

具有同步整流套件的高电压半桥线路电平控制 (LLC) 谐振 DC/DC 转换器软件

Daniel Chang

摘要

这份应用报告提出了一个使用 TMS320F2802x 微控制器来控制一个具有同步整流系统的半桥 LLC 谐振 DC/DC 转换器的解决方案。此 Piccolo™ TMS320F2802x 系列器件是 C2000™ 微控制器系列的产品成员，C2000 微控制器实现了电源系统的成本有效设计。借助于这些器件，有可能用一个高效且准确的方法来控制功率级。除此之外，C2000 微控制器的速度也使得它能够集成很多附加任务，以在一个正常系统中增加芯片数量和复杂性。这些任务有可能包括同步整流、系统管理和多种通信协议。

这份文档涵盖了如何运行且熟悉具有同步整流套件的 HVLLC 项目的高压半桥 LLC 谐振 DC/DC 转换器。要对 LLC 谐振 DC/DC 转换器和它们的设计注意事项进行深入的讨论，并且希望获得一个设计过程示例的话，请参见德州仪器 (TI) 的 SEM1900 题目 3，设计一个 LLC 谐振半桥电源转换器。本文档将不再赘述 SEM1900 题目 3 中的讨论细节。

内容

1	系统概览	2
2	硬件设置	7
3	软件设置	9
4	构建 1: 开环路运行以检查功能性	13
5	构建 2: 闭环运行	15
6	构建 3: 模拟比较器被启用时的闭环运行	17
7	参考书目	28

图片列表

1	TMDSHVRESLLCKIT 电路图	3
2	HVLLC 项目文件	4
3	硬件特性	8
4	工作区启动器	9
5	创建一个目标配置	10
6	配置一个新目标	11
7	设定一个缺省目标配置	11
8	将项目导入到您的工作区中	12
9	构建级 1 方框图	13
10	为构建 1 配置观察窗口	14
11	构建级 2 方框图	15
12	为构建 2 配置观察窗口	16
13	构建级 3 方框图	17
14	为构建 3 配置观察窗口	18
15	输出电压: 负载调节 ($V_{\text{输入}} = 390 \text{ VDC}$)	20
16	输出电压: 线路调节 ($I_{\text{O}} = 1 \text{ A}$)	20

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

17	开关频率与负载之间的关系 ($V_{\text{输入}} = 390 \text{ VDC}$)	21
18	效率与负载之间的关系 ($V_{\text{输入}} = 390 \text{ VDC}$)	21
19	PWM 时序示例 (黄色 - HB HS, 蓝色 - HB LS, 紫色 - SR2, 绿色 - SR1)	22
20	半桥零电压开关 (黄色 - HB HS PWM, 蓝色 - HB LS PWM, 紫色 - MP 电压)	22
21	整流器零电流开关 (黄色 - SR 电流, 蓝色 - SR Mosfet Vds, 绿色 - SR PWM)	23
22	谐振回路运行波形 (黄色 - HB HS PWM, 绿色 - 回路电流, 蓝色 - 电容器电压)	23
23	运行电流波形 (黄色 - HB HS PWM, 紫色 - HB LS PWM, 绿色 - 组合 SR 电流, 蓝色 - 回路电流)	24
24	瞬态响应: 10% ↔ 60% 负载阶跃 (蓝色 - 输出电压, 绿色 - 负载电流)	24
25	瞬态响应: 10% → 60% 负载阶跃 (蓝色 - 输出电压, 绿色 - 负载电流)	25
26	瞬态响应: 60% → 10% 负载阶跃 (蓝色 - 输出电压, 绿色 - 负载电流)	25
27	瞬态响应: 50% ↔ 100% 负载阶跃 (蓝色 - 输出电压, 绿色 - 负载电流)	26
28	瞬态响应: 50% → 100% 负载阶跃 (蓝色 - 输出电压, 绿色 - 负载电流)	26
29	瞬态响应: 100% → 50% 负载阶跃 (蓝色 - 输出电压, 绿色 - 负载电流)	27
30	高频电压纹波 (蓝色 - 输出电压)	27

图表列表

1	脉宽调制 (PWM) 和模数转换器 (ADC) 资源分配	3
2	主要变量	4
3	所使用的数字电源库	5
4	最终项目的 RAM 存储器用量	6
5	最终项目的 CPU 使用	6
6	系统特性	6

1 系统概览

1.1 硬件预览

具有同步整流套件配电板的高压半桥 LLC 谐振 DC/DC 转换器的电气规格如下:

- 输入电压: 375 至 405 VDC
- 额定输出功率: 300W
- 输出电压: 12 VDC
- 额定输出电流: 25A
- 输出电压线路稳压 ($I_o = 1A$): $\leq 1\%$
- 输出电压负载调节 ($V_{\text{输入}} = 390V$): $\leq 1\%$
- 输出电压峰值至峰值纹波 ($V_{\text{输入}} = 390V$ 且 $I_o = 25A$): $\leq 120mV$
- 效率 ($V_{\text{输入}} = 390V$ 且 $I_o = 25A$): $> 90\%$
- 开关频率 (正常运行): 80kHz 至 150kHz
- 谐振频率: $f_0 \sim 130kHz$

图 1 中以电路图格式显示了此主板的 LLC 谐振功率级并图示了主要连接和被映射到 C2000 MCU 的反馈值；表 1 列出这些资源。请注意，每个 C2000 MCU 上并不提供所有资源。要获得更加详细的信息，请见电路原理图和器件专用数据表。

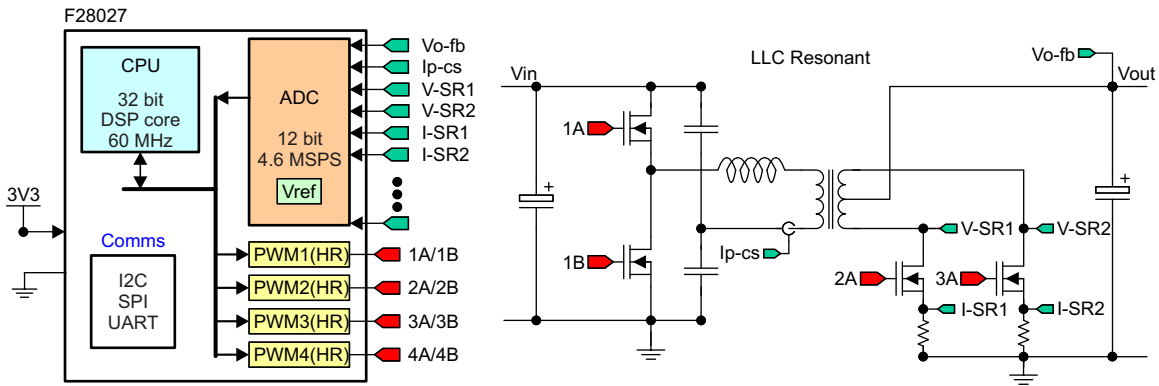


图 1. TMDSHVRESLLCKIT 电路图

表 1. 脉宽调制 (PWM) 和模数转换器 (ADC) 资源分配

宏名称		信号名称	PWM/ADC 通道	说明	
LLC 谐振 + SR	[M1]	PWM-1	PWM-1A	PWM-1A	半桥高侧 PWM 信号
		PWM-2	PWM-1B	PWM-1B	半桥低侧 PWM 信号
		PWM-3	PWM-2A	PWM-2A	整流器 1 PWM 信号 (负半周期)
		PWM-4	PWM-3A	PWM-3A	整流器 2 PWM 信号 (正半周期)
		Vo-fb	Vo-fb	ADC-A7	输出电压感测
		V-SR1	V-SR1	ADC-A2	整流器 1 Vds 电压感测 (与 I-SR1 复用)
		V-SR2	V-SR2	ADC-A4	整流器 2 Vds 电压感测 (与 I-SR2 复用)
		Ip-cs	Ipri-cs	ADC-B1	谐振回路电流感测 (经整流)
		I-SR1	I-SR1	COMP1 (ADC-A2)	整流器 1 电流感测 (与 V-SR1 复用)
		I-SR2	I-SR2	COMP2 (ADC-A4)	整流器 2 电流感测 (与 V-SR1 复用)

1.2 软件概述

1.2.1 构建选项

为了使您能够逐渐构建并理解此项目，此项目被 HVLLC-Main.c 和 HVLLC-ISR.asm 文件中的 #if 选项分成了不同的构建。所使用的构建由 HVLLC-Settings.h 中的变量 INCR_BUILD 设定。下面是 HVLLC 项目中不同可用构建的简要说明。

- 构建 1: 用于检查 DC/DC 级功能性的开环运行。
- 构建 2: DC/DC 级的闭环运行
- 构建 3: 模拟比较器被启用时的 DC/DC 级闭环运行

HVLLC 项目中的主要变量被用来控制和监视支持同步整流的半桥 LLC 谐振 DC/DC 转换器，表 2 中列出了这些主要变量以及每个变量的简要说明。图 2 中显示了主要项目文件。表 3 中显示了所使用的数字电源库。

表 2. 主要变量

LLC_Enable	启用半桥脉宽调制 (PWM) 信号
SR_Enable	启用同步整流 PWM 信号
Comp_Enable (只用于构建 3)	启用模拟比较器以实现基于电流电平的同步整流 (SR) 关闭
Gui_Vset (只用于构建 2, 3)	设定闭环输出电压目标值 (Q9)
Gui_Vout	监视输出电压 (Q9)
Pgain (只用于构建 2, 3)	针对比例积分微分 (PID) 控制环路的成比例增益
Igain (只用于构建 2, 3)	针对 PID 控制环路的积分增益
Dgain (只用于构建 2, 3)	针对 PID 控制环路的微分增益
Duty{X}	设定 PWM 信号的占空比; 标称值 50%
周期	PWM 的周期 (频率的倒数)。在构建 1 中, 这由用户手动设定。在构建 2 和 3 中, 这由控制环路设定。
Min_Period (只用于构建 2, 3)	软件强制最小 PWM 周期限值
Max_Period (只用于构建 2, 3)	软件强制最大 PWM 周期限值
RED	半桥 PWM 上升边沿延迟 (高侧上升边沿延迟)
FED	半桥 PWM 下降边沿延迟 (低侧上升边沿延迟)
REM{X}	SR{X} PWM 上升边沿裕量 (上升边沿延迟)
FEM{X}	SR{X} PWM 下降边沿裕量 (下降边沿提前)
COMP{X} (只用于构建 3)	SR{X} PWM 比较器触发值 (寄存器计数)。

1.2.2 主项目文件

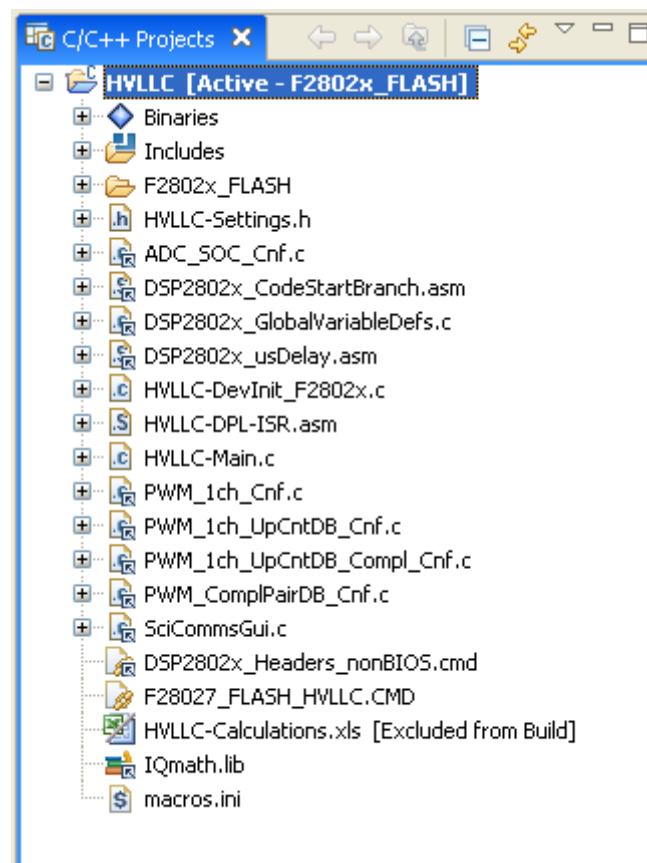


图 2. HVLLC 项目文件

本项目中使用的关键框架文件为：

- HVLLC-Main.c - 这个文件被用来初始化、运行和管理此应用。这个文件是支持应用的“大脑”。
- HVLLC-DevInit_F2802x.c – 这个文件负责器件（此情况下为 F28027）的初始化和配置，并且包括设置时钟，锁相环 (PLL)，通用输入输出 (GPIO) 等的功能。
- HVLLC-ISR.asm - 这个文件包含时序关键“控制类型”代码。这个文件有一个初始化部分，此部分由 C 语言可调用的汇编程序 subroutine `_DPL_Init` 执行一次。这个文件还包含 `_DPL_ISR` 例程，此例程的执行速度由 PWM 或者用来触发它的定时器确定。
- HVLLC-Settings.h - 这个文件被用来设定项目的全局定义（构建选项）。请注意，它被链接至 LED-ColorMix-Main.c 和 LED-ColorMix-ISR.asm 两个文件。
- HVLLC-Calculations.xls - 这是一个数据表文件，此文件计算将 MCU 使用的 Q 值数字转换为实际值时所使用的不同比例因子。变量 `K-Vout` 和 `iK_Vset` 是比例因子的示例。

1.2.3 数字电源库文件

为了减少每次开发代码所花费的精力，使用了一个采用宏区块的方法。这些宏区块是数字电源库的一部分，用 C 语言可调用汇编语言编写，并且可具有一个 C 语言和汇编语言构件。表 3 显示了这个项目中使用的数字电源库文件的列表。

表 3. 所使用的数字电源库

C 语言配置函数	ASM 宏
ADC_SOC_Cnf.c	ADCDRV_1ch.asm
无	CNTL_2P2Z.asm
PWM_1ch_Cnf.c	无
PWM_CmplPairDB_Cnf.c	PWMDRV_LLC_CmplPairDB.asm
PWM_1ch_UpCntDB_Cnf.c	PWMDRV_LLC_1ch_UpCntDB.asm
PWM_1ch_UpCntDB_Cmpl_Cnf.c	PWMDRV_LLC_1ch_UpCntDB_Cmpl.asm

