应用报告
ZHCA534–May 2013

使用 C2000 微控制器的移相全桥 (PSFB) 控制

Hrishikesh Nene

摘要

移相全桥 (PSFB) 转换器用于多种应用中的直流至直流 (DC-DC) 转换，例如，在电信系统中，将一个高电压总线转换为一个中间分配电压，通常情况下更接近于 48V。由于这个拓扑结构包括一个变压器，PSFB 级提供电压转换以及线路电压的隔离。

这份应用报告介绍了数控 PSFB 系统的实施细节，此系统在德州仪器 (TI) 的高压移相全桥 (HVPSFB) 套件上执行。 这个套件将一个 400V DC 输入电压转换为一个经稳压的 12V DC 输出，并且适用于高达 600W 的运行。 对峰值电流模式控制 (PCMC) 和电压模式控制 (VMC) 实施进行了说明。 这些高度集成、基于微控制器的实现方法特有自适应零电压开关 (ZVS) 和多种不同的同步整流体系，在这里也进行了讨论。 还提供了生成这些控制体系所需的复杂栅极驱动波形的详细信息，以及在改变运行条件下优化系统性能的智能时序控制。 还包含了一个运行 HVPSFB 项目的分步指导。 这个解决方案的亮点有负载高于额定负载 10% 时的恒定高系统效率、基于片载硬件机制的全新 PCMC 波形生成，以及简单系统执行。

注: 如果您希望快速评估此套件，而又不想仔细检查实施细节，那么除了本文档，请参见随附的快速开始指南（ 在www.ti.com/controlsuite网站内的 QSG-HVPSB-Rev1.1.pdf）。

内容

1 简介 3
2 功能说明 10
3 软件概述 - PCMC 15
4 运行递增构建的过程 - PCMC 19
5 软件概述 - VMC 35
6 运行递增构建的过程 - VMC 38
7 参考书目 51

图片列表

1 一个移相全桥电路 4
2 PSFB PWM 波形 4
3 PSFB 系统方框图 5
4 效率与负载之间的关系（PCMC 和 VMC 执行） 6
5 12A 负载上的 ZVS 和 LVS 开关 (a) 有源到无源桥臂转换 (ZVS)，(b) 无源到有源桥臂转换 (LVS) 6
6 PCMC 瞬态响应 (a) 0% 至 80% 负载阶跃，(b) 80% 至 0% 负载阶跃 7
7 PCMC GUI 7
8 HVPSFB 电路板和控制器卡 8
9 TMS320F28027(Piccolo-A) 控制器卡电路原理图 8
10 HVPSFB 基板电路原理图 9
11 PCMC 方框图 10
12 PCMC PWM 波形 11
13 VMC 方框图 12

14 VMC 波形 12
15 针对输出电压感测的定点 ADC 转换触发效应 13
16 PCMC 软件流程 15
17 PCMC 软件块 16
18 PCMC 控制流程 17
19 构建 1 软件区块 20
20 C 和 C++ 项目 23
21 构建 2 软件区块 28
22 恒定电流和恒定功率软件流程图 30
23 变压器初级电压、初级感测电流以及驱动两个对角相对开关（Q1 和 Q3）的 PWM 波形 33
24 VMC 软件流程 35
25 VMC 软件区块 36
26 VMC 控制流程 37
27 构建 1 软件区块 39
28 构建 2 软件区块 47

图表列表

1 库模块 16
2 针对 PCMC 的递增构建选项 18
3 示例 RAMPMAXREF 和 DACVAL 值 21
4 HVPSFB 信号接口基准 - PCMC 21
5 库模块 35
6 针对 VMC 的递增构建选项 37
7 针对基准的相位值 39
8 HVPSFB 信号接口基准 - VMC 40

SPRABR1 — http://www-s.ti.com/sc/techlit/SPRABR1

1 简介

移相全桥 (PSFB) DC-DC 转换器经常被用于降低高 DC 总线电压或者为服务器电源、电信用整流器、电池充电系统和可再生能源系统等大功率应用提供中间隔离。 传统上来讲，微控制器已经被限制为仅仅在这些系统中执行监控或通信任务。 随着高性能微控制器器件的出现，除了处理传统微控制器功能外，现在可将微控制器用于这些系统中的封闭控制环路中。 到数字电源控制的转换意味着之前在硬件中执行的功能，现在在软件中实现。 除了灵活性，这也大大简化了系统。 这些系统能够执行高级控制策略，来在不同条件下对功率级进行最优控制，并且提供系统级智能。

一个 PSFB 控制器包含四个电源电子开关（例如金属氧化物半导体场效应晶体管 (MOSFET) 或绝缘栅双极型晶体管 (IGBT)），这些开关在隔离变压器上形成一个全桥，而二极管整流器或 MOSFET 开关用于次级侧上的同步整流 (SR)。 这个拓扑结构可使得所有开关器件随着 ZVS 进行切换，从而实现具有更低开关损耗的高效转换器。 在这个操作中，通过根据负载情况来改变针对初级侧开关的死区时间，可在整个负载范围内在全桥的一个桥臂上实现用于开关的 ZVS，而在全桥的另一个桥臂上针对开关实现零电压或低电压或者谷值切换。

对于这样一个隔离式拓扑结构，需要在次级测上进行信号整流。 对于具有低输出电压或高输出电流额定值的系统，执行同步整流而不执行二极管整流，可通过避免二极管整流损耗来实现最佳性能。 在这个操作中，在次级测上用不同的切换机制来执行电流双同步整流，以便在变化的负载条件下实现最佳性能。

可在诸如 VMC，平均电流模式控制 (ACMC) 或 PCMC 的多种模式中控制一个 DC-DC 转换器系统。 执行这些不同的控制模式来控制同一功率级通常需要重新设计控制电路，以及对功率级感测电路进行改动。 在使用一个基于微控制器的系统时，所有这些模式可在只需最小改动或无需改动的情况下使用同一设计进行试验。 这样一个系统在这里使用 VMC 和 PCMC 控制机制执行。

由于其固有的电压前馈、自动逐周期电流限值、磁通均衡和其它优势，PCMC 是电源转换器优先考虑的控制机制。 在 PSFB 系统内执行 PCMC 需要具有精密时序控制的负载脉宽调制 (PWM) 波形生成。 一个生成这个波形的新方法已经出现，这个方法使用德州仪器 (TI) Piccolo™ 系列 TMS320F2802x 和 TMS320F2803x 微控制器，而无需任何额外支持电路。 独特的可编程片载斜率补偿硬件被用来提供适当的斜率补偿，这个斜率补偿可确保开环稳定性并消除和限制输出上的次谐波振荡。 对于使用一个微控制器执行的 PCMC，经稳压的输出电压取决于输出电压纹波的数量，而输出电压纹波数量由负载而定。 详细解释了这个关系并提出了不同的解决方案。

借助于这里使用的 600W PSFB，可实现大于 95% 的峰值效率以及负载低至 10% 时的大于 90% 的效率。

1.1 基本操作

图 1显示了一个经简化的移相全桥电路。 MOSFET 开关 Q_A，Q_B，Q_C和 Q_D组成了变压器 T1 初级侧上的全桥。 Q_A和 Q_B在 50% 占空比时切换，此时它们之间的相移为 180°。 相似地，Q_C和 Q_D在 50% 占空比时切换，此时它们之间的相移为 180°。 针对全桥桥臂 Q_C-Q_D的 PWM 开关信号相对于那些针对桥臂 Q_A-Q_B的开关信号进行移相。 这个相移的数量决定了对角开关间的重叠量，这反过来决定了传输的能量数量。 D_1，D_2提供次级测上的二极管双整流，此时 L_o和 C_o形成了输出滤波器。 电感器 L_R帮助提供与 MOSFET 电容进行谐振操作的变压器泄露电感，并且使 ZVS 变得更加容易。 请注意，在变压器 T1 的两侧上有两个不同的接地 G_1和 G_2。

