-Main.c, HV2PHILLLC-Settings.h, and so forth) may need to be modified.
9 Design Files

9.1 Schematics
To download the schematics, see the design files at TIDM-1001.

9.2 Bill of Materials
To download the bill of materials (BOM), see the design files at TIDM-1001.

9.3 PCB Layout Recommendations

9.3.1 Layout Prints
To download the layer plots, see the design files at TIDM-1001.

9.4 Altium Project
To download the Altium project files, see the design files at TIDM-1001.

9.5 Gerber Files
To download the Gerber files, see the design files at TIDM-1001.

9.6 Assembly Drawings
To download the assembly drawings, see the design files at TIDM-1001.

10 Software Files
To download the software files, see the design files at TIDM-1001.

11 Related Documentation
2. STMicroelectronics, An introduction to LLC resonant half-bridge converter, Application Report (AN2644), 2008
3. Texas Instruments, C2000 Resonant DC/DC Developer's Kit, TMDSRESDCKIT Tools Folder
7. Texas Instruments, C2000 DPSWorkshop, Wiki Page
8. Texas Instruments, powerSUITE - Digital Power Supply Design Software Tools for C2000™ MCUs, powerSUITE Tools Folder
9. Texas Instruments, C2000™ 32-bit microcontrollers, Microcontrollers (MCU) Overview
11.1 Trademarks

controlSUITE, Delfino, Code Composer Studio are trademarks of Texas Instruments, Inc..
Agilent is a trademark of Agilent Technologies, Inc..
Simco is a registered trademark of Simco Electronics.
IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ("TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT. AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED “AS IS” AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include, without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2018, Texas Instruments Incorporated