由于外设的配置在 C 语言中完成，此配置的修改十分方便。ASM 驱动程序宏为实时运行提供所需的紧凑代码。全部细节请参见数字电源库文档。项目中每个文件的作用描述如下。

- ADC_SOC_Cnf.c – 被用来按照规定配置 ADC 外设。
- ADCDRV_1ch.asm - 这个宏对 ADC 模块的用法进行了摘要。此驱动程序宏将 ADCRegister 的结果复制到一个 NetBus 阵列变量中
- CNTL_2P2Z.asm - 这个宏是一个使用无限脉冲响应 (IIR) 滤波器结构实现的二阶补偿器。这个函数所需的 5 个系数在 C 语言后台循环中被声明为一个长数组。
- PWM_1ch_Cnf.c – 被用来配置 PWM4。PWM4 被用来触发控制环路中断处理例程 (ISR)
- PWM_CmplPairDB_Cnf.c – 被用来配置 PWM1。PWM1 被用来控制半桥金属氧化物半导体场效应晶体管 (MOSFET)。PWM1 也被用来触发 PWM 更新 ISR。
- PWM_1ch_UpCntDB_Cnf.c – 被用来配置 PWM3。PWM3 被用来控制同步整流器 MOSFET 2。
- PWM_1ch_UpCntDB_Cmpl_Cnf.c – 被用来配置 PWM2。PWM2 被用来控制同步整流器 MOSFET 1。
- PWMDRV_LLC_CmplPairDB.asm - 被用来在运行期间根据用户和/或控制环路输入来更新 PWM1 寄存器。
- PWMDRV_LLC_1ch_UpCntDB.asm - 被用来在运行期间根据用户和/或控制环路输入来更新 PWM3 寄存器。
- PWMDRV_LLC_1ch_UpCntDB_Cmpl.asm - 被用来在运行期间根据用户和/或控制环路输入来更新 PWM2 寄存器。

1.2.4 RAM 和 CPU 使用

表 4. 最终项目的 RAM 存储器用量

构建级	程序存储器用量 2802x	数据存储器用量 2802x⁽¹⁾
构建 3 (闪存)	220 个字	499 个字

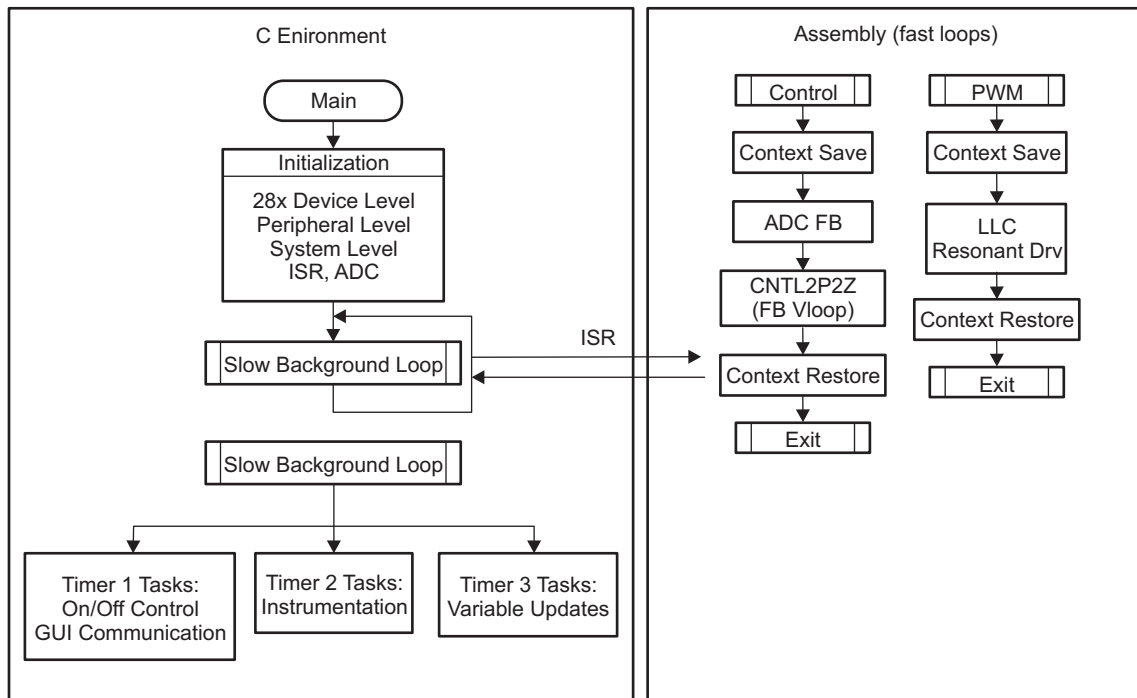
⁽¹⁾ 不包括堆栈尺寸

表 5. 最终项目的 CPU 使用

宏名称	周期数量
控制环路 ISR (100kHz)	82
执行环境保存, 恢复, ISR 管理等	27
ADCDRV_1ch	6
CNTL_2P2Z	47
PWM 更新 ISR (100kHz 至 130kHz)	121
执行环境保存, 恢复, ISR 管理等	27
PWMDRV_LLC_CompIPairDB	26
PWMDRV_LLC_1ch_UpCntDB	38
PWMDRV_LLC_1ch_UpCntDB_CompI	31
总的周期数量	203
60MHz 时的 CPU 使用率 (100kHz 时)	33.83%

表 6. 系统特性

开发/仿真	Code Composer Studio™ 支持实时调试的 v4.1 版本 (或更高版本)
目标控制器	TMS320F2802x
PWM 频率	PWM1, PWM2, PWM3 - 可在 100kHz 至 130kHz 之间变化 PWM4 - 固定 100kHz
中断	控制环路 ISR - 固定 100kHz。由 EPWM4 触发 PWM 更新 ISR - 可在 100kHz 至 130kHz 之间变化 由 EPWM1 触发



2 硬件设置

图 3 中列出了具有同步整流电路板的半桥 LLC 谐振 DC/DC 转换的某些主要连接器和特性。每个组件的名称以它们的宏编号开始，然后是参考名称。例如，[M2]-J1 是指宏 M2 内的跳线 J1，而 [Main]-J1 是指位于其它已定义的宏区块外电路板上的 J1。全部细节请参见硬件指南文档。

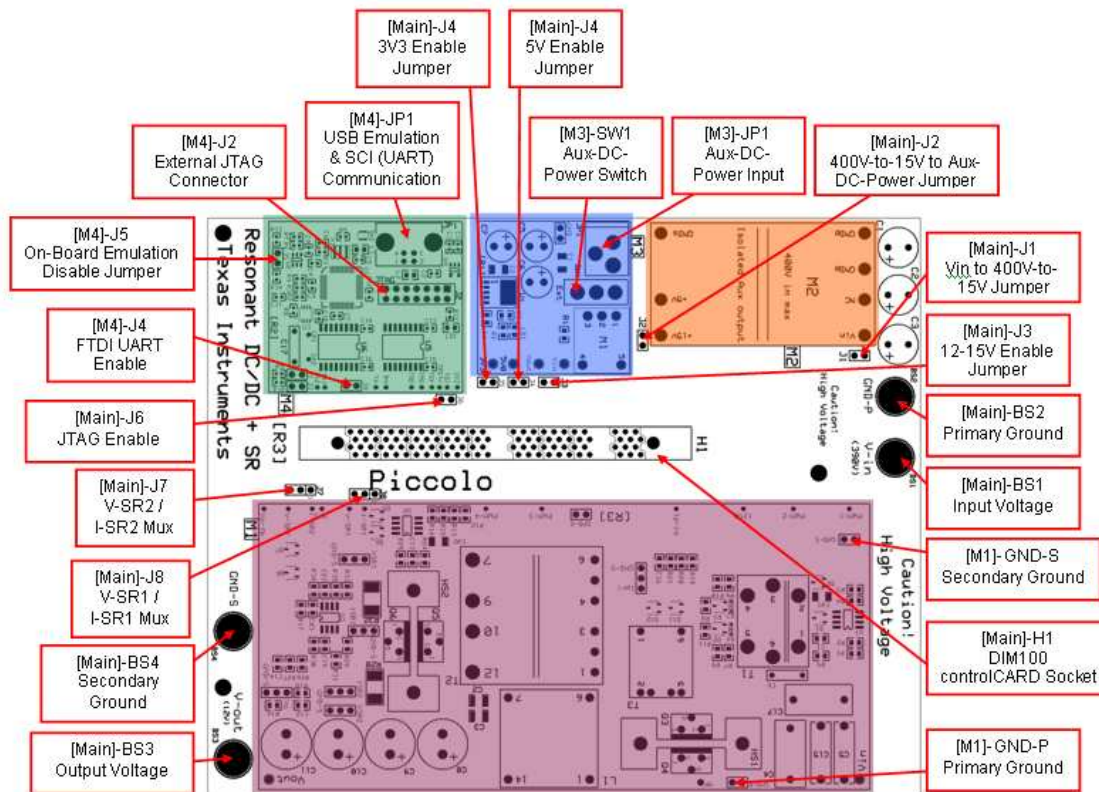


图 3. 硬件特性

执行或验证以下步骤来为具有同步整流套件的高压半桥 LLC 谐振 DC/DC 转换器的使用做好准备：

1. 将一个 F28027 控制卡插入到插槽 [Main]-H1 中。
2. 使用一条 USB 线缆将您的计算机连接到电路板。靠近 USB 连接器 [M4]-JP1 的 [M4]-LD1 应该被打开。
3. 检验以下的跳线设置：
 - (a) 在 [Main]-J1, J2 上未放置跳线。
 - (b) 跳线被放置在 [Main]-J3, J4, J5, J6 上。
 - (c) 跳线被放置在 [Main]-J7, J8 的引脚 1-2 上。
 - (d) 一个跳线被放置在 [M4]-J4 上。
4. 将 12V VDC 电源连接至 [M3]-JP1 来为辅助电源轨供电。
5. 在 [Main]-BS1, BS2 之间连接一个 390V VDC, 1A (最大值) 电源。
6. 在 [Main]-BS3, BS4 之间连接一个 300W (最大值) 负载。
7. 设定电源开关 [M3]-SW1, 使其指向“Ext”标签。[M3]-LD1 应该打开, 并且一个绿光发光二极管 (LED) 应该接通 controlCARD。

注：如果从未安装过 Code Composer Studio, 也许需要安装驱动程序以使电路板正常工作。如果当 USB 线缆被连接在电路板和计算机之间时出现一个弹出窗口, 使用安装向导从包含在此套件中的 USB 驱动的 XDS100v1 目录中安装驱动程序。

3 软件设置

3.1 安装 **Code Composer** 和 **ControlSUITE™**

1. 如果还未安装，从包含在此套件中的 USB 驱动中安装 Code Composer v4.x。
2. 打开<http://www.ti.com/controlsuite>并运行 controlSUITE 安装程序。选择安装“Multi-DC/DC Color LED Kit”（多 DC/DC 彩色 LED 套件）软件并且还允许安装程序自动下载将校验的软件。

3.2 将 **Code Composer Studio** 设置为与此套件一同工作

1. 打开“Code Composer Studio v4”。
2. 一旦 Code Composer Studio 打开，将出现工作区启动器 (workspace launcher) 要求您选择一个工作区位置，（请注意，工作区是硬盘驱动器上的一个位置，在这里保存了用户针对 IDE 的所有设置。例如，打开哪个项目、选择什么配置、它们的保存位置等）。这可以是磁盘上的任一位置，图 4 中提到的位置仅供参考。还要注意的，如果这不是您第一次运行 Code Composer，那么将不会出现这个对话框。
 - (a) 单击“Browse...”（浏览）按钮。
 - (b) 通过生成所需的新文件夹来创建下面的路径
 - (c) “C:\Documents and Settings\\My Documents\CCSv4_workspaces\HVLLC”
 - (d) 取消选中“Use this as the default and do not ask again”（使用这个作为缺省路径，不要再次询问）框。
 - (e) 单击“OK”

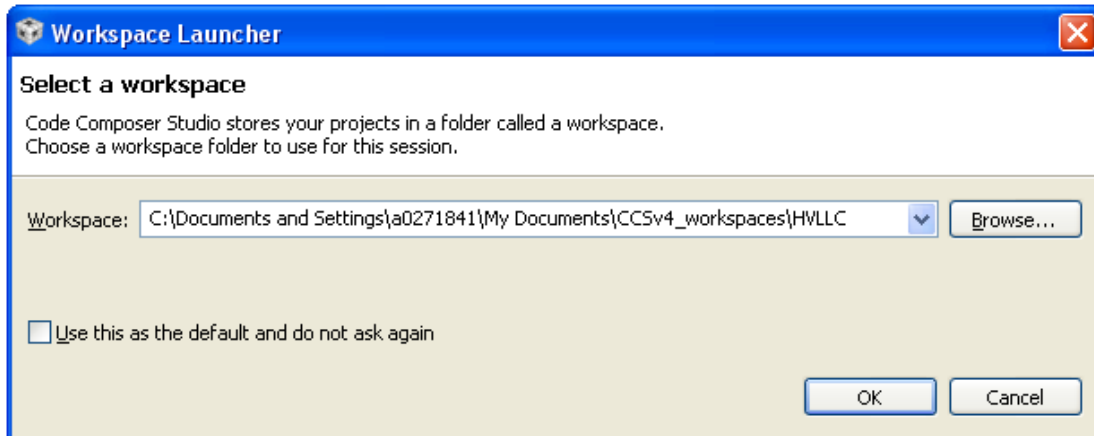


图 4. 工作区启动器

- 配置 Code Composer，使其知道将被连接到哪个 MCU。单击“Target → New Target Configuration...”（目标 → 新目标配置）。将新配置命名为“XDS100 F28027.ccxml”。确保选中“Use shared location”（使用共用位置），然后单击“Finish”（完成）。