图 2提供图 1中的系统所需的开关波形。

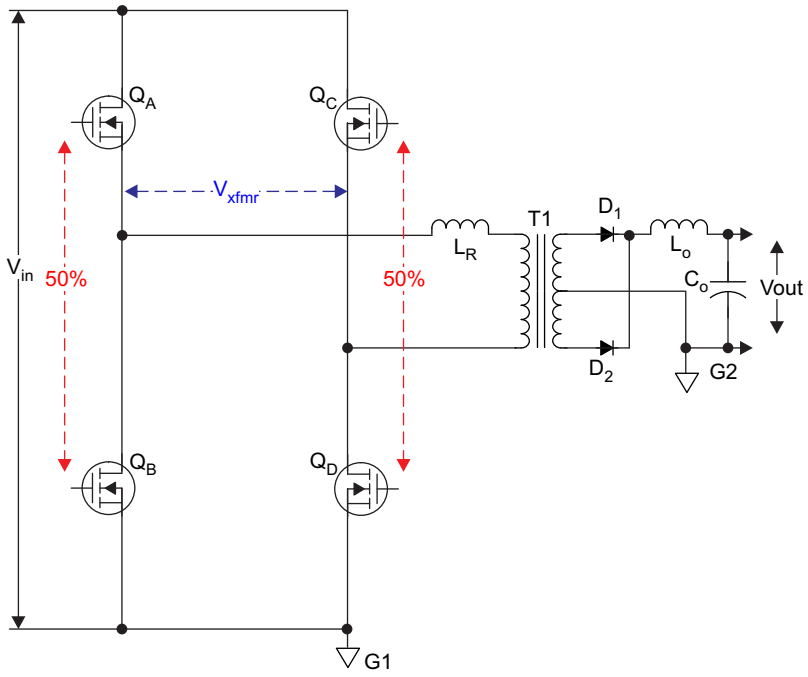


图 **1.** 一个移相全桥电路

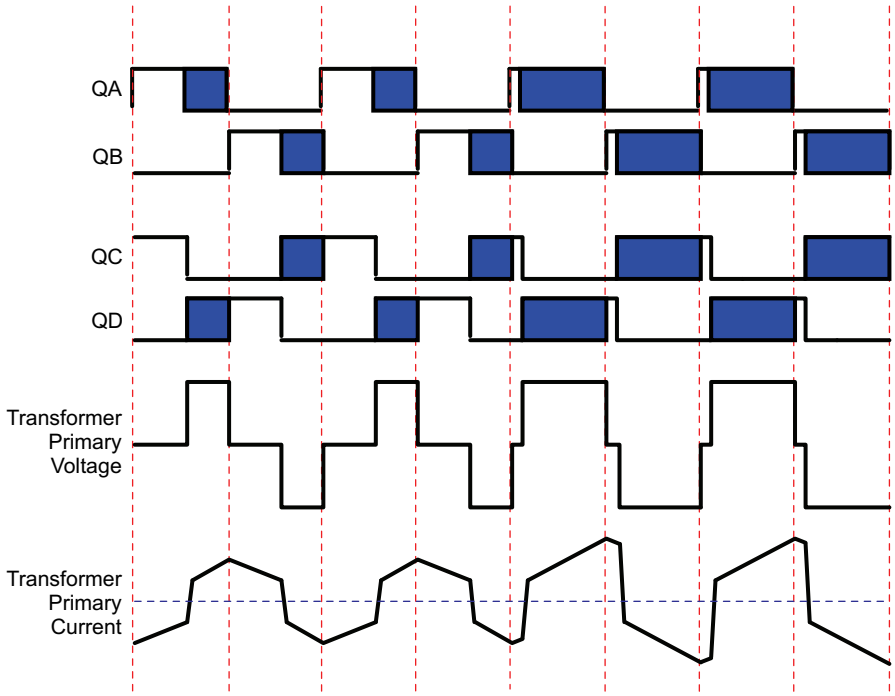


图 **2. PSFB PWM** 波形

1.2 HVPSFB 电路板上的 PSFB 实施

图 3显示了一个在 HVPSFB 电路板上执行的 PSFB 电路简化方框图。 开关 Q_A，Q_B，Q_C和 Q_D分别与开关 Q1，Q4，Q2 和 Q3图 2相对应。 开关 Q5 和 Q6 被用于次级上的同步整流。

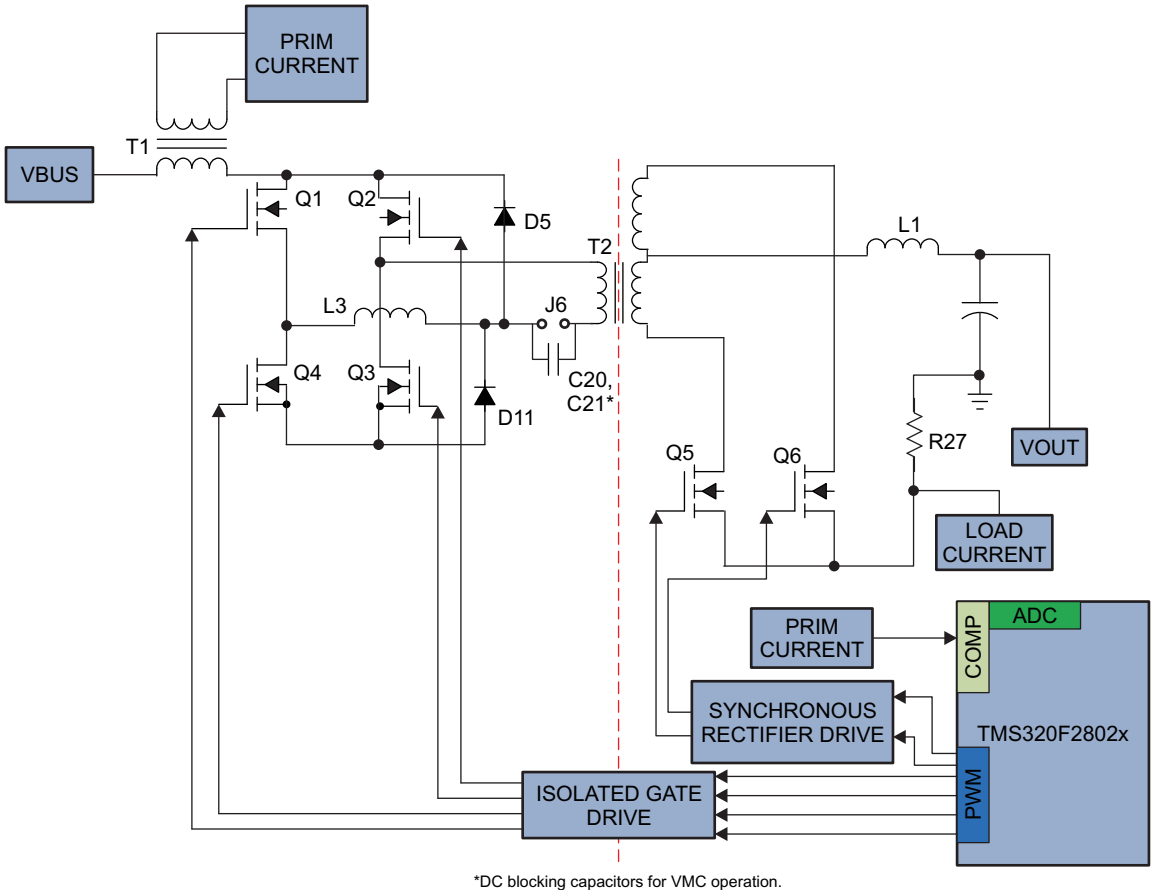


图 3. PSFB 系统方框图

此控制算法在一个 C2000™ 微控制器上 (MCU) 执行。 通过反馈信号和 PWM 输出，此 MCU 与 PSFB 功率级进行交互操作。 此控制器位于这个设计的次级侧上。 确定此控制器相对于隔离边界的位置是设计一个隔离式 DC-DC 系统时的关键一步。 对于具有多个输出轨或处理很多信号并且控制次级侧上的环路或者与应用中其它系统（在次级上）进行通信的系统，将此控制器放置在次级侧将更有利。

驱动全桥两个桥臂的 PWM 信号间的相移确定了传送给负载的能量数量。 这个相移是受控参数。

为了稳压并将输出电压保持在要求的电平上，MCU 通过控制这个相移来实现 DC-DC 转换。

在不同的运行模式中控制这样一个系统要求生成复杂的 PWM 驱动波形，以及快速且高效控制环路计算。 借助于诸如 PWM 模块，带有数模转换器 (DAC) 和斜率补偿硬件的模拟比较器，以及带有一个高效 32 位 CPU 的 12 位高速模数转换器 (ADC) 等高级片载控制外设，可在 C2000 微控制器上执行这一操作。 在下面的部分中给出了一个软件算法的详细说明。

1.3 HVPSFB 套件亮点

以下是 HVPSFB 套件的关键特性列表：

- 400V DC 输入（370Vdc 至 410Vdc 运行），12V DC 输出
- 峰值效率大于 95%。 负载低至 10% 时，效率大于 90%。

- 50A (600W) 额定输出
- 移相全桥 (PSFB) 电路拓扑
- 100kHz 开关频率
- 无针对 PCMC 功能支持电路的 PCMC
- 多个同步整流 (SR) 开关机制
- 整个负载范围内的自适应零电压开关 (ZVS) 和低压开关 (LVS)
- 允许快速且简单系统调节以实现最佳性能的高效图形用户界面 (GUI)
- 故障保护：输入欠压 (UV) 和过压 (OV)，过流，输出 UV（CC 和 CP 模式）
- 恒定电流 (CC) 和恒定功率 (CP) 功能
- 可选电压模式控制 (VMC)

图 4至图 6提供某些使用这个套件获得的结果。

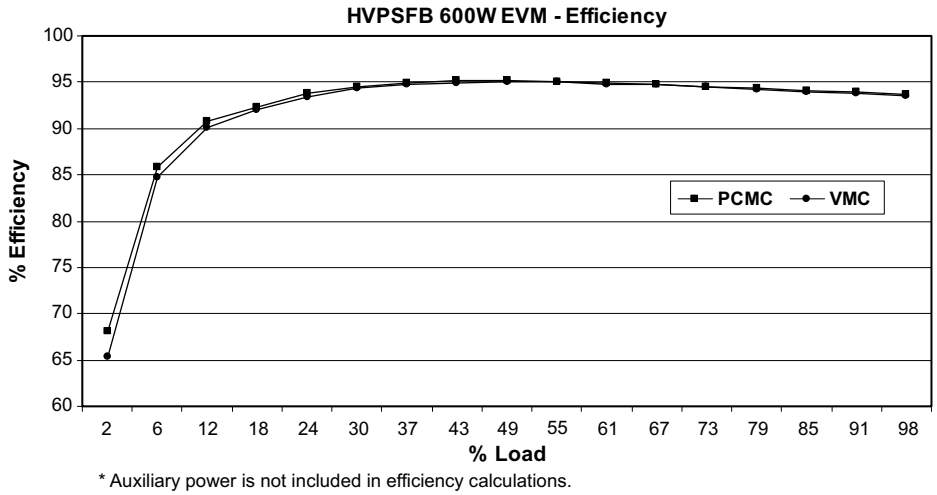


图 **4.** 效率与负载之间的关系（**PCMC** 和 **VMC** 执行）

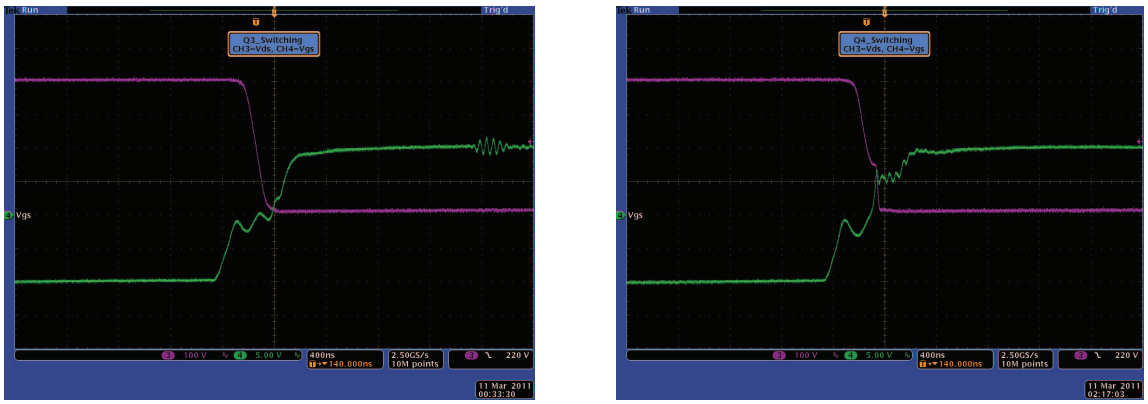


图 **5. 12A** 负载上的 **ZVS** 和 **LVS** 开关 **(a)** 有源到无源桥臂转换 **(ZVS)**，**(b)** 无源到有源桥臂转换 **(LVS)**