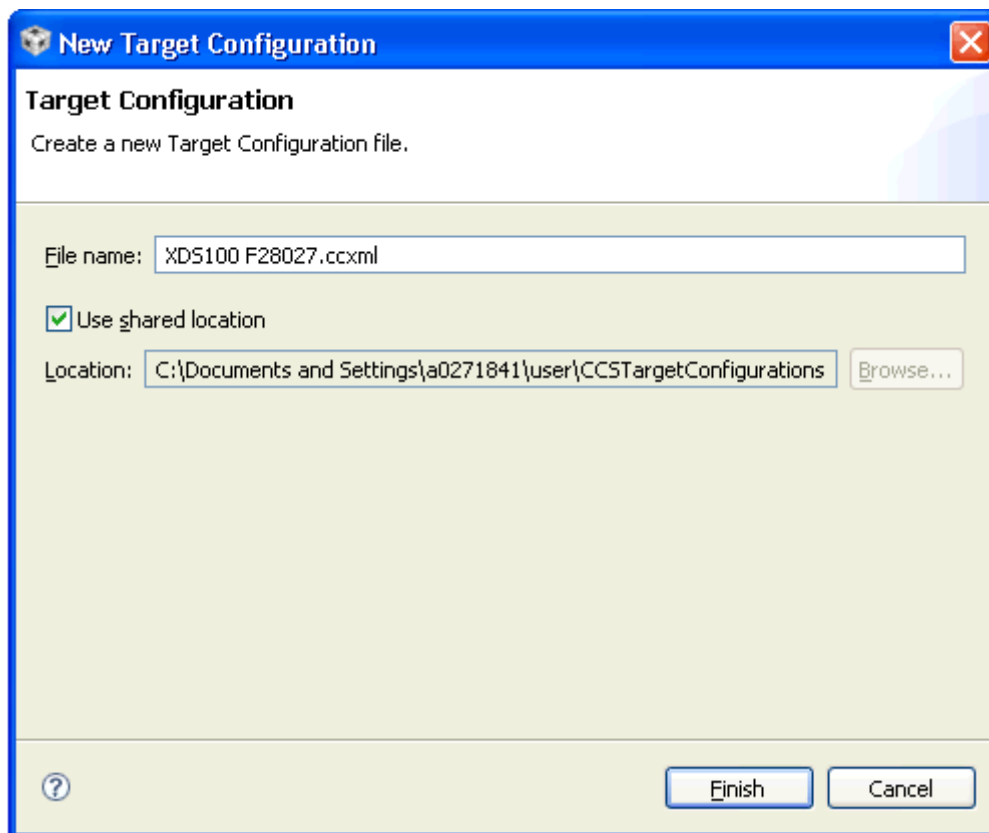


图 5. 创建一个目标配置

4. 这将打开一个如图 5 中所见的新标签。如下选择并进入选项：
 - (a) 连接 - 德州仪器 (TI) XDS100v1 USB 仿真器
 - (b) 器件 - TMS320F28027
 - (c) 点击 Save (保存)
 - (d) 关闭“XDS100 F28027.ccxml”标签页

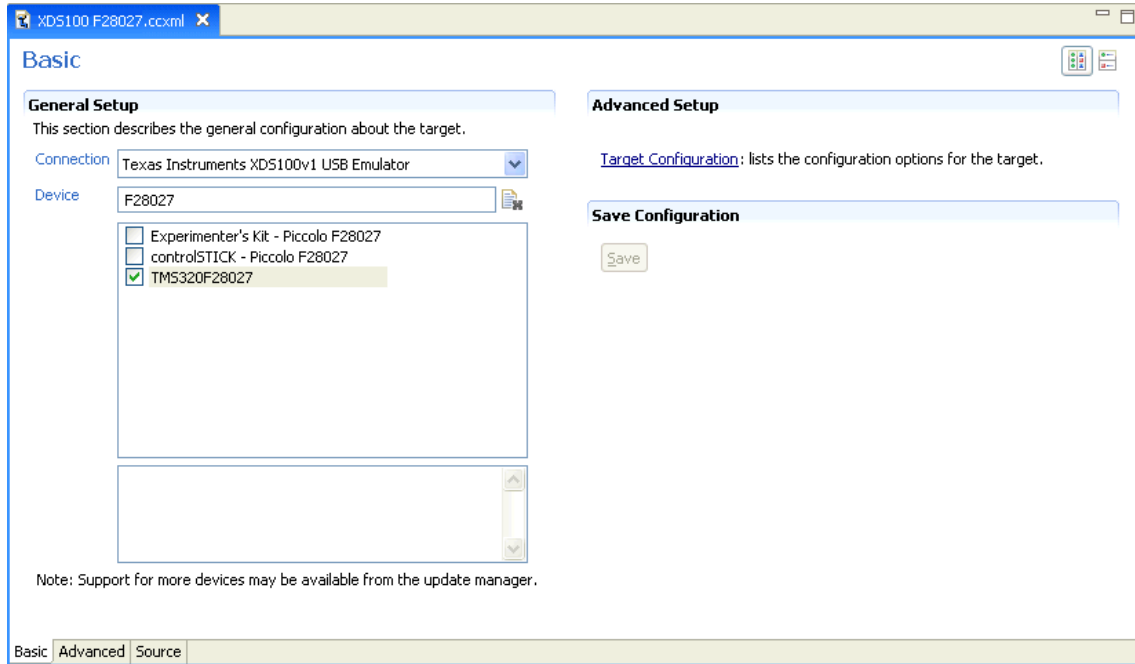


图 6. 配置一个新目标

5. 假定这是您第一次使用 Code Composer，现在“XDS100 F28027”配置被设定为针对 Code Composer 的缺省目标配置。这通过进入“View → Target Configurations”（查看 → 目标配置）来选中。在“User Defined”（由用户定义）部分，右键单击“XDS00 F28027.ccxml”文件并选择“Set as Default”（设定为缺省值）。这个标签页还使得您能够再次使用已有的目标配置并将它们连接至特定项目。

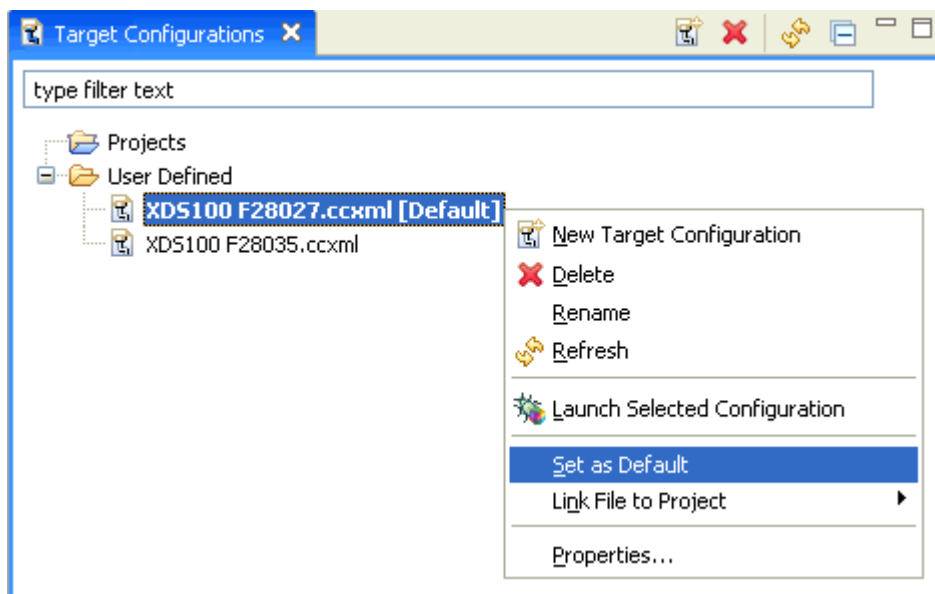


图 7. 设定一个缺省目标配置

6. 通过单击“Project → Import Existing CCS/CCE Eclipse Project”（项目 → 导入已有的 CCS/CCE Eclipse 项目）来将项目添加到您当前的工作区中。
 - (a) 浏览套件文件夹内的项目目录。缺省位置是：
C:\TI\controlSUITE\development_kits\TMDSHVRESLLCKIT_v1.0\HVLLC
 - (b) 单击“Finish”（完成）来将项目载入工作区。

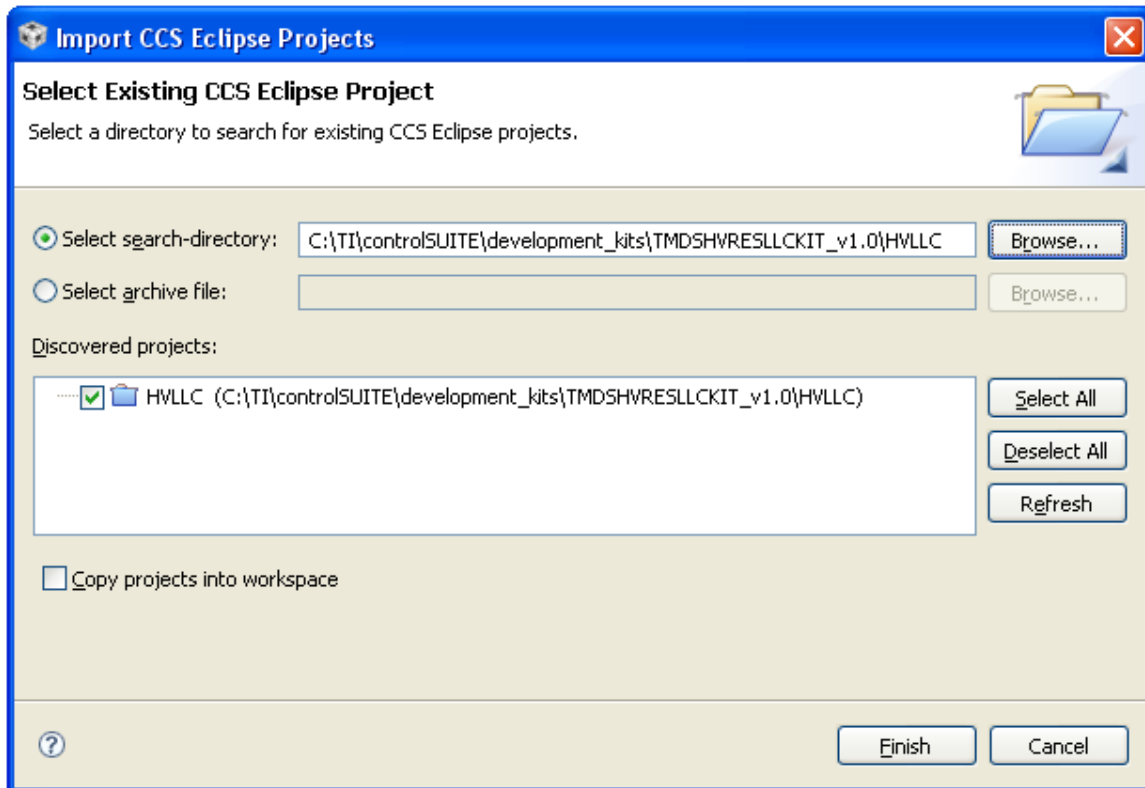


图 8. 将项目导入到您的工作区中

7. HVLLC 项目将被设定为激活的项目。右键单击项目名称并单击“Save as Active Project”（保存为激活的项目）。展开项目的文件构建。

3.3 递增系统构建

为了使您能够逐渐构建并理解此项目，此项目被 HVLLC-Main.c 和 HVLLC-ISR.asm 文件中的 #if options 分成了不同的构建。所使用的构建由 HVLLC-Settings.h 中的变量 INCR_BUILD 设定。下面是 HVLLC 项目中不同可用构建的简要说明。

- 构建 1: 开环路运行以检查 DC/DC 级的功能性
- 构建 2: DC/DC 级的闭环运行
- 构建 3: 模拟比较器被启用时 DC/DC 级的闭环运行

4 构建 1: 开环路运行以检查功能性

注: 这一部分假设硬件设置和软件设置部分已经完成, 在继续执行前仔细检查这些部分。

这个构建的目标如下:

- 验证 PWM 和 ADC 信号运转正常
- 验证电路板上的功率级运转正常
- 手动控制电路板上功率级的周期 (频率的倒数), 并且评估其输出

图 9 中描述了软件中使用的系统组件。

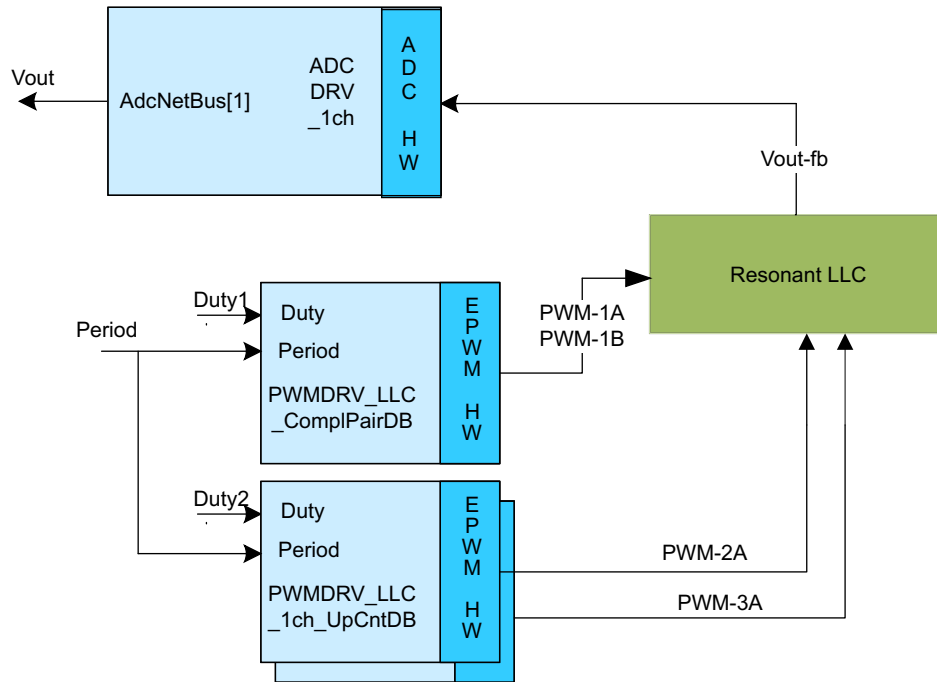




图 9. 构建级 1 方框图

4.1 检查此项目

- 在最初的引导过程完成后, 软件从主函数中开始执行。打开 HVLLC-Main.c 并找到 main() 函数 (第 241 行)。
- 软件在 main() 函数中做的第一件事情就是调用一个 HVLLC-DevInit_F2802x.c 中被称为 DeviceInit() 的函数。通过双击项目窗口中的文件名可打开并检查 HVLLC-DevInit_F2802x.c。在这个文件中, 不同的外设时钟被启用或被禁用并且定义了功能引脚分配, 此引脚分配配置了哪个外设从哪个引脚输出。
 - 请确认 ADC 和 PWM-14 外设时钟被启用 (第 99-117 行)。还需要确认 GPIO00-GPIO07 被配置为 PWM 输出 (第 136-182 行)。
- 此项目与递增构建一同提供, 在这些构建中系统的不同组件和宏区块被逐一连接来形成完整系统。这有助于逐步调试并了解此系统。
 - 从 C/C++ 项目标签页上打开文件 HVLLC-Settings.h 并确保 INCR_BUILD 被置位为 1 且保存这个文件。在您测试了构建 1 后, 这个变量需要被重新定义来移动到构建 2, 以此类推, 直到完成所有构建。