SPRABR1 — http://www-s.ti.com/sc/techlit/SPRABR1

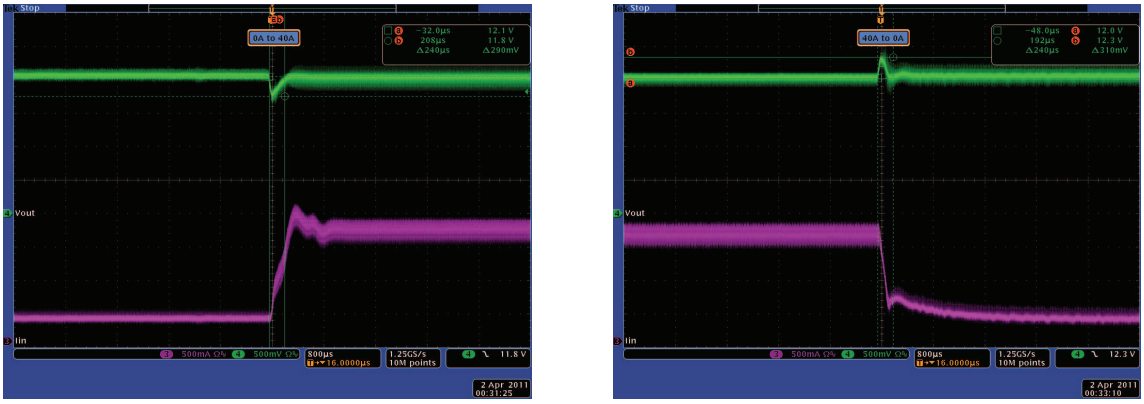


图 **6. PCMC** 瞬态响应 **(a) 0%** 至 **80%** 负载阶跃，**(b) 80%** 至 **0%** 负载阶跃

在整个运行范围内实现 ZVS 和 LVS 开关。 对于所有额定负载 10% 的负载，系统效率在 90% 以上，同时峰值效率大于 95%。 对于负载内 80% 的阶跃变化，可实现输出峰值偏差少于额定输出的 3%，而稳定时间少于 250µs。 PCMC 阶跃响应看起来好像一个阻尼减幅一阶系统。 这个基于 C2000 MCU (TMS320F2802x) 的实施能够生成并且控制 PCMC 和 VMC 控制机制所需的复杂栅极驱动波形，同时仍提供数控解决方案所特有的一定程度的智能。