- 返回 HVLLC-Main.c 文件。请注意不同的递增构建配置代码。构建级 1 配置代码出现在 478-493 行上。其中配置了 ADC, PWM 和宏区块连接。请注意, 如果 INCR_BUILD 被置位为 1, 第 496-539 行将被编译器忽略, 这是因为它们不属于当前的构建。在 HVLLC-ISR.asm 中, 也用相似的方法根据 INCR_BUILD 的值来启用或禁用代码的。

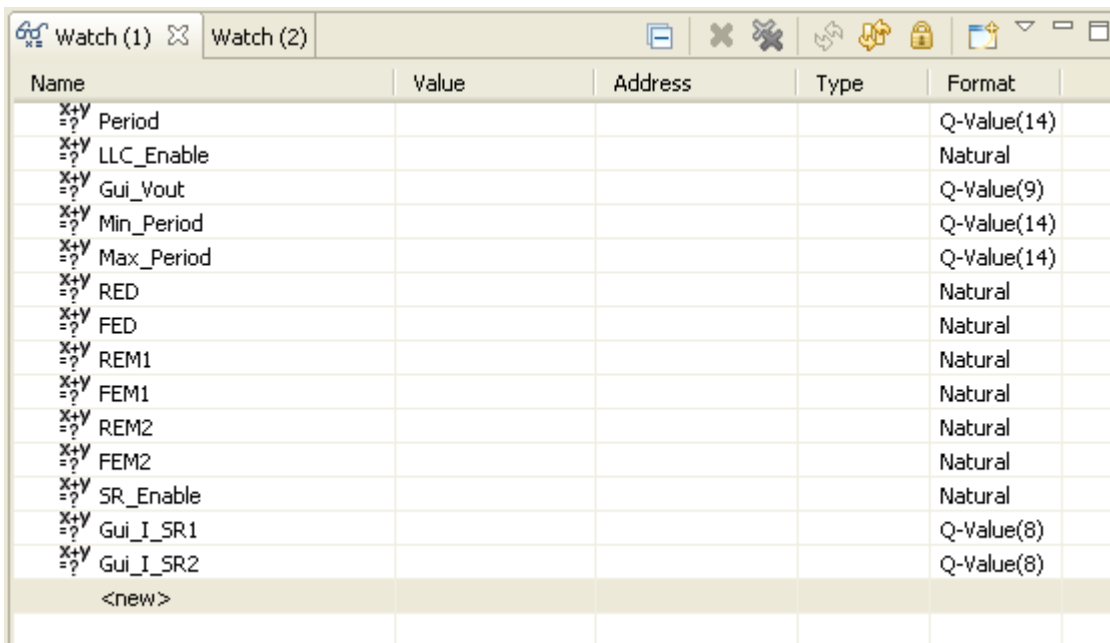
4.2 构建和加载项目

1. 右键单击项目名称并单击“Rebuild Project”（重建项目），请观察控制台窗口。项目中的任何错误将被显示在控制台窗口中。
2. 在成功完成项目构建时, 单击“Debug”（调试）按钮 ，此按钮位于屏幕的左上侧。现在, IDE 将自动连接至目标, 将输出文件载入到器件内并换到调试视图。
3. 单击实时模式  按钮, 此按钮名为“Enable silicon real-time mode”（启用芯片实时模式）。这使得您能够实时编辑并查看变量, 而无需暂停程序。
4. 也许会出现一个消息框。如果是这样的话, 选择 YES 来启用调试事件。这将把状态寄存器 1 (ST1) 的位 1 (DGBM 位) 置位为“0”。DGBM 是调试使能屏蔽位。当 DGBM 位被置位为“0”时, 为了更新调试器窗口, 存储器和寄存器值可被传递到主机处理器。

4.3 设置观察窗口和曲线图


1. 单击: 菜单条上的 View → Watch 打开一个观察窗口来查看项目中正在使用的变量。将图 10 中出现的变量添加到观察窗口。通过右键单击一个特定变量然后选择一个 Q 值来改变一个变量的格式。改变每个变量的格式来与图形相匹配。在构建 1 中时, 这个观察窗口中的变量被用来控制和监控电路板的状态。

提示: 在按住 Shift 键的同时单击可以选择多个变量或项, 然后右键单击并改变所有选中的变量的格式。




Name	Value	Address	Type	Format
Period				Q-Value(14)
LLC_Enable				Natural
Gui_Vout				Q-Value(9)
Min_Period				Q-Value(14)
Max_Period				Q-Value(14)
RED				Natural
FED				Natural
REM1				Natural
FEM1				Natural
REM2				Natural
FEM2				Natural
SR_Enable				Natural
Gui_I_SR1				Q-Value(8)
Gui_I_SR2				Q-Value(8)
<new>				

图 10. 为构建 1 配置观察窗口




2. 单击观察窗口中的 Continuous Refresh（持续刷新）按钮  这将启用窗口的实时运行模式。

4.4 运行代码

1. 通过按下 Debug Tab（调试标签页）内的 Run（运行）按钮来运行代码 。现在项目应该运行, 而

观察窗口中的值应该持续更新。

2. 在主电源被设定为 0V 或被关闭时检验 PWM 和 ADC 电路:
 - (a) 验证所有电压和电流反馈变量接近于 0。由于常常在 ADC 上出现某些低电平噪声, 所以它们不太可能正好为 0.0。
 - (b) 通过将“LLC_Enable”和“SR_Enable”置位为 1 来启用 PWM。使用一个示波器来检验 PWM 波形。调整“Period”(周期)值, 并检验 PWM 波形变化的频率。
3. 将负载设定为 1A 的最小值。
4. 将主电源斜升至 390 VDC, 并且验证在输出上出现电压。
5. 小心调整“Period”, 并在您减少“Period”时检验“Gui_Vout”在增加, 在您增加“Period”时“Gui_Vout”在减少。
6. 小心调整“RED”和“FED”来观察它们是如何影响半桥 PWM 信号的。
7. 小心调整“REM(X)”和“FEM(X)”来观察它们是如何影响整流器 PWM 信号的。
8. 一旦完成, 关闭或设定主电源为 0 VDC。

9. 禁用实时模式  并复位处理器  (Target → Reset → Reset CPU) (目标 → 复位 → 复位 CPU), 然后通过单击  (Target → Terminate All) (目标 → 终止所有) 来终止所有调试会话。这将从 MCU 上断开 Code Composer Studio。

5 构建 2: 闭环运行

这个构建的目标如下:

- 使用 PID 控制来调节具有闭环反馈的输出电压。

图 11 中描述了软件中使用的系统组件。

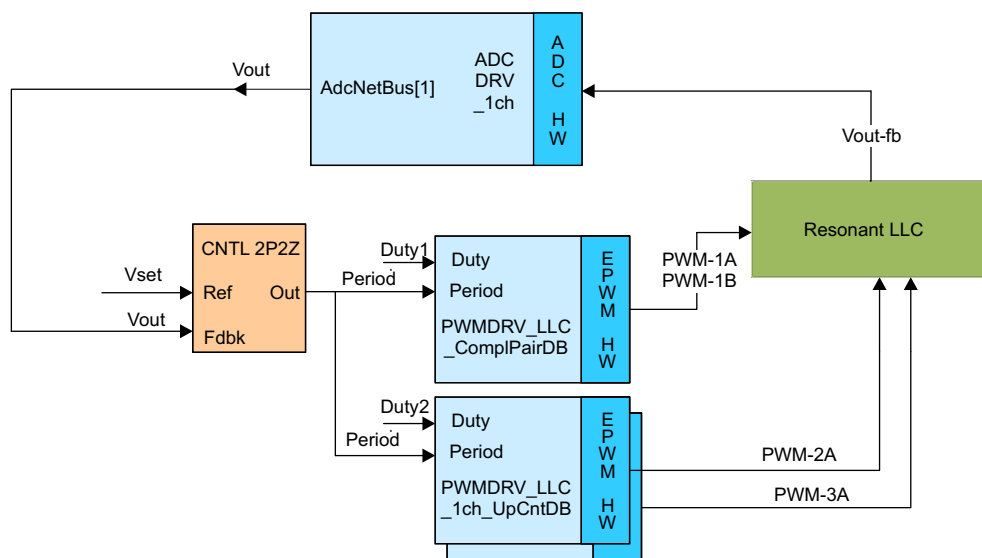




图 11. 构建级 2 方框图

针对此构建的指令比第一个构建中的指令更简洁。要获得进一步的指导, 请参见 4 节中的指令。在运行构建 2 级之前运行构建 1。

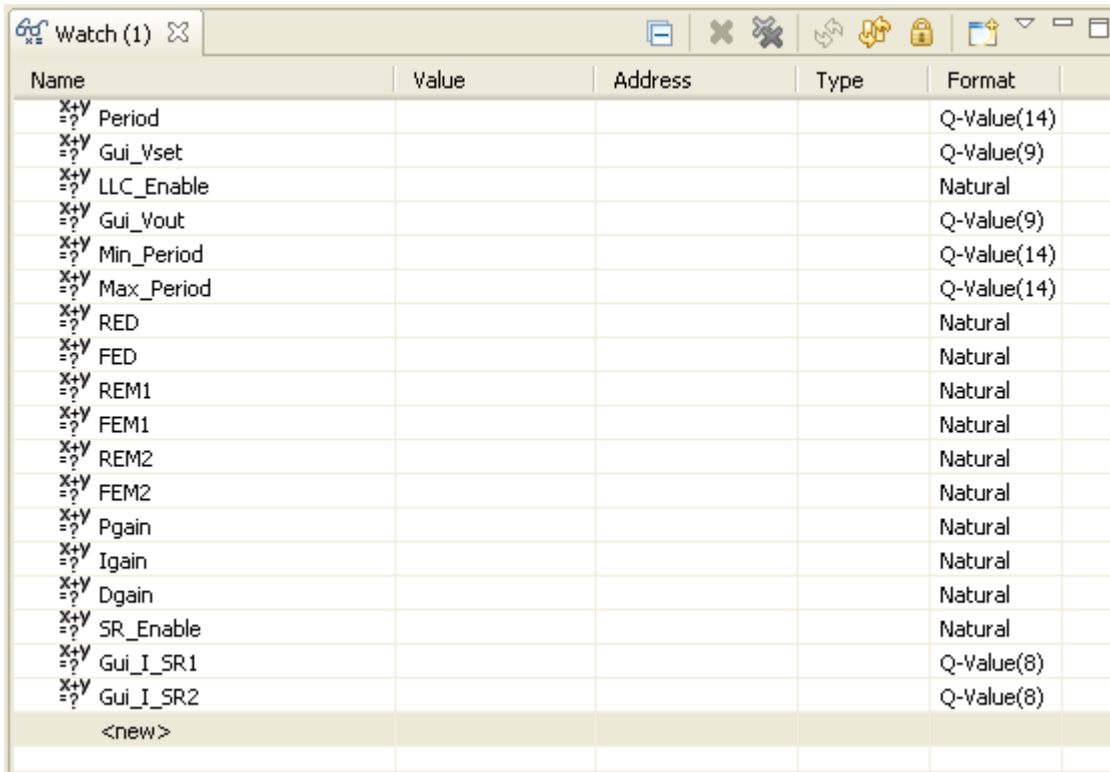
5.1 构建和加载项目

1. 打开 HVLLC-Settings.h 并将递增构建级改为 2 (#define INCR_BUILD 2)。保存此文件。
2. 右键单击文件名并选择“Rebuild Project”。

- 成功完成构建时，单击“Debug”按钮 ，此按钮位于屏幕的左上角。现在，IDE 将自动连接至目标，将输出文件载入到器件内并切换到调试视图。
- 单击实时模式按钮 ，此按钮名为“Enable silicon real-time mode”。这使得您能够实时编辑并查看变量，而无需暂停程序。

5.2 设置观察窗口和曲线图

- 将一个观察窗口添加到工作区；添加变量，并将其设定为使用如图 12 中所示的正确格式。在构建 2 中，这个观察窗口中的变量被用来控制和监控电路板的状态。







Name	Value	Address	Type	Format
X+Y =? Period				Q-Value(14)
X+Y =? Gui_Vset				Q-Value(9)
X+Y =? LLC_Enable				Natural
X+Y =? Gui_Vout				Q-Value(9)
X+Y =? Min_Period				Q-Value(14)
X+Y =? Max_Period				Q-Value(14)
X+Y =? RED				Natural
X+Y =? FED				Natural
X+Y =? REM1				Natural
X+Y =? FEM1				Natural
X+Y =? REM2				Natural
X+Y =? FEM2				Natural
X+Y =? Pgain				Natural
X+Y =? Igain				Natural
X+Y =? Dgain				Natural
X+Y =? SR_Enable				Natural
X+Y =? Gui_I_SR1				Q-Value(8)
X+Y =? Gui_I_SR2				Q-Value(8)
<new>				

图 12. 为构建 2 配置观察窗口

- 单击观察窗口中的 Continuous Refresh（持续刷新）按钮。 

5.3 运行代码

1. 通过按下 Debug Tab (调试标签页) 内的 Run 按钮来运行代码 。
2. 通过将“LLC_Enable”和“SR_Enable”置位为 1 来启用 PWM。
3. 将负载设定为 1A 的最小值。
4. 将主电源斜升至 390 VDC。
5. 将“Gui_Vset”设定为 12 VDC。“Period”应该自动改变, 而“Gui_Vout”应该变化为 12 VDC 以与“GUI_Vset”相匹配。
6. 改变负载时所做的试验。请注意, “Period”(频率的倒数)随着负载的增加而增加, 随着负载的下降而下降。
7. 一旦完成, 关闭或将主电源设定为 0 VDC。
8. 禁用实时模式  并复位处理器  (Target → Reset → Reset CPU), 然后通过单击  (Target → Terminate All) 来终止所有调试会话。