图 7提供了随这个套件一同提供的软件包中 PCMC GUI 的简图。 针对 VMC 实施的独立 GUI 也包括在内。

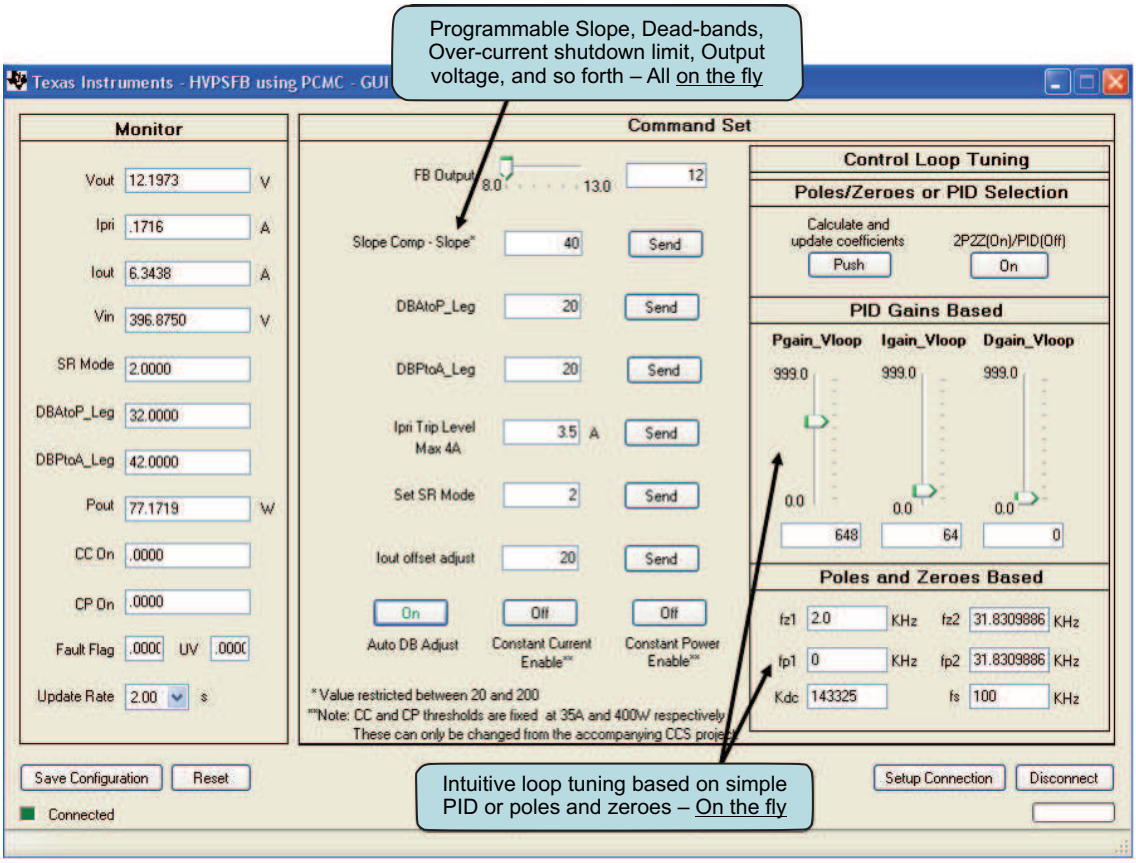


图 **7. PCMC GUI**

1.4 认识电路板上的关键组件

图 8中显示了实际硬件上的某些关键组件。

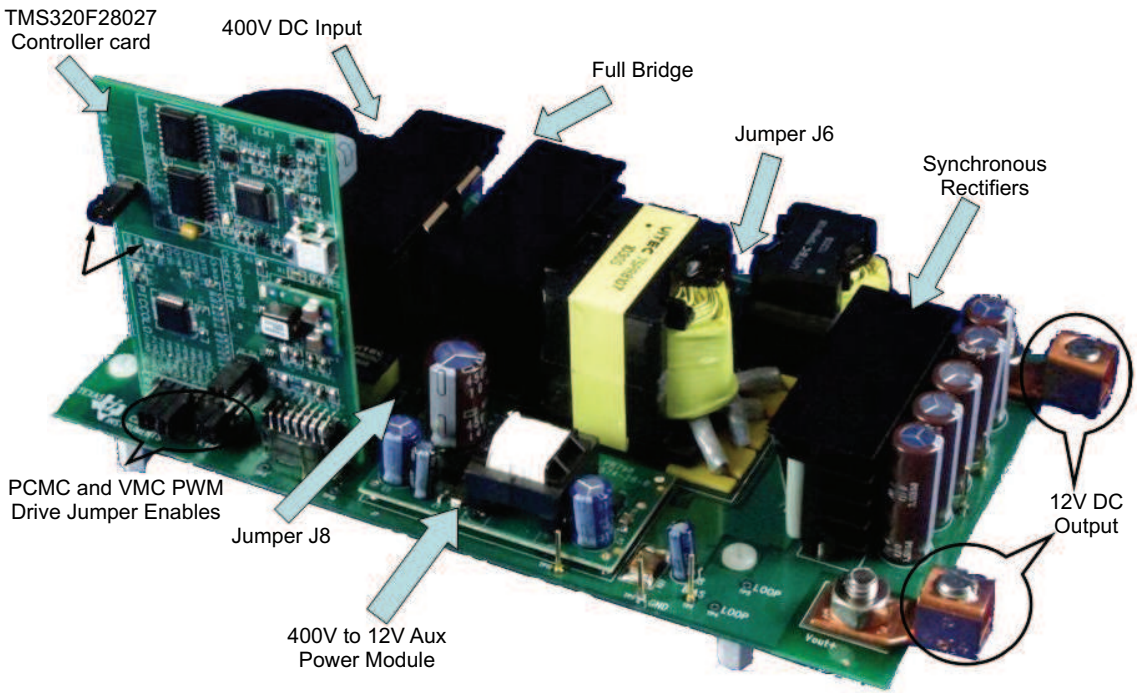


图 8. HVPSFB 电路板和控制器卡

图 9和图 10提供了 Piccolo-A 控制器卡和 HVPSFB 基板的电路原理图。

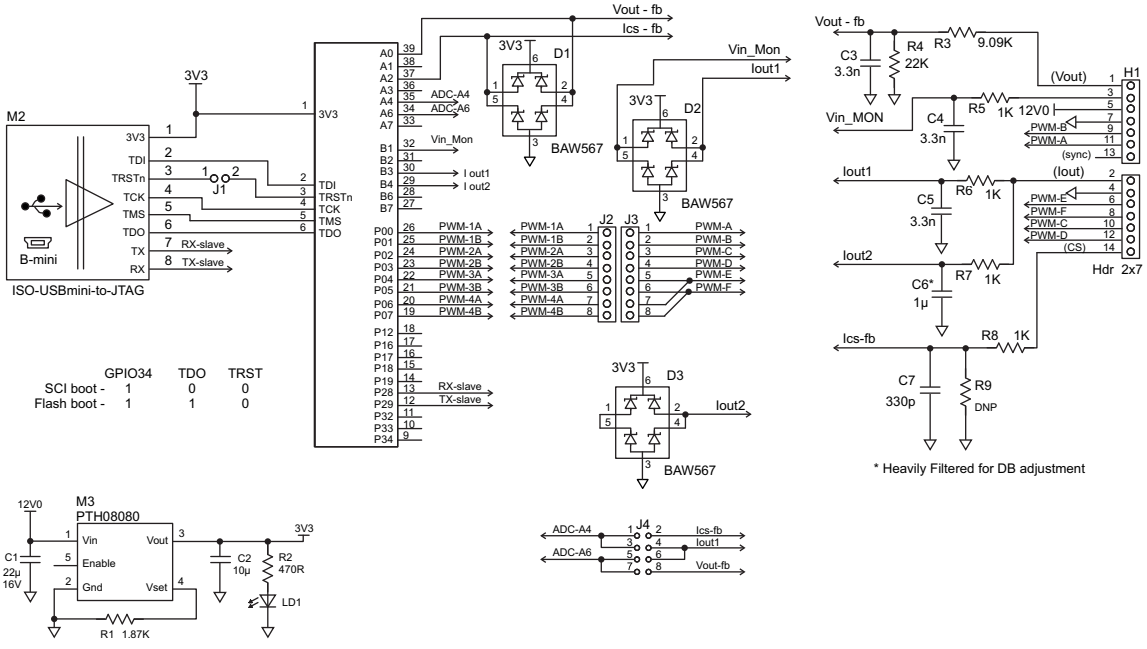


图 9. TMS320F28027(Piccolo-A) 控制器卡电路原理图

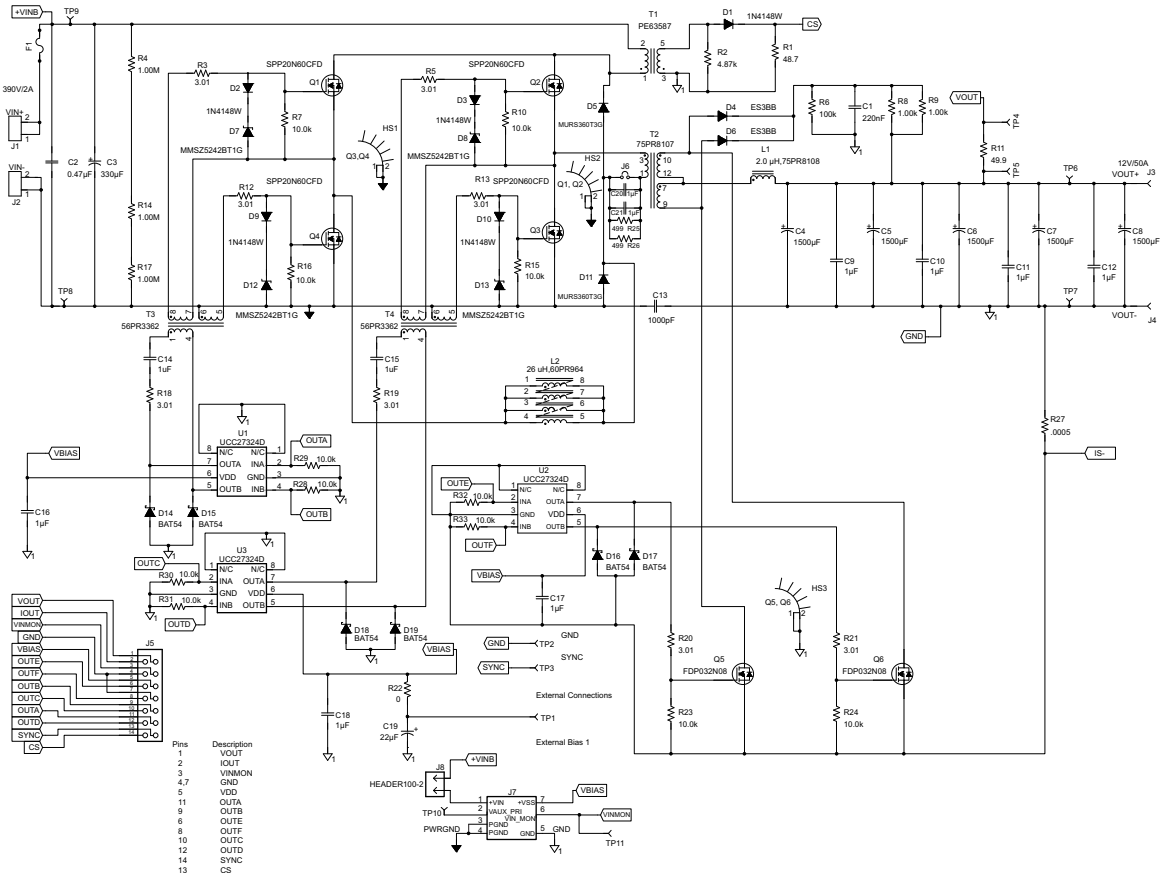


图 **10. HVPSFB** 基板电路原理图

2 功能说明

2.1 峰值电流模式控制 (PCMC)

执行针对 PSFB 系统的 PCMC 需要具有精密时序控制的复杂 PWM 波形生成。 德州仪器 (TI) 的 Piccolo 系列器件特有高级片载控制外设，这些外设使得这些实施变成可能，并且无需针对此用途的任何外部支持电路。 这些外设包括片载模拟比较器，DAC，高级 PWM 源和独特的可编程片载斜率补偿硬件。

图 11提供了一个 PCMC 实施的方框图表示法。 变压器初级电流与峰值电流基准相比较，此电流基准由电压环路使用片载模拟比较器 1 计算得出。如图 12中所示，在每半个开关周期，当变压器初级电流达到要求的峰值基准值时，驱动开关（Q2 和 Q3）的 PWM 波形中的一个在电源传输阶段的末尾被立即'复位'。 同一桥臂内的驱动其它开关的 PWM 波形在一个可编程死区窗口后被'置位'。 还应用了适当的斜率补偿，此斜率补偿添加一个具有可编程负斜率的斜升到峰值基准电流信号中。 一个桥臂内 PWM 的'复位'和'置位'操作会导致驱动两个桥臂的 PWM 信号间的相移。 因此，对角开关之间的这个相移量以及重叠量取决于峰值基准电流的数量。 峰值基准电流越高，对角开关间的重叠越长，从而使得更多的能量传输至次级。 此控制器通过控制峰值基准电流值来控制这个能量传输，进而调节输出。 因此，这个峰值基准电流是受控参数。

这个实施的一个主要特性就是同一个峰值基准电流命令在所有运行条件下被用于开关周期的全部两个半周期。 这为变压器初级提供了最优的磁通均衡，从而减少了出现饱和的机会。

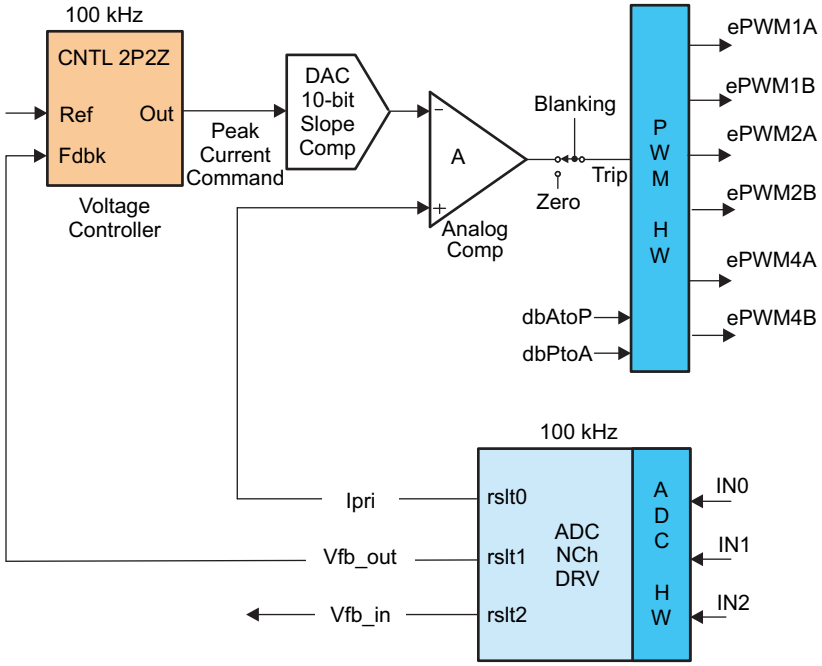


图 **11. PCMC** 方框图

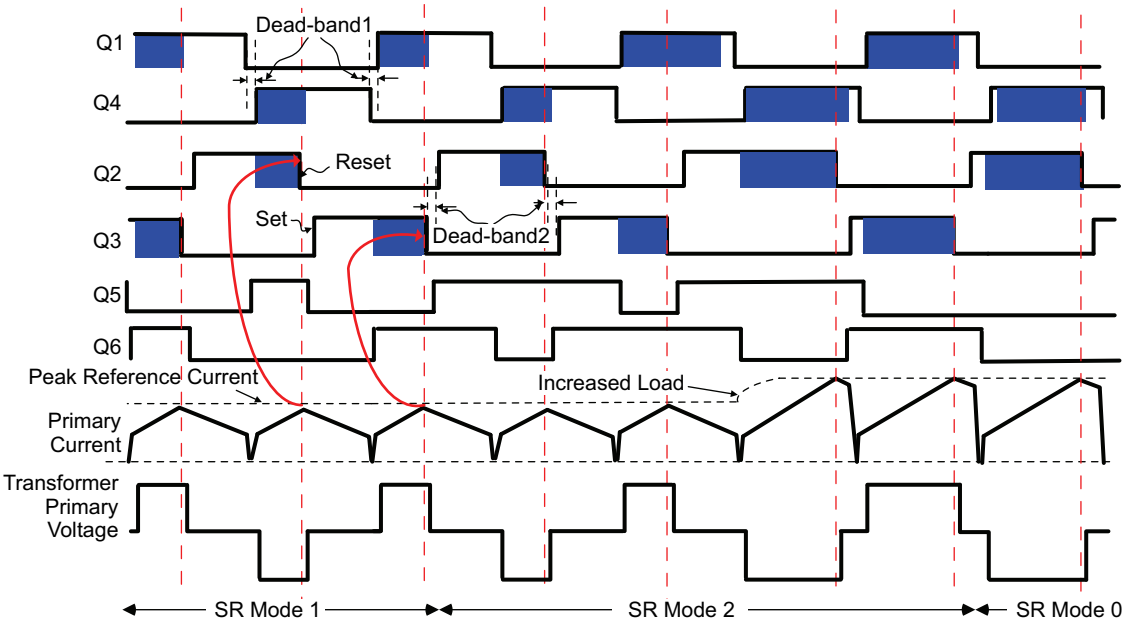


图 **12. PCMC PWM** 波形

2.2 电压模式控制 (VMC)

在 VMC 实施中，每个桥臂内的开关由具有固定 (50%) 占空比和频率的互补 PWM 信号驱动。 如图 13中所示，对于驱动另外一个桥臂内开关的 PWM 信号，此控制器直接驱动和控制 PWM 信号的相移，此相移驱动桥的一个桥臂内的开关。 这个相移控制对角位置开关的重叠量，在图 14中进行了清晰显示。 对角开关间的重叠越长，施加在变压器初级绕组上的输入电压的持续时间越长，从而使得更多能量传输至次级。 此控制器通过直接控制驱动两个桥臂的 PWM 信号间的相移来控制这个能量传输，从而调节输出。 因此，这个相移是受控参数。 应该注意的是，在使用 VMC 实施时，需要在变压器初级中包含一个 DC 隔直电容器以避免因长时间的磁通失衡而有可能导致的变压器饱和。 因此，对于 VMC 实施，应该移除图 8中的跳线 J6。