6 构建 3: 模拟比较器被启用时的闭环运行

这个构建的目标如下:

- 使用模拟比较器根据 SR 电流电平来动态地调整 SR PWM 关闭时序。

图 13 中描述了软件中使用的系统组件:

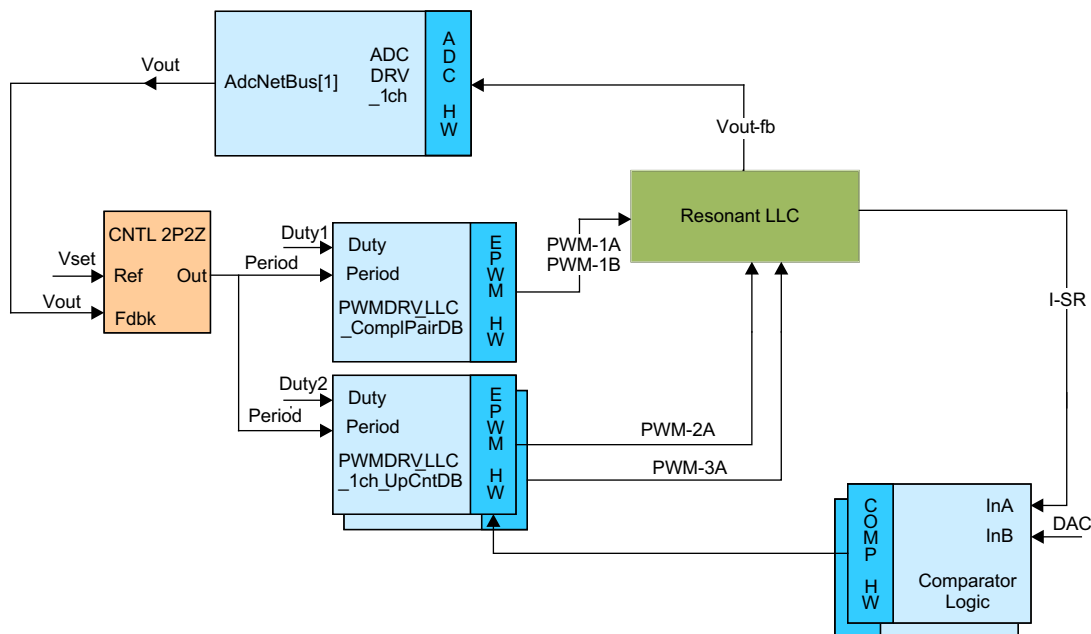




图 13. 构建级 3 方框图

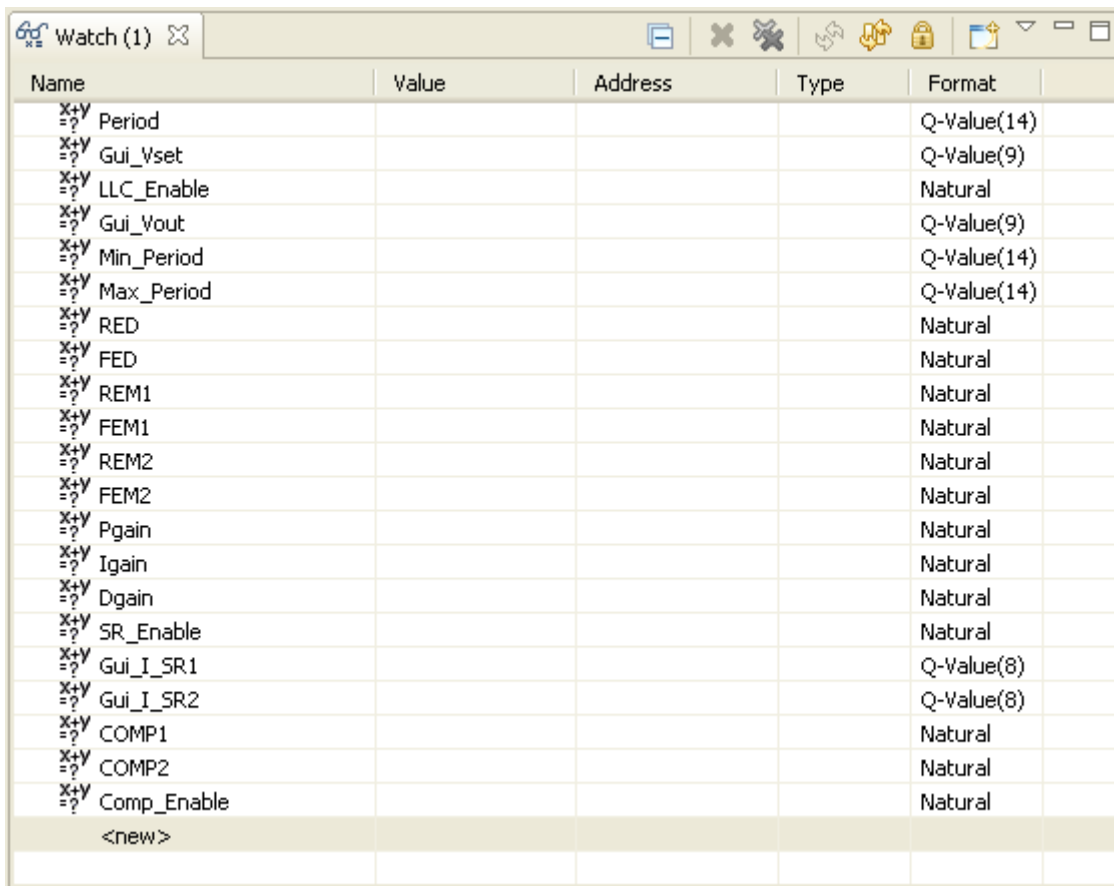
针对此构建的指令比第一个构建中的指令更简洁。要获得进一步的指导, 请参见 5 节中的指令。在运行构建 3 级前, 请先运行构建 2。

6.1 构建和加载项目

1. 打开 HVLLC-Settings.h 并将递增构建级改为 3 (#define INCR_BUILD 3)。保存此文件。
2. 右键单击文件名并选择“Rebuild Project”。
3. 成功完成构建时，单击“Debug”按钮 ，此按钮位于屏幕的左上角。现在，IDE 将自动连接至目标，将输出文件载入到器件内并切换到调试视图。
4. 单击实时模式按钮 ，此按钮名为“Enable silicon real-time mode”。这使得您能够实时编辑并查看变量，而无需暂停程序。

6.2 设置观察窗口和曲线图

1. 将一个观察窗口添加到工作区；添加变量，并将其设定为使用如图 14 中所示的正确格式。在构建 3 中时，这个观察窗口中的变量被用来控制和监控电路板的状态。







Name	Value	Address	Type	Format
X+Y =? Period				Q-Value(14)
X+Y =? Gui_Vset				Q-Value(9)
X+Y =? LLC_Enable				Natural
X+Y =? Gui_Vout				Q-Value(9)
X+Y =? Min_Period				Q-Value(14)
X+Y =? Max_Period				Q-Value(14)
X+Y =? RED				Natural
X+Y =? FED				Natural
X+Y =? REM1				Natural
X+Y =? FEM1				Natural
X+Y =? REM2				Natural
X+Y =? FEM2				Natural
X+Y =? Pgain				Natural
X+Y =? Igain				Natural
X+Y =? Dgain				Natural
X+Y =? SR_Enable				Natural
X+Y =? Gui_I_SR1				Q-Value(8)
X+Y =? Gui_I_SR2				Q-Value(8)
X+Y =? COMP1				Natural
X+Y =? COMP2				Natural
X+Y =? Comp_Enable				Natural
<new>				

图 14. 为构建 3 配置观察窗口

2. 单击观察窗口中的 Continuous Refresh (持续刷新) 按钮。 

6.3 运行代码

1. 通过按下 Debug Tab (调试标签页) 内的 Run 按钮来运行代码 。
2. 通过将“LLC_Enable”和“SR_Enable”置位为 1 来启用 PWM。
3. 使用一个示波器来探测 SR PWM 和 SR 电流信号。
4. 将负载设定为 1A 的最小值。
5. 将主电源斜升至 390 VDC。
6. 将“Gui_Vset”设定为 12 VDC。
7. 将负载设定为 10A。
8. 在示波器上, 在 SR PWM 信号的下降边沿相对于 SR 电流下降返回为 0 的地方做一个标记。
9. 将“Comp_Enable”置位为 1。观察 SR PWM 信号的下降边沿是如何位移的。现在 SR PWM 信号的下降边沿应该靠近 SR 电流落回至 0 的地方。
10. 改变负载的试验。请注意, SR PWM 信号的下降边沿跟踪 SR 电流在何处落回为 0。
11. 改变“COMP(X)”值时进行的试验。这将影响模拟比较器允许 SR PWM 关闭时的电流电平。
12. 一旦完成, 关闭或将主电源设定为 0 VDC。
13. 禁用实时模式  并复位处理器  (Target → Reset → Reset CPU), 然后通过单击  (Target → Terminate All) 来终止所有调试会话。

注: 模拟比较器被用来延迟 SR PWM 的下降边沿, 并且不会将下降边沿推进到超过“FEM(X)”所规定的位置。

6.4 试验结果

在使用高压半桥 LLC 谐振 DC/DC 转换器套件进行试验时，这个部分包含多种不同的可用作基准的试验结果。

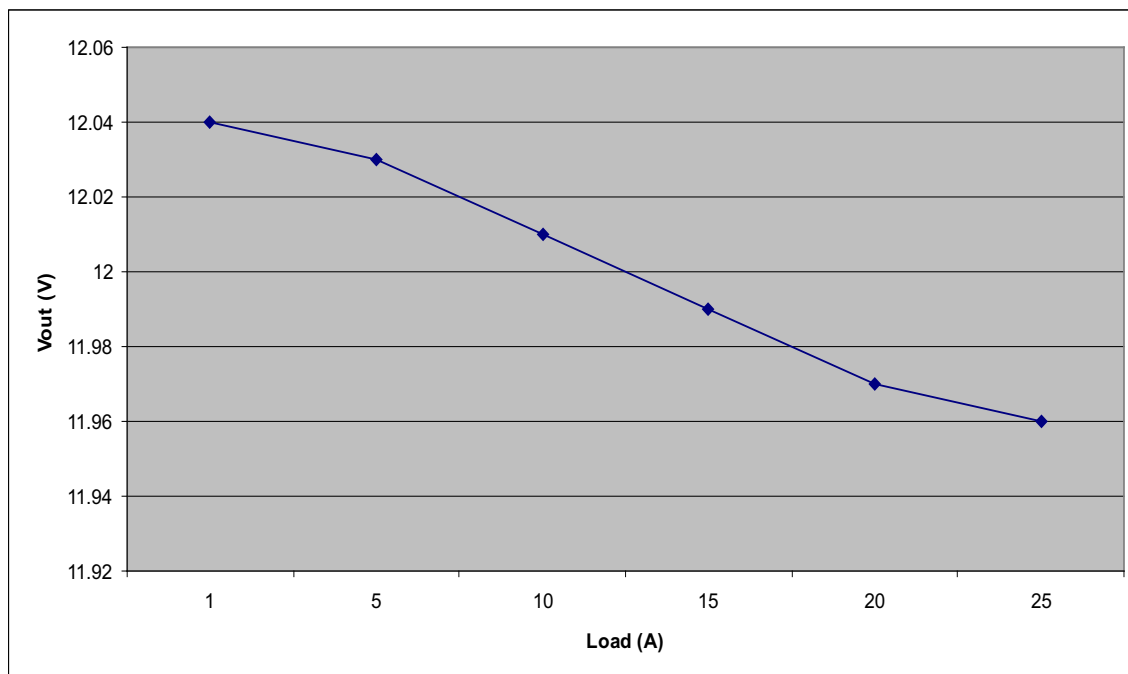


图 15. 输出电压：负载调节 ($V_{\text{输入}} = 390 \text{ VDC}$)

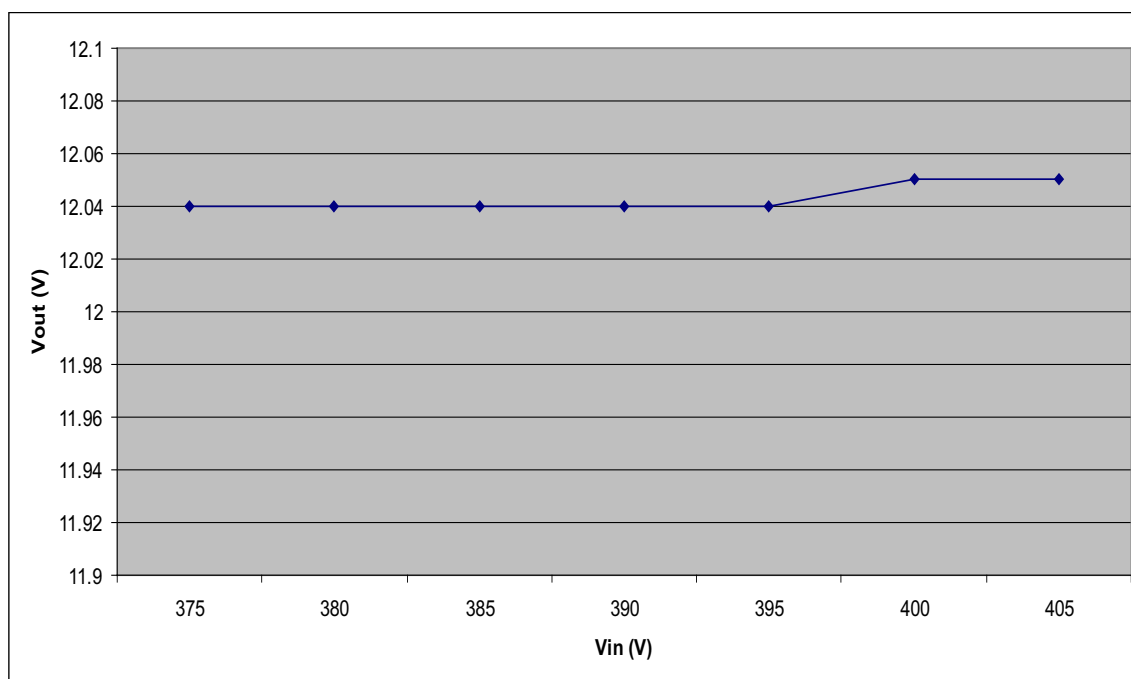


图 16. 输出电压：线路调节 ($I_o = 1 \text{ A}$)

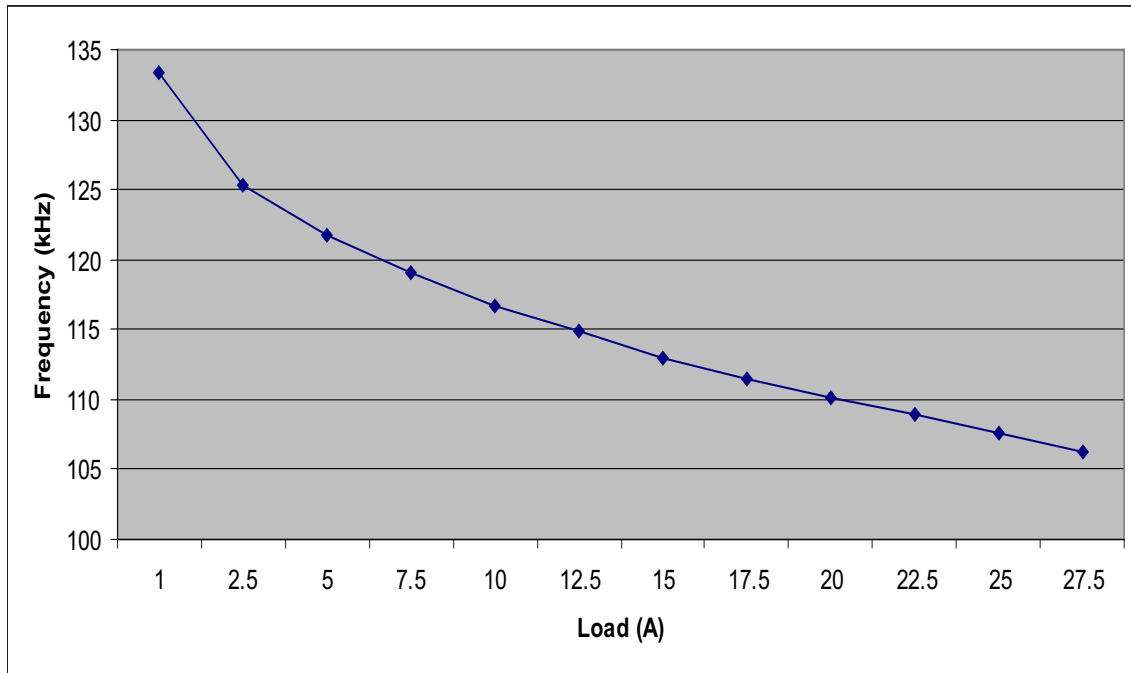


图 17. 开关频率与负载之间的关系 ($V_{\text{输入}} = 390 \text{ VDC}$)

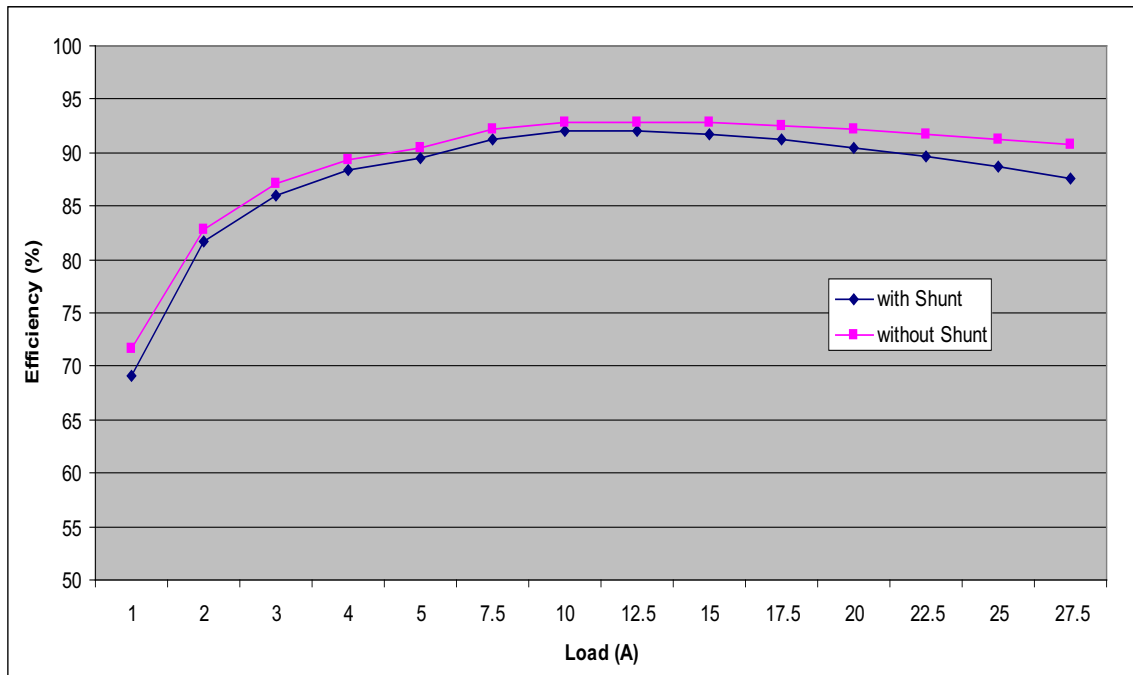


图 18. 效率与负载之间的关系 ($V_{\text{输入}} = 390 \text{ VDC}$)

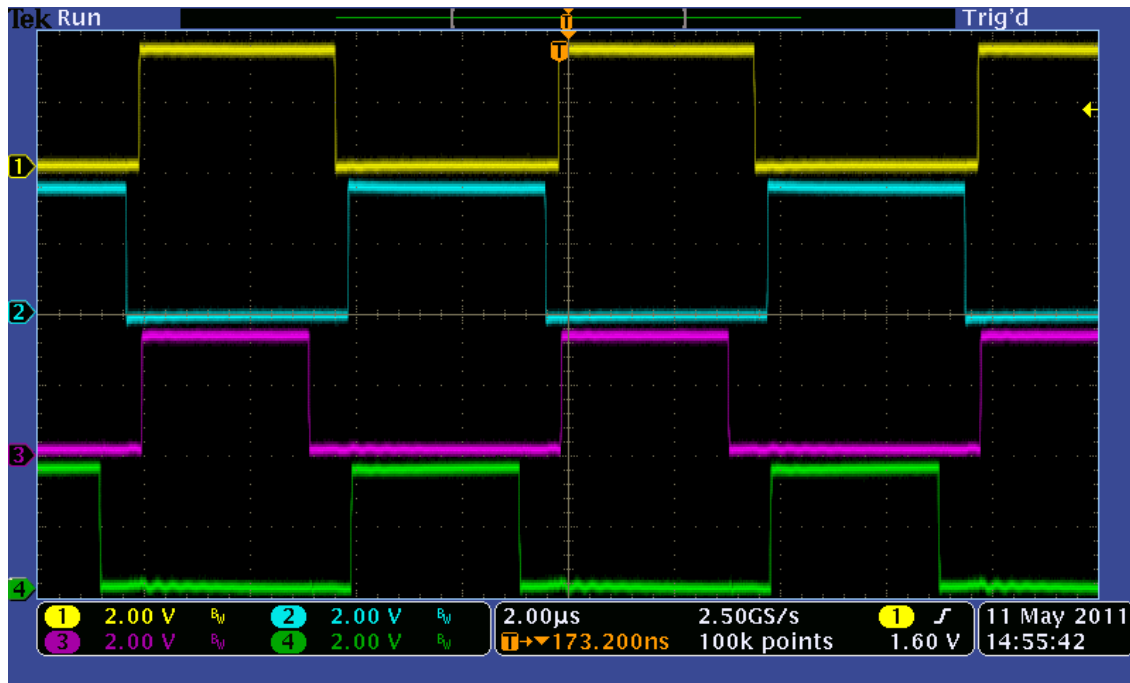


图 19. PWM 时序示例 (黄色 - HB HS, 蓝色 - HB LS, 紫色 - SR2, 绿色 - SR1)

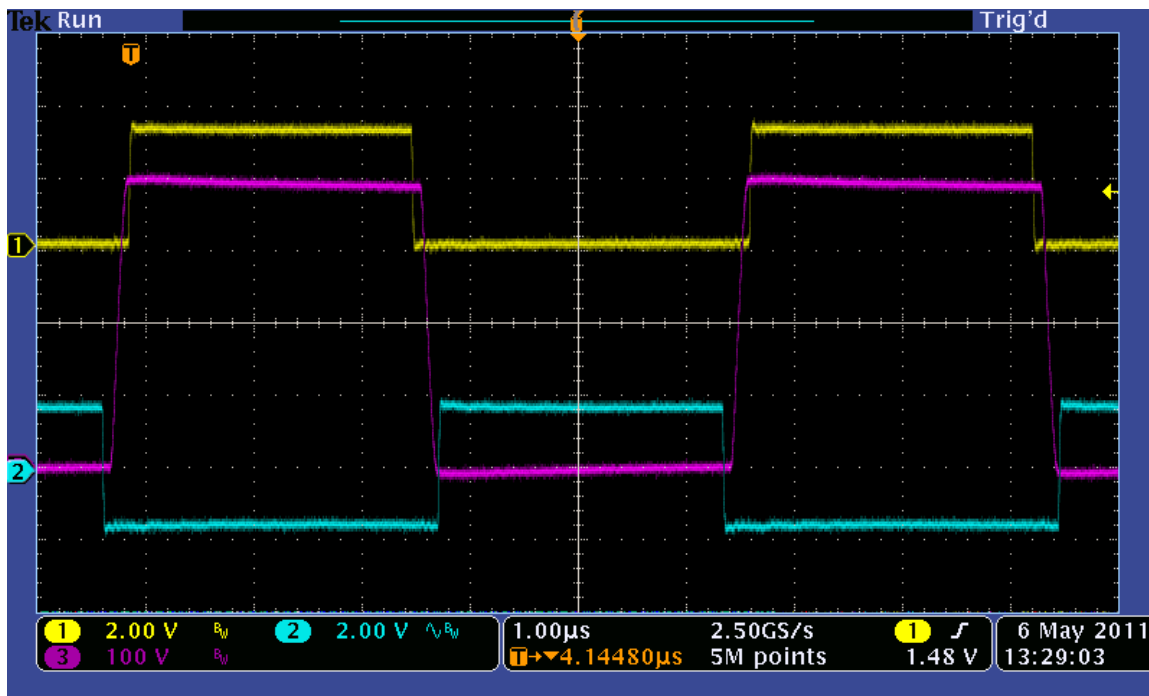


图 20. 半桥零电压开关 (黄色 - HB HS PWM, 蓝色 - HB LS PWM, 紫色 - MP 电压)

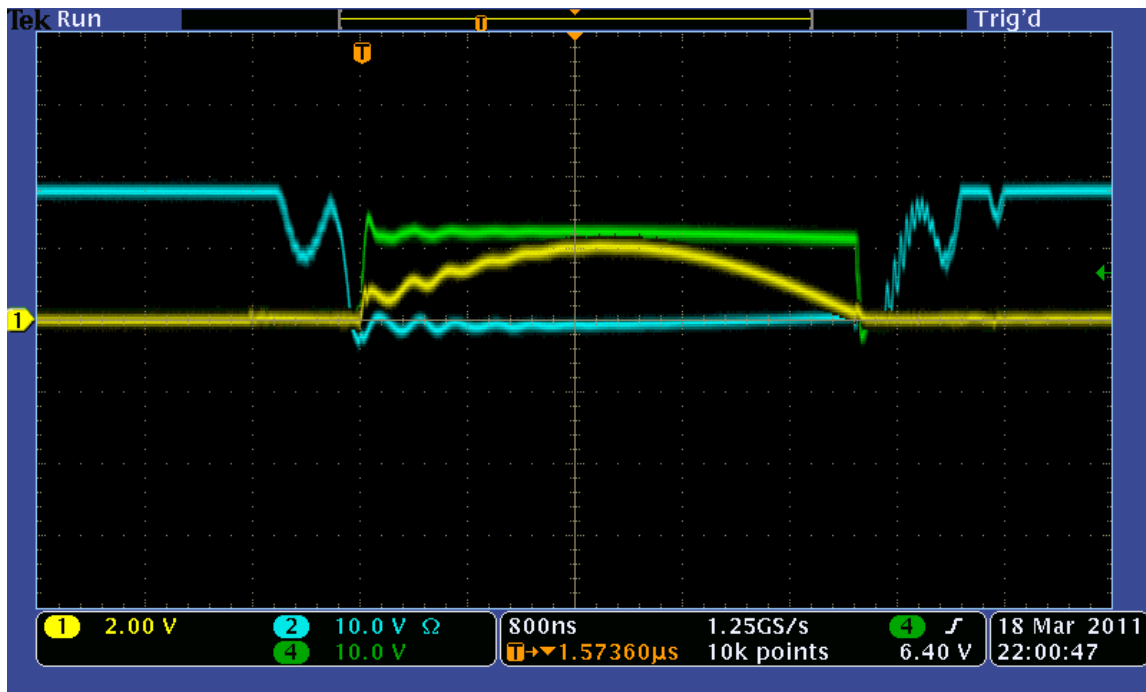


图 21. 整流器零电流开关 (黄色 - SR 电流, 蓝色 - SR Mosfet Vds, 绿色 - SR PWM)