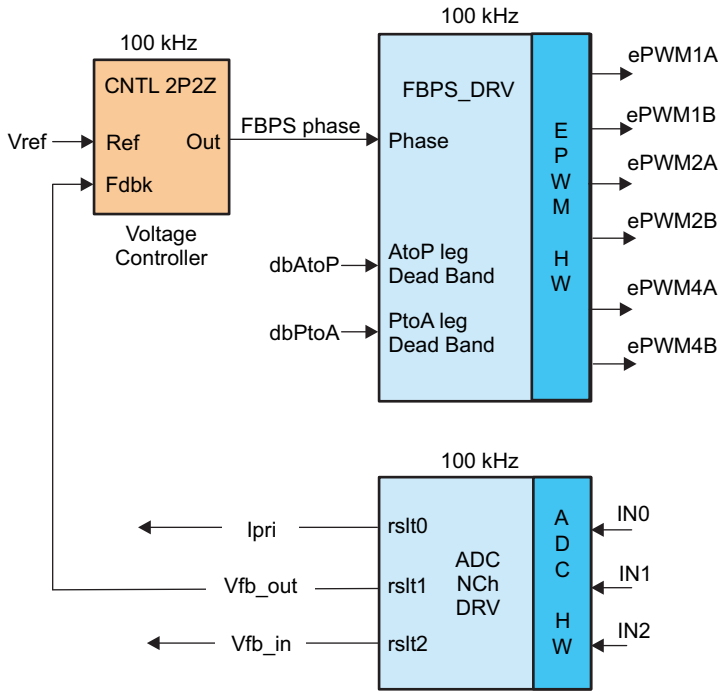


图 **13. VMC** 方框图

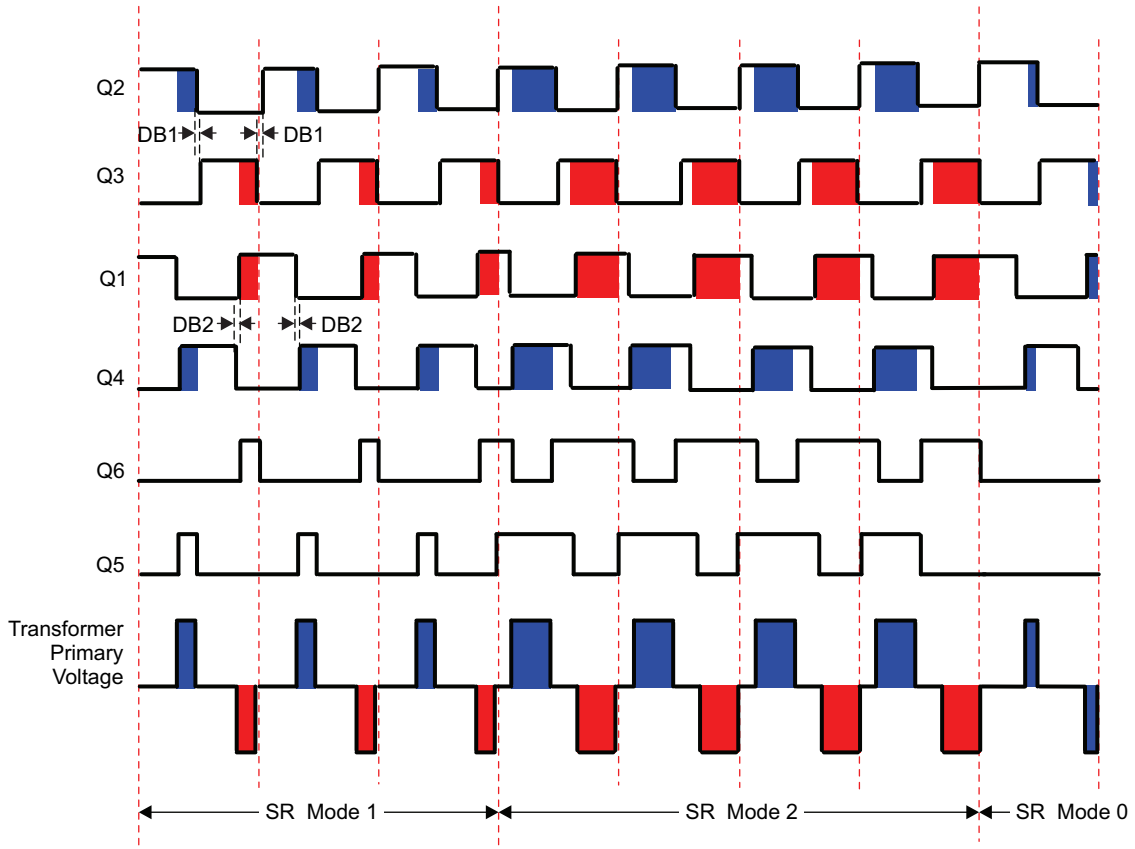


图 **14. VMC** 波形

2.3 零电压开关 (ZVS) 和低压开关 (LVS)

MOSFET 开关接通前，PSFB DC-DC 转换器使用电路中的寄生元件来确保其上的零电压，从而提供了软开关。 这大大减少了与硬开关相关的开关损耗量。

对于这里讨论的系统，针对 Q2-Q3 桥臂内开关的开关转换终止能量传输间隔。 因此，这个桥臂被称为'有源到无源'桥臂。 当在这个桥臂中发生开关转换时，初级绕组内的电流接近于它针对那半个 PWM 开关周期的最大振幅。 反射负载电流在这段时间内帮助初级电路内的能量循环，这使得这个桥臂内的开关间电压有可能接近零伏特。 在整个负载范围内，这个 Q2-Q3 桥臂内的开关有可能实现 ZVS。 应该注意的是，由于负载减少，需要增加死区时间来实现或接近 ZVS。

针对 Q1-Q4 桥臂内开关的开关转换启动电源传输间隔。 因此，这桥臂被称为'无源到有源'桥臂。 在这些开关转换期间，初级电流减少。 它跨过零电流值并改变方向。 这导致针对 ZVS 的更低可用能量。 事实上，对于低负载条件下的运行，这些开关上的电压也许在将它们打开前没有变为零值。 当这些开关上的电压处于最小值时，通过打开这些开关，可将开关损耗保持至最低。 这被称为谷值开关或低压开关 (LVS)。 当负载变化时，打开开关以实现 LVS 变化，从而要求死区时间调整的时间与 Q2-Q3 桥臂开关相似。

2.4 同步整流

在任何给定的时间，同步整理器可工作在以下 3 种模式中的任何一种模式中运行：

- 模式 0：这是通过将同步整流器保持关闭状态实现的典型二极管电流倍增器模式。 它用于极低负载运行，在此类运行中，同步整流器开关损耗大于同步整流所节省的电能。
- 模式 1：在这个模式中，同步整流器开关的运行方式与理想二极管的运行方式相类似。 当运行在极低至低负载，通常当突发模式被使用时，使用这个模式。 在这个模式中，同步整流器 MOSFET 只有当相应的对角桥驱动信号重叠时才接通。
- 模式 2：用于所有其它负载情况。 在这个模式中，同步整流器 MOSFET 只有当相应的反向对角桥驱动信号重叠时才关闭。

图 12和图 14描述了在这些模式中，用于驱动同步整流器开关所生成的波形。 在 PWM 输出上无缝执行模式转换，而不产生任何毛刺脉冲或任何异常脉冲十分重要，即使在大负载瞬态或意外相移变化指令时也是如此，以确保系统的安全运行。

2.5 变化负载的输出电压调节

经稳压的输出电压受变化的负载条件影响。 这个问题存在的原因在于输出电压上的纹波在频率为开关频率的两倍时出现。 低负载时，峰值至峰值电压纹波相对较小，而在较高负载时，它大幅增加。 如果在一个开关周期内的一个固定点上触发 ADC 转换，如图 15中所示，感测模数转换器 (ADC) 电压结果小于高负载时的值（对于同一平均电压来说）。

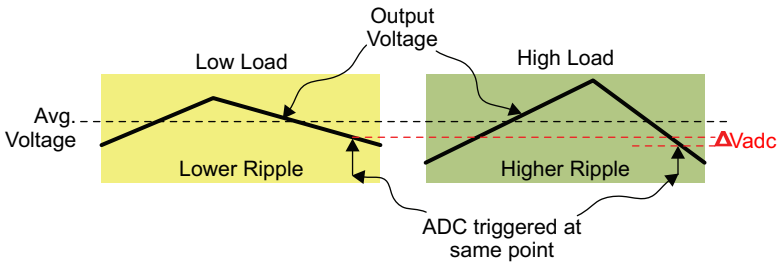


图 15. 针对输出电压感测的定点 ADC 转换触发效应

因此，对于同一平均输出电压，一个更小的输出电压对于高负载运行中的控制器可见。 此控制器针对这个 ΔVadc 进行补偿，当负载增加时，这个操作会导致实际输出电压的增加。 这里建议了几个有可能大大减小这个效应的解决方案。

- 为了直接感测平均输出电压，可在每半个 PWM 开关周期内的适当时间触发 ADC 转换启动。 在使用

PCMC 实施时，占空比不是预先确定的，而这个触发点未知。 这个方法十分适合于 VMC。

- 可在低速率时计算平均输出电压，而一个外部较慢速环路可被用来调整电压基准或反馈。 然而，这有可能影响动态性能。
- 通过在一个或两个纹波周期内对其过采样，在一个逐周期基础上计算平均输出电压。 由于平均输出电压在一个满纹波周期上被计算，避免了更高或更低峰值至峰值纹波所带来的影响。 此外，由于此平均值在一个或两个纹波周期内被计算并且被用于下一个 PWM 开关周期，动态运行状态和控制环路性能不会受到太大影响。 在使用这个方法时，在一个单 PWM 半周期内需要多个 ADC 转换。 这就是推荐用于 PCMC 实施的方法。 在这个实施中，输出电压在一个 PWM 开关周期内被过采样 8 倍。
- ADC 输入上的已增加的滤波能够减弱纹波并减少 ΔVadc。 然而，这影响了系统动态性能以及可实现的环路带宽，而此时输出电压的运行方式仍保持不变。