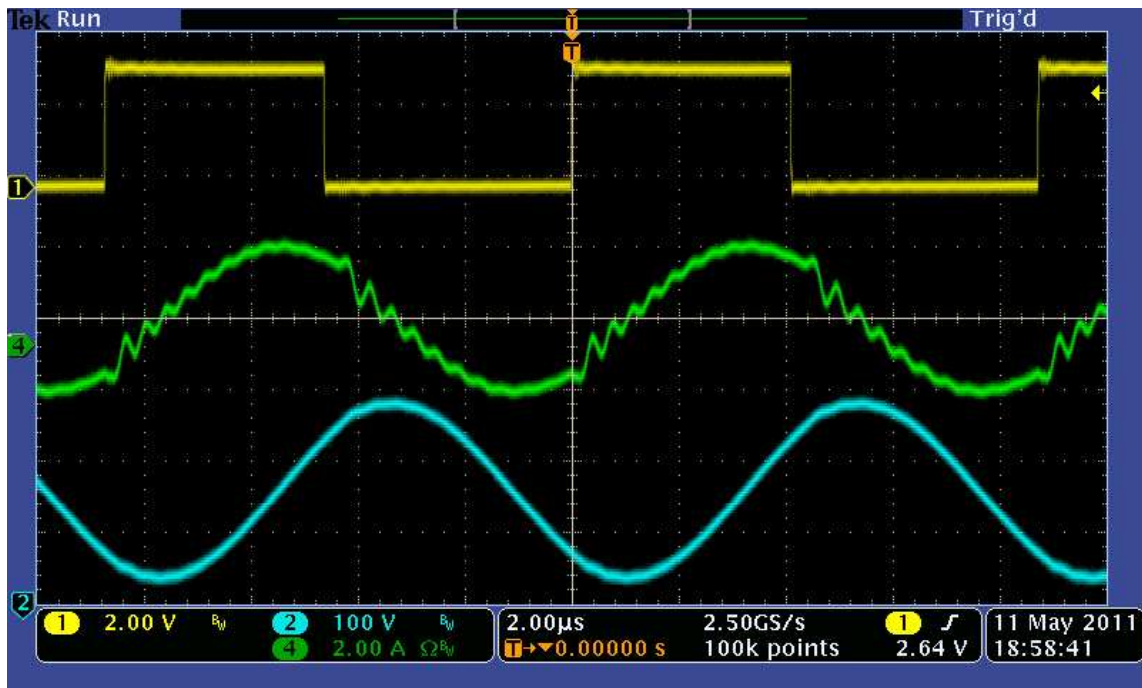


图 22. 谐振回路运行波形 (黄色 - HB HS PWM, 绿色 - 回路电流, 蓝色 - 电容器电压)

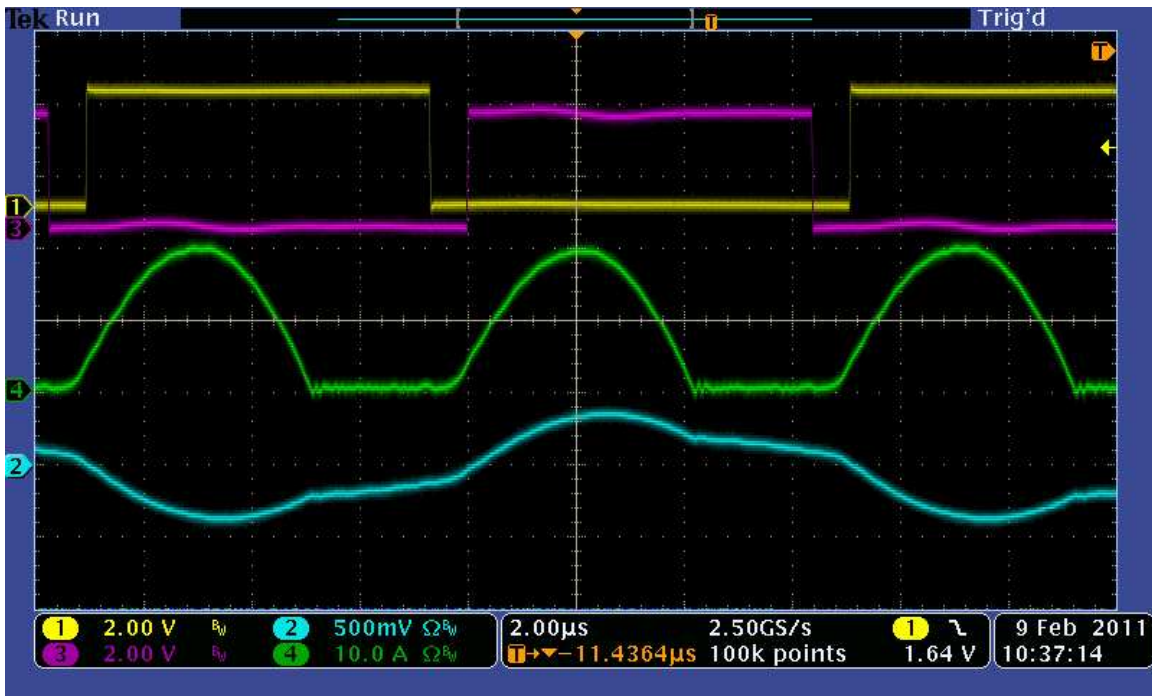


图 23. 运行电流波形（黄色 - HB HS PWM，紫色 - HB LS PWM，绿色 - 组合 SR 电流，蓝色 - 回路电流）

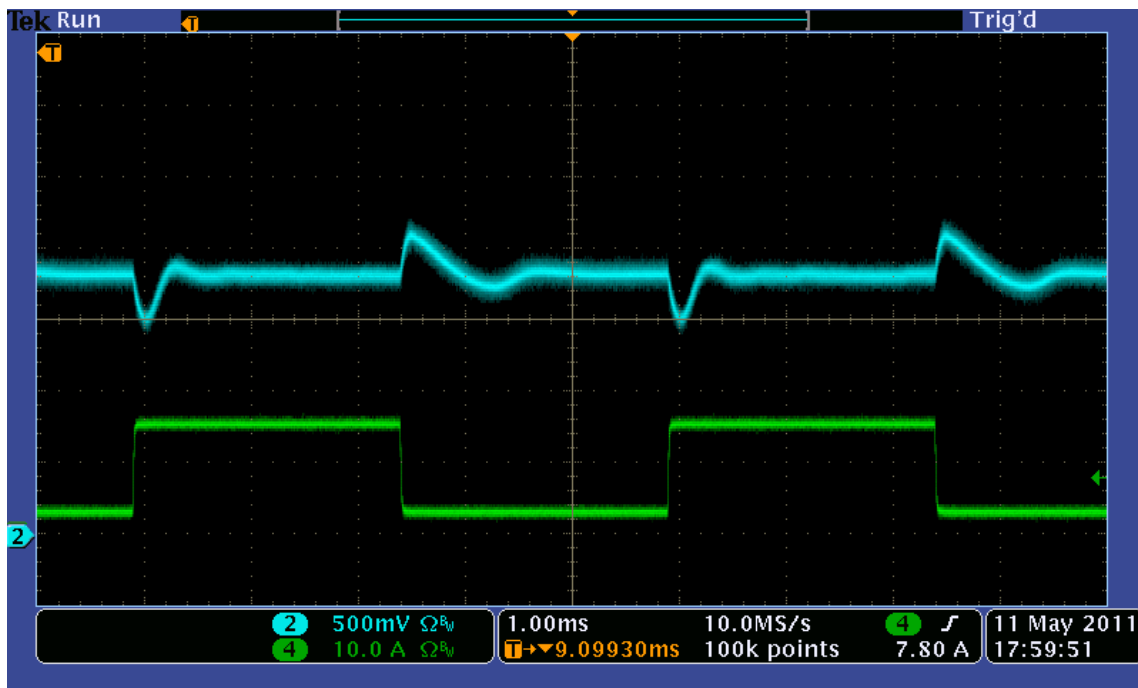


图 24. 瞬态响应：10% ↔ 60% 负载阶跃（蓝色 - 输出电压，绿色 - 负载电流）

注： 使用 2P2Z 系数和 2790µF 输出电容来取得瞬态结果。

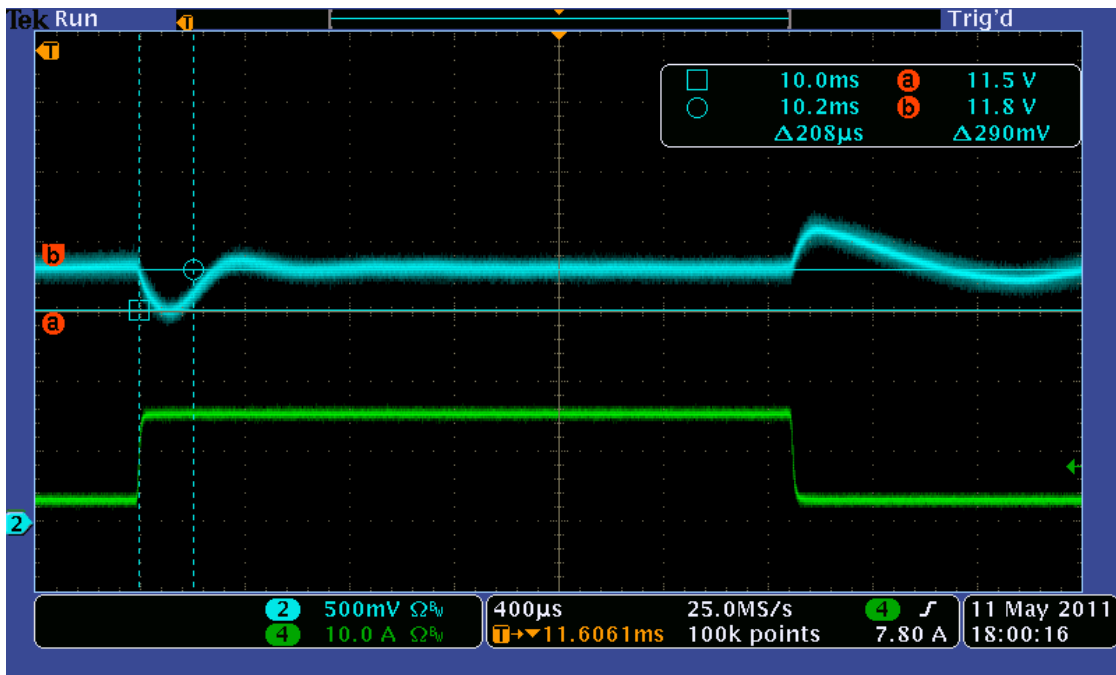


图 25. 瞬态响应: 10% → 60% 负载阶跃 (蓝色 - 输出电压, 绿色 - 负载电流)

注: 使用 2P2Z 系数和 2790µF 输出电容来取得瞬态结果。

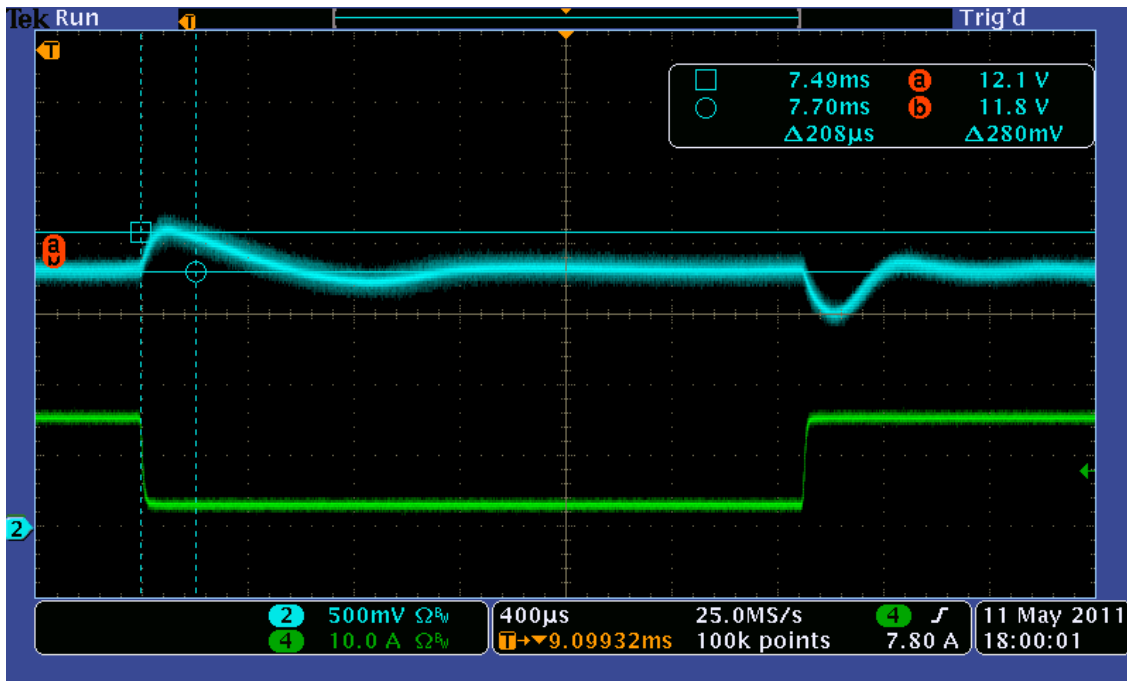


图 26. 瞬态响应: 60% → 10% 负载阶跃 (蓝色 - 输出电压, 绿色 - 负载电流)

注: 使用 2P2Z 系数和 2790µF 输出电容来取得瞬态结果。

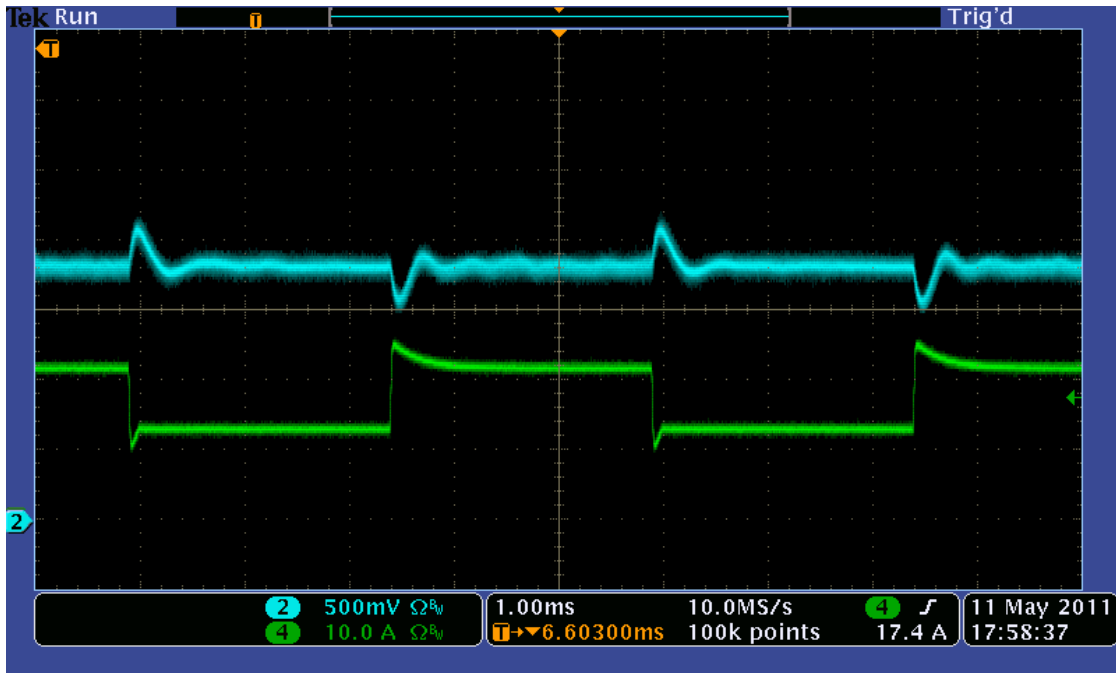


图 27. 瞬态响应: 50% ↔ 100% 负载阶跃 (蓝色 - 输出电压, 绿色 - 负载电流)