3 软件概述 - PCMC

3.1 软件控制流程

HVPSFB_PCMC 项目使用 “C-background/ASM-ISR” 架构。 它使用 C 语言代码作为针对应用的主要支持程序，并负责所有系统管理任务、决策制定、智能和主机交互。 汇编代码被严格限制为中断处理例程 (ISR)，此例程运行所有关键控制代码，这代码通常包括 ADC 读取、控制计算和 PWM 与 ADC 更新。图 16 描述了针对这个项目的一般软件流程。

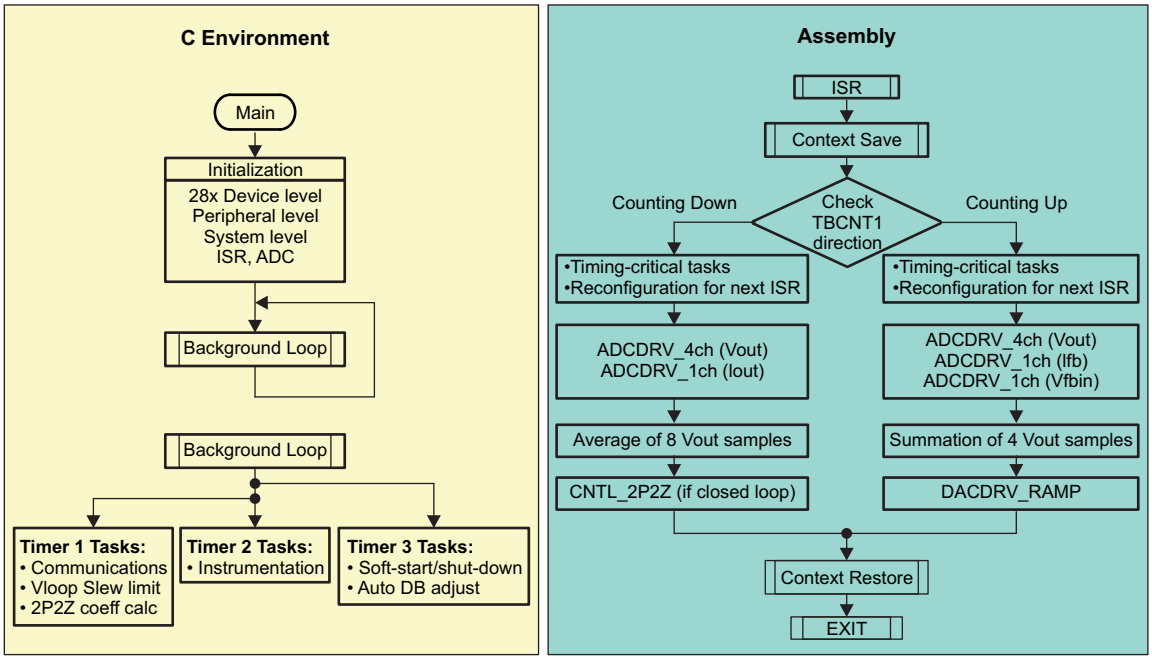


图 **16. PCMC** 软件流程

本项目中使用的关键框架 C 语言文件为：

- *HVPSFB-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”。
- *HVPSFB-DevInit.c*- 这个文件负责微控制器一次初始化和配置，并且包括诸如设置时钟，锁相环 (PLL)，通用输入/输出 (GPIO) 等功能。

ISR 包括一个单个文件：

- *HVPSFB-DPL-ISR.asm* - 这个文件包含所有时间关键“控制类型”代码。 这个文件具有一个初始化部分（一次执行）和一个运行时间部分，在频率为 PWM 开关频率两倍时执行。

电源库函数（模块）从这个框架被调用。

库模块也许具有一个 C 语言和一个汇编组件。 在这个项目中，使用了以下库模块。 在表 1中列出了 C 语言和相应的汇编模块。

表 1. 库模块

C 配置函数	**ASM** 初始化宏	**ASM** 运行时间宏
DAC_Cnf.c	DACDRV_RAMP_INIT n	DACDRV_RAMP n
ADC_SOC_Cnf.c	ADCDRV_4CH_INIT m,n,p,q	ADCDRV_4CH m,n,p,q
ADC_SOC_Cnf.c	ADCDRV_1CH_INIT n	ADCDRV_1CH n
	CNTL_2P2Z_INIT n	CNTL_2P2Z n

控制块也以图形的方式显示在图 17中。

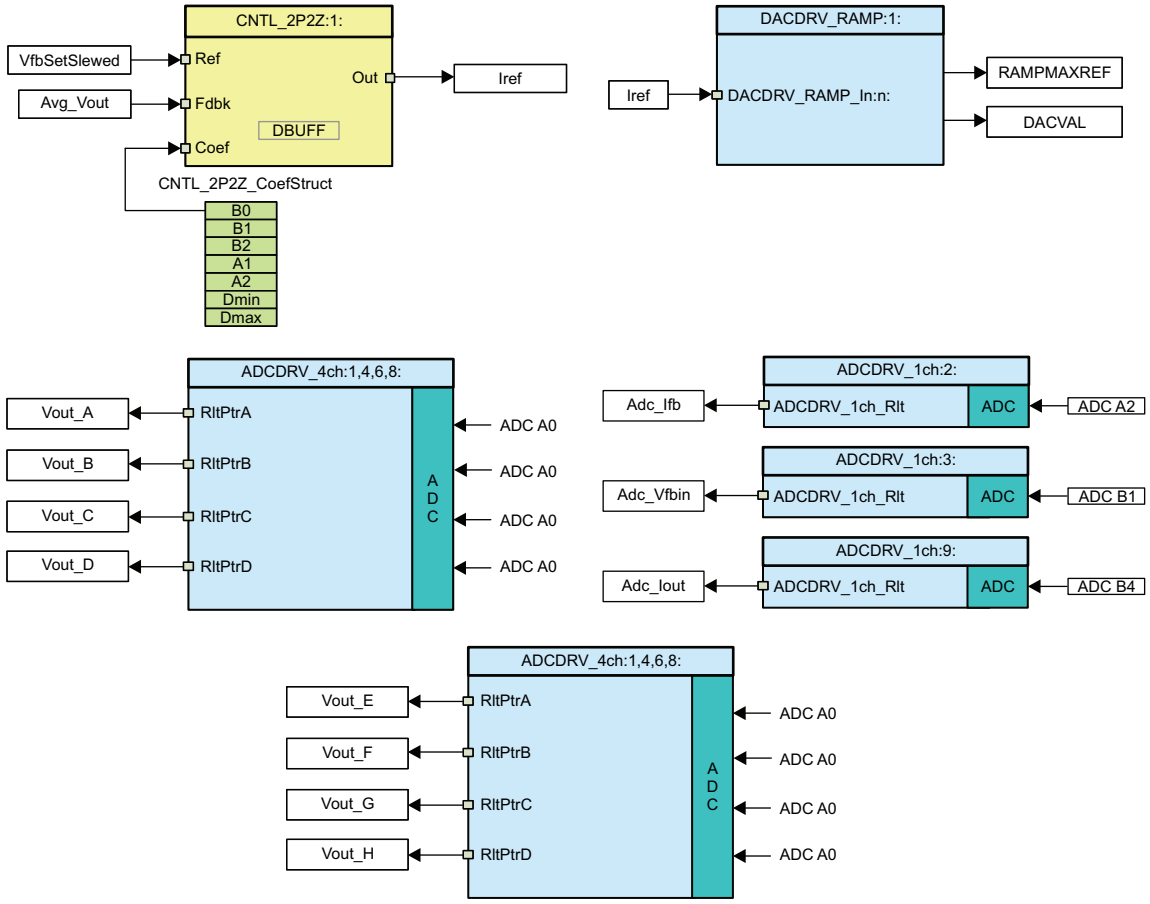


图 17. PCMC 软件块

请注意图 17中针对模块的颜色编码。 ‘深蓝色’的区块代表 C2000 微控制器控制器上的硬件模块。 ‘蓝色’区块是这些模块的软件驱动程序。 ‘黄色’区块是用于控制环路的控制器区块。 虽然在这里使用了一个 2 极 2 零控制器，对于这个应用，一个 PI 和 PID，一个 3 极 3 零或任何其它控制器也同样适用。 这样一个模块库结构使得图形化和理解图 18中显示的整个系统软件流程变得更加方便。 它还可多种功能性的简单使用、添加和删除。 通过执行一个递增构建方法可在这个项目中充分证明这一事实。 这在下一部分进行了更加详细的讨论。

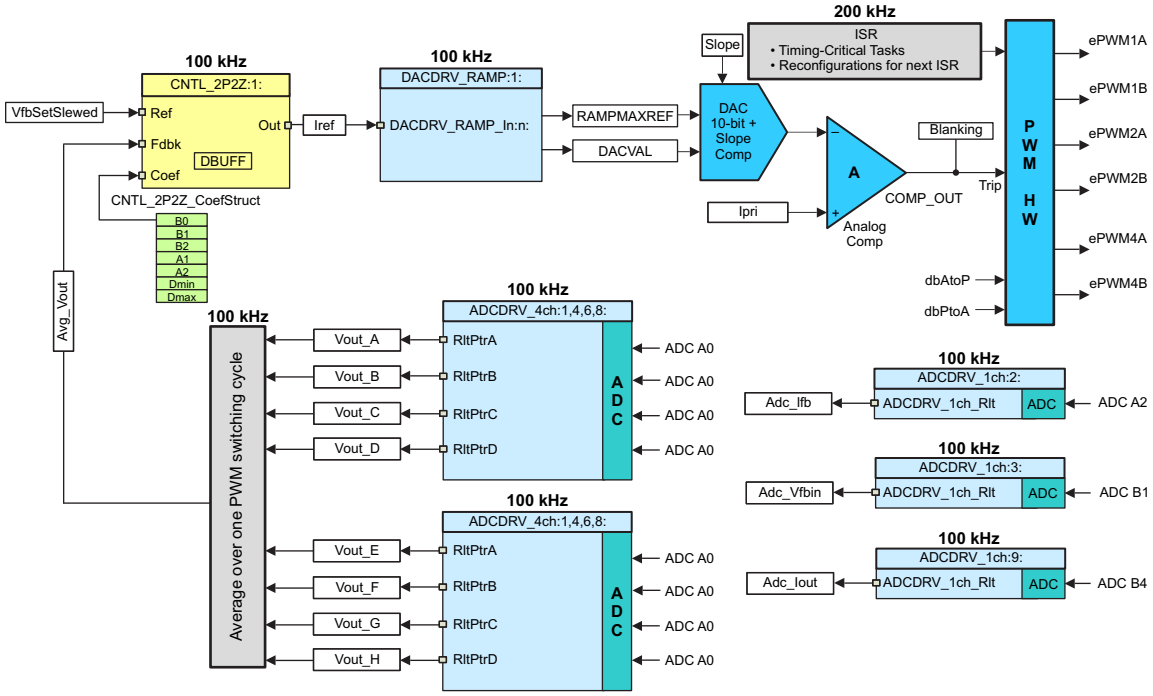


图 **18. PCMC** 控制流程

此系统由两个反馈环路控制：一个外部电压环路（使用软件控制区块执行）和一个内部峰值电流环路（使用片载模拟比较器执行）DAC 和 PWM 硬件资源。图 18还给出了控制区块的执行速率。 例如，电压环路控制器的执行速率为 100kHz（与 PWM 频率一样）。 请注意，中断处理例程 (ISR) 执行速率为 PWM 频率的两倍。 下面解释了上面执行的控制。