注: 使用 2P2Z 系数和 2790 μ F 输出电容来取得瞬态结果。

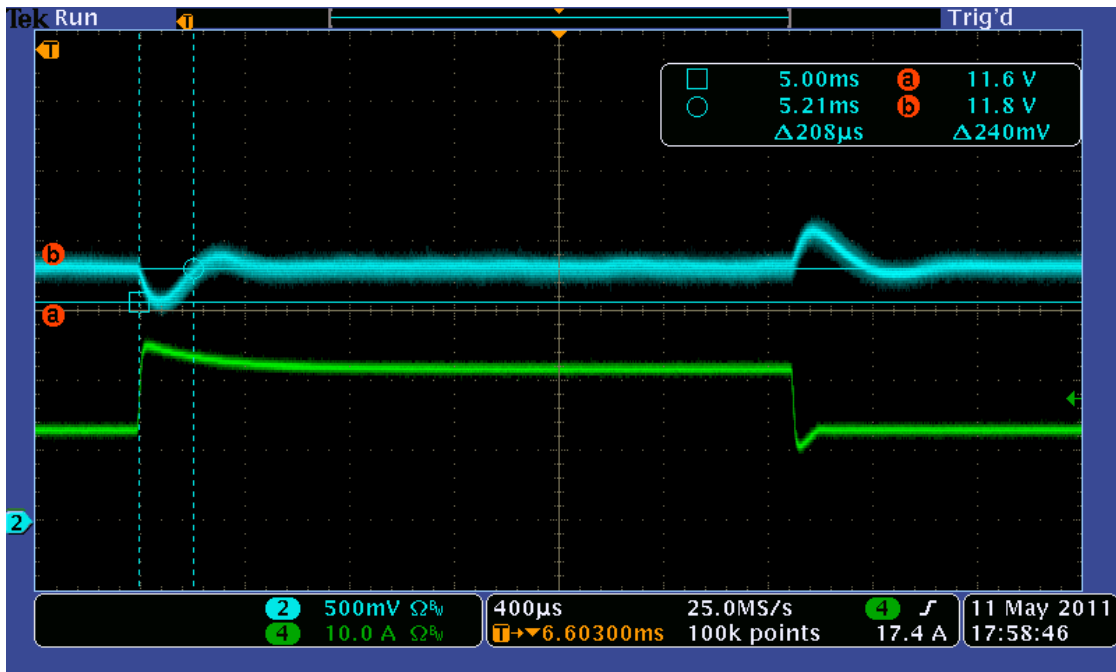


图 28. 瞬态响应: 50% → 100% 负载阶跃 (蓝色 - 输出电压, 绿色 - 负载电流)

注: 使用 2P2Z 系数和 2790 μ F 输出电容来取得瞬态结果。

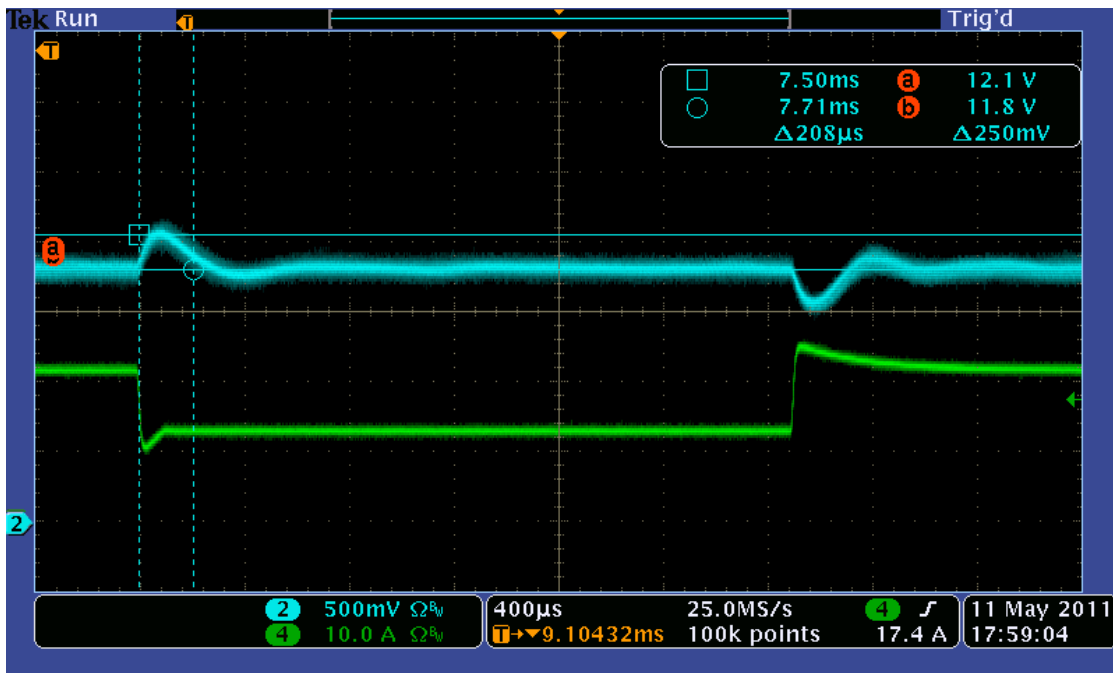


图 29. 瞬态响应: 100% → 50% 负载阶跃 (蓝色 - 输出电压, 绿色 - 负载电流)

注: 使用 2P2Z 系数和 2790µF 输出电容来取得瞬态结果。

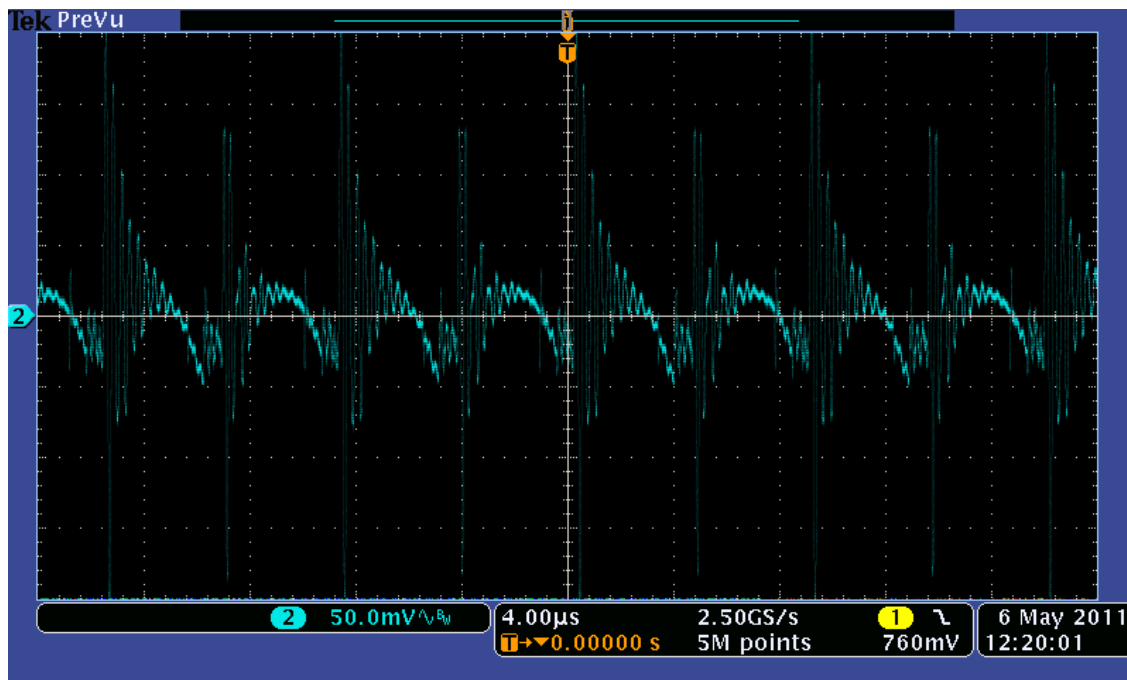


图 30. 高频电压纹波 (蓝色 - 输出电压)

7 参考书目

要获得更多信息，请见位于以下链接内的指南：www.ti.com/controlsuite

- HVLLC-SWGuide - 提供与 CCS4 项目相关的详细信息。
\\TMDSHVRESLLCKIT_v1.0\~Docs\ HVLLC-SWGuide[R3].pdf
- HVLLC-HWGuide - 提供与电路板上的硬件相关的详细信息。
\\TMDSHVRESLLCKIT_v1.0\~Docs\ HVLLC-HWGuide[R3].pdf
- HVLLC-HWdevPkg - 一个包含与电路板上硬件相关的文件的文件夹（电路原理图，物料清单，Gerber 文件，印刷电路板 (PCB) 布局布线等）。
\\TMDSHVRESLLCKIT_v1.0\~HVLLC-HWdevPkg[R3]\

重要声明

德州仪器(TI) 及其下属子公司有权根据 JESD46 最新标准, 对所提供的产品和服务进行更正、修改、增强、改进或其它更改, 并有权根据 JESD48 最新标准中止提供任何产品和服务。客户在下订单前应获取最新的相关信息, 并验证这些信息是否完整且是最新的。所有产品的销售都遵循在订单确认时所提供的TI 销售条款与条件。

TI 保证其所销售的组件的性能符合产品销售时 TI 半导体产品销售条件与条款的适用规范。仅在 TI 保证的范围内, 且 TI 认为有必要时才会使用测试或其它质量控制技术。除非适用法律做出了硬性规定, 否则没有必要对每种组件的所有参数进行测试。

TI 对应用帮助或客户产品设计不承担任何义务。客户应对其使用 TI 组件的产品和应用自行负责。为尽量减小与客户产品和应用相关的风险, 客户应提供充分的设计与操作安全措施。

TI 不对任何 TI 专利权、版权、屏蔽作品权或其它与使用了 TI 组件或服务的组合设备、机器或流程相关的 TI 知识产权中授予的直接或隐含权作出任何保证或解释。TI 所发布的与第三方产品或服务有关的信息, 不能构成从 TI 获得使用这些产品或服务的许可、授权、或认可。使用此类信息可能需要获得第三方的专利权或其它知识产权方面的许可, 或是 TI 的专利权或其它知识产权方面的许可。

对于 TI 的产品手册或数据表中 TI 信息的重要部分, 仅在没有对内容进行任何篡改且带有相关授权、条件、限制和声明的情况下才允许进行复制。TI 对此类篡改过的文件不承担任何责任或义务。复制第三方的信息可能需要服从额外的限制条件。

在转售 TI 组件或服务时, 如果对该组件或服务参数的陈述与 TI 标明的参数相比存在差异或虚假成分, 则会失去相关 TI 组件或服务的所有明示或暗示授权, 且这是不正当的、欺诈性商业行为。TI 对任何此类虚假陈述均不承担任何责任或义务。

客户认可并同意, 尽管任何应用相关信息或支持仍可能由 TI 提供, 但他们将独力负责满足与其产品及其应用中使用的 TI 产品相关的所有法律、法规和安全相关要求。客户声明并同意, 他们具备制定与实施安全措施所需的全部专业技术和知识, 可预见故障的危险后果、监测故障及其后果、降低有可能造成人身伤害的故障的发生机率并采取适当的补救措施。客户将全额赔偿因在此类安全关键应用中使用任何 TI 组件而对 TI 及其代理造成的任何损失。

在某些场合中, 为了推进安全相关应用有可能对 TI 组件进行特别的促销。TI 的目标是利用此类组件帮助客户设计和创立其特有的可满足适用的功能安全性标准和要求的终端产品解决方案。尽管如此, 此类组件仍然服从这些条款。

TI 组件未获得用于 FDA Class III (或类似的生命攸关医疗设备) 的授权许可, 除非各方授权官员已经达成了专门管控此类使用的特别协议。

只有那些 TI 特别注明属于军用等级或“增强型塑料”的 TI 组件才是设计或专门用于军事/航空应用或环境的。购买者认可并同意, 对并非指定面向军事或航空航天用途的 TI 组件进行军事或航空航天方面的应用, 其风险由客户单独承担, 并且由客户独力负责满足与此类使用相关的所有法律和法规要求。

TI 已明确指定符合 ISO/TS16949 要求的产品, 这些产品主要用于汽车。在任何情况下, 因使用非指定产品而无法达到 ISO/TS16949 要求, TI 不承担任何责任。

	产品		应用
数字音频	www.ti.com.cn/audio	通信与电信	www.ti.com.cn/telecom
放大器和线性器件	www.ti.com.cn/amplifiers	计算机及周边	www.ti.com.cn/computer
数据转换器	www.ti.com.cn/dataconverters	消费电子	www.ti.com.cn/consumer-apps
DLP® 产品	www.dlp.com	能源	www.ti.com.cn/energy
DSP - 数字信号处理器	www.ti.com.cn/dsp	工业应用	www.ti.com.cn/industrial
时钟和计时器	www.ti.com.cn/clockandtimers	医疗电子	www.ti.com.cn/medical
接口	www.ti.com.cn/interface	安防应用	www.ti.com.cn/security
逻辑	www.ti.com.cn/logic	汽车电子	www.ti.com.cn/automotive
电源管理	www.ti.com.cn/power	视频和影像	www.ti.com.cn/video
微控制器 (MCU)	www.ti.com.cn/microcontrollers		
RFID 系统	www.ti.com.cn/rfidsys		
OMAP应用处理器	www.ti.com.cn/omap		
无线连通性	www.ti.com.cn/wirelessconnectivity	德州仪器在线技术支持社区	www.deyisupport.com

邮寄地址: 上海市浦东新区世纪大道 1568 号, 中建大厦 32 楼 邮政编码: 200122
Copyright © 2013 德州仪器 半导体技术 (上海) 有限公司