在每半个 PWM 开关周期内，输出电压在四个点上采样，每个点相隔的时间相等。 这些输出电压样本的平均*Avg_Vout*在一个 PWM 周期内计算。 *Avg_Vout*与电压控制器内的电压基准命令 (*Vref*) 的已转换版本 (*VfbSetSlewed*) 相比较。 然后，此电压控制器输出被转换为一个适当的 DAC 指令。 这是峰值电流基准指令，此指令最终控制全桥的两个桥臂之间相位重叠的数量来调节输出电压。 针对斜率补偿机制的*斜率*值可从较慢系统级任务内的控制流程外部进行设定。 提供了一个至少为 0.04V/µs 的缺省斜率。 要获得更多详细信息，请参见 [1] [2]和 [5]内的计算结果。 此片载模拟比较器将变压器初级电流与斜率补偿峰值电流基准相比较。 此比较器输出直接驱动器件上的 PWM 硬件。 某些时间关键配置代码在 ISR 内部执行，此 ISR 在一个 PWM 周期内被触发两次。 *dbAtoP*和*dbPtoA*参数分别为全桥的‘有源到无源’和‘无源到有源’桥臂提供死区时间值。 这些值被用来在负载范围内实现 ZVS 和 LVS。

3.2 *递增构建*

这个项目被分成两个递增构建。 这使得学习和熟悉此电路板和软件变得更加容易。 这个方法对也有利于调试和测试电路板。

构建选项被显示在以下部分中。 要选择一个特定的构建选项：

1. 将*HVPSFB-Settings.h*文件中的宏 INCR_BUILD 设定为如表 2中显示的相应构建选项。

2. 一旦选择了构建选项，则通过选择 → rebuild-all compiler option（重建所有编译器选项）来编译整个项目。

下一部分提供了运行每个构建选项的更多细节。

表 2. 针对 PCMC 的递增构建选项

递增构建选项	
INCR_BUILD = 1	峰值电流环路与恒定 I 指令和开电压环路相符合（检查 PWM 驱动电路和感测电路）
INCR_BUILD = 2	闭合电流和电压环路（满 PSFB）

4 运行递增构建的过程 - PCMC

提出 PSFB 系统的主源文件，ISR 汇编文件和针对 C 语言框架的项目文件位于以下目录（请使用软件包的最新版本）。 版本 1.1 是截止 2012 年 3 月的最新软件版本，它位于 www.ti.com/controlsuite ...\controlSUITE\development_kits\HVPSFB_v1.1\HVPSFB_PCMC 内。 包含这个软件的项目针对 Code Composer Studio™ v4。

WARNING

在电路板上会有高压出现。 它应该只在实验室环境中由经验丰富的电源专业人员处理。 为了安全评估这个电路板，应该使用一个适当隔离的高压 **DC** 电源。 在 **DC** 电源被施加到电路板上之前，必须将一个电压计和一个适当的阻性或电力负载连接至输出。 当此器件加电时，一定不要触摸此器件。

请按照以下步骤来构建和运行包括在*HVPSFB_PCMC*软件内的示例。

4.1 构建 1：峰值电流环路与开电压环路相符

4.1.1 目的

这个构建的目的是为了评估系统的峰值电流模式运行、验证 DAC 和 ADC 驱动器模块、验证电路板上的 MOSFET 驱动器电路和感测电路并逐渐熟悉 Code Composer Studio 的操作。 由于这个系统正在开环路运行，ADC 测量值只用于这个构建内的仪器仪表用途。 研究了构建并运行一个项目所需的步骤。

4.1.2 概述

构建 1 中的软件已经被配置，这样您就能够通过查看示波器上的不同波形并且通过从 Code Composer Studio 中交互地调整峰值电流基准指令来观察这条指令对输出电压的变化所带来的影响，来快速地评估 DAC 驱动器模块。 此外，您可以通过在观察视图中查看 ADC 采样数据来评估 ADC 驱动器模块。

如之前部分中所提到的那样，DAC 和 ADC 驱动器宏例示在 _DPL_ISR 内部执行。图 19显示了这个构建中使用的区块。 峰值电流基准指令被写入*DACDRV_RAMP*模块。 这个模块源自一个适当的 16 位值斜升最大值基准寄存器 (RAMPMAXREF)，这是用于斜率补偿的斜升起始值。 这个模块还将一个适当的 10 位值驱动至 DAC 值寄存器 (DACVAL)，如果无需斜率补偿或者斜率补偿由外部提供，那么可使用这个值。

片载模拟比较器将变压器初级电流与斜率补偿峰值电流基准相比较。 比较器输出被连接至 PWM 模块的触发区逻辑。ePWM1 模块运行为用于系统的主控时基。 它运行在上-下计数模式，而其它 PWM 模块运行在上计数模式。ePWM1A 和 ePWM1B 驱动 Q1 和 Q4 全桥开关，而 ePWM2A 和 ePWM2B 驱动 Q2 和 Q3 全桥开关。 ePWM4A 和 ePWM4B 驱动 Q5 和 Q6 同步整流器开关。 只要比较器输出在一个 PWM 半周期内变为高电平，瞬时为高电平的 ePWM2 模块输出（ePWM2A 或 ePWM2B）被立即下拉为低电平，而其它 PWM2 模块输出在一个适当的死区时间窗口后被拉至高电平(*dbAtoP*)。ePWM4A 和 ePWM4B 输出驱动方式类似。 图 12中显示了这些波形。

应该注意的是，这个斜率补偿斜升生成、比较器操作和 PWM 波形生成全都由硬件生成，而无需图 19中深蓝区块所表示的软件参与。 在为下半个 PWM 周期做准备时，某些寄存器配置在 ISR 内部完成。 这个时间关键代码在 ISR 的开始执行，不应被重新排序或改变。 汇编 ISR_DPL_ISR 例程由 ePWM1 触发。 请注意，ISR 触发频率是 PWM 开关频率的两倍。

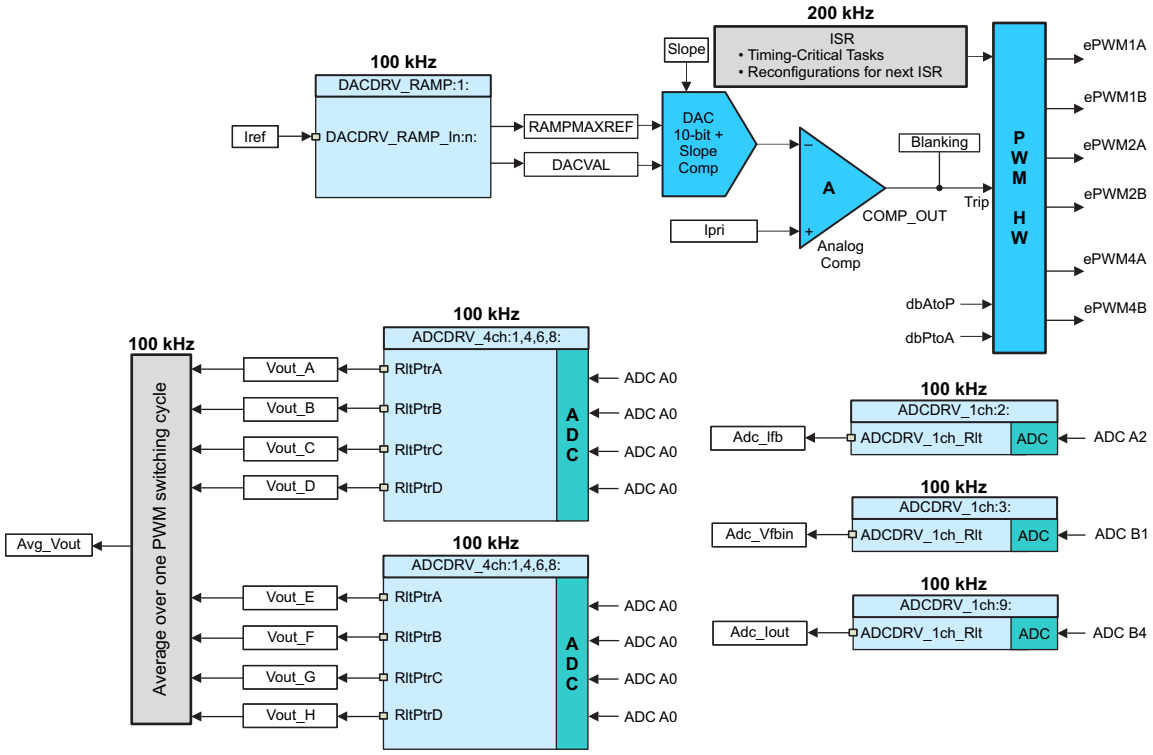


图 **19.** 构建 **1** 软件区块

*RAMPMAXREF*和*DACVAL*值取自输入*Iref*（Q24 变量）指令。表 3给出了*RAMPMAXREF*和*DACVAL*值，这些值从*Iref*计算得出'。

表 **3.** 示例 **RAMPMAXREF** 和 **DACVAL** 值

Iref (Q24)	**RAMPMAXREF = (Iref/2^8)**	**DACVAL = (Iref /2^14)**	**峰值电流基准（最大值 = 6.776A）**
2097152d	8192	128	0.845A
8388608d	32768	512	3.39A
16776704d	65534	1023	6.77A

ADC 驱动器模块被用来读取 12 位 ADC 结果并将它们转换为 Q24 值。 在每半个 PWM 周期 PWM1 转换 A 启动 (SOCA)，PWM3 SOCA 和 PWM3 转换 B 启动 (SOCB) 被用来触发四个针对输出电压的 ADC 转换。 这些转换触发点之间的相隔时间相等。 在每个 ISR 中读取这四个结果，因此在每个 PWM 周期内总共提供 8 个输出电压的 ADC 转换结果。 ISR 中的几行代码被用来根据这 8 个结果在一个 PWM 周期内计算平均输出电压。

输出电压感测电路由简单分压器组成。 一个匝数比为 1:100 的电流变压器，和一个感测电阻器 (48.7Ω) 被用来感测全桥变压器初级电流。 输入电压感测由辅助电源模块 (J7) 提供。 要获得与这些计算结果相关的更多细节，请见www.ti.com/controlsuite内的 HVPSFB-Calculations.xls 文件。

4.1.3 保护

在这个阶段，是时候介绍这个电路板上使用的关断机制了。 在这里，使用片载模拟比较器 2 来执行针对变压器初级电流的过流保护。使用内部 10 位 DAC 来设定基准触发电平并将其馈入这个比较器的反相端子。 这些比较器输出被配置为，只要感测到的电流大于设定的限值，就在 ePWM1 上生成一个单次触发操作。 这个 C2000 器件上触发机制的灵活性为在不同的触发事件上采取不同的操作提供了可能性。 在这个项目中，为了保护功率级，ePWM1A 和 ePWM1B 输出被立即驱动为低电平。然后 ePWM2A，ePWM2B，ePWM4A 和 ePWM4B 输出由软件强制为低电平。 在执行器件复位前，所有输出保持这个状态。 要获得与这些计算结果相关的更多细节，请参见www.ti.com/controlsuite内的 HVPSFB-Calculations.xls 文件。

当运行在恒定电流或恒定功率模式下时，软件还执行一个输出欠压关断机制。

软件中还执行输入欠压和过压闭锁。

4.1.4 资源映射

在表 4中总结了 C2000 MCU 和 HVPSFB 级之间的关键信号连接。 请注意，VMC 和 PCMC 项目的 PWM 映射是不同的。 控制器卡上的跳线（J2，J3 - PCMC 和 VMC PWM 驱动跳线使能）需要针对 VMC 模式正确配置（缺省跳线位置被设定用于 PCMC 运行）。 下面是针对这两个模式的跳线配置：

- PCMC：J2(1) → J3(1)，J2(2) → J3(2)，J2(3) → J3(3)，J2(4) → J3(4)，J2(7) → J3(7)，J2(8) → J3(8)
- VMC：J2(1) → J3(3)，J2(2) → J3(4)，J2(3) → J3(1)，J2(4) → J3(2)，J2(7) → J3(8)，J2(8) → J3(7)

表 **4. HVPSFB** 信号接口基准 **- PCMC**

信号名	说明	连接至 **C2000** 控制器
ePWM-1A	针对全桥开关 Q1 的 PWM 驱动	GPIO-00
ePWM-1B	针对全桥开关 Q4 的 PWM 驱动	GPIO-01
ePWM-2A	针对全桥开关 Q2 的 PWM 驱动	GPIO-02
ePWM-2B	针对全桥开关 Q3 的 PWM 驱动	GPIO-03
ePWM-4A	针对同步整流器开关 Q5 的 PWM 驱动	GPIO-06
ePWM-4B	针对同步整流器开关 Q6 的 PWM 驱动	GPIO-07

表 4. HVPSFB 信号接口基准 - PCMC (continued)

信号名	说明	连接至 C2000 控制器
Vout	PSFB 输出电压	ADC-A0
Ifb	变压器初级电流	ADC-A2 和 COMP1A
Ifb	变压器初级电流	ADC-A4 和 COMP2A (1)
Vfbin	PSFB 输入电压	ADC-B1
Iout1	PSFB 输出电流	ADC-B3
Iout2	PSFB 输出电流（被严重过滤）	ADC-B4

(1) 控制器卡上的缺省跳线 J4 配置。 输入可由跳线 J4 选择配置。

注: HVPSFB_PCMC 和 HVPSFB_VMC 项目使用位于它们自己项目目录内的*DSP2802x_Comp.h*和*DSP2802x_EPWM.h*头文件，而不是那些位于*device_support*目录内的文件。 这两个*device_support*目录内的文件将在*ControlSuite*的晚些时间更新，到那时这些更新过的文件可用于这个两个项目。

4.2 过程

4.2.1 启动 Code Composer Studio 并打开一个项目

使用以下步骤来快速执行这个构建：

1. 请确保插装了电路板上的跳线 J8 和 J6。
2. 缺省情况下，控制器卡的 Piccolo 宏上的电阻器 R6 和跳线 J1 被移除以启用闪存引导。 重新插装电阻器 R6 和跳线 J1 来运行并编辑 RAM 或编辑闪存。
3. 为了实现仿真，将 USB 连接器接至 Piccolo 控制器卡。 建议使用一个输出被设定为大约 400V DC 的合适的隔离式 DC 电源。 当它被连接至主电路板的 J1 和 J2 上时，DC 电源应该保持关闭状态。
4. 使用一条 20AWG 600V 电线将电源接至 J1 和 J2。 请确保这个连接的极性是正确的。 在 J3 和 J4 的 DC 输出上施加一个适当的阻性 或 DC 电力负载至移相全桥系统。
5. 此时不要接通 400V DC 电源。
6. 将 TP1 和 TP2 之间的偏置电源加电至大约 11V DC（这个电压必须小于 12V）。
7. 双击桌面上的 Code Composer Studio 图标。
8. 将 Code Composer Studio 扩展到您的屏幕大小。
9. 如果欢迎屏幕打开的话，请关闭它。 此项目包含开发一个可执行输出文件 (.out) 所需的全部文件和构建选项，此可执行输出文件能够在 MCU 硬件上运行。
10. 单击菜单栏上的 Select Project → Import Existing Code Composer Studio and CCE Eclipse Project（选择项目 → 导入现有的 Code Composer Studio 和 CCE Eclipse 项目）。 在 Select root directory（选择根目录）下 → 导航至并选择*..\controlSUITE\development_kits\HVPSFB_v1.1\HVPSFB_PCMC*目录。 请确保在 Projects 标签页内选中 HVPSFB_PCMC。
11. 单击完成。 这个项目调用所有需要的工具（编译器、汇编程序、连接程序）来建立项目。
12. 单击 Project 左侧的加号符号 (+)（在左侧的项目窗口中）。 您的项目窗口与以下的图形类似。

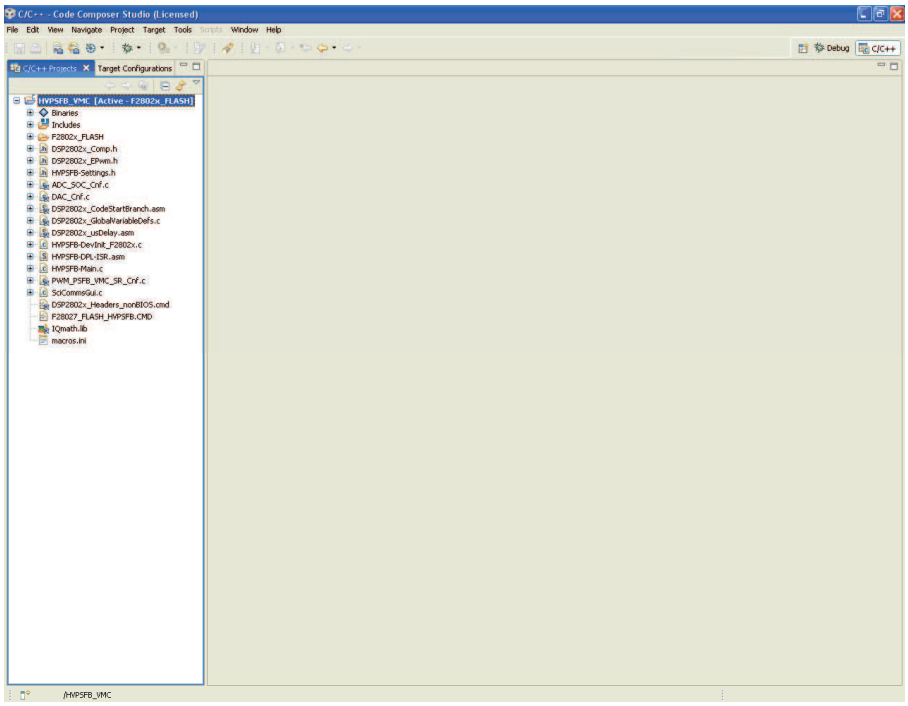


图 **20. C** 和 **C++** 项目

4.2.2 器件初始化、主、和 **ISR** 文件

注: *不要*对源文件做出任何改动 - *检查即可*。

1. 通过双击项目窗口中的文件名可打开并检查*HVPSFB-DevInit_F2802x.c*。 请注意，系统时钟、外设时钟预分频和外设时钟使能已经被设置。 请注意，共用通用输入输出 (GPIO) 引脚已经被配置。
2. 打开并检查*HVPSFB-Main.c*。 请注意对*DeviceInit()*函数的调用和其它变量初始化。 此外，请注意针对不同递增构建选项（特别是您现在打算编译的构建），ISR 初始化以及 (;;) 环路所用背景的代码。
3. 确定并检查以下专门针对构建 1 的初始化代码下，主文件内的代码。这就是控制流程内 DACDRV_RAMP 区块被连接和初始化的位置。

```
DacDrvCnf (1,  1280,  1, 2, Slope);     // Comp1, DACval = 1280 (Initial), Slope compensation
is used,
                                        // Ramp is PWM3 Synced, Initial Slope
                                        //  DAC connections
DACDRV_RAMP_In1 = &Iref;                // Controls the DAC reference voltage
```

4. 确定并检查以下专门针对构建 1 的初始化代码下，主文件内的代码。这就是控制流程内 ADCDRV_4CH 和 ADCDRV_1CH 区块的多个例示被配置、初始化以及被连接的位置。

```
#define     Vfb_outR        AdcResult.ADC         RESULT1     //
#define     IfbR            AdcResult.ADC         RESULT2     //
#define     Vfb_inR         AdcResult.ADC         RESULT3     //

#define     IoutR           AdcResult.ADC         RESULT9     //

// Channel Selection for Cascaded Sequencer
    ChSel [0] = 0;          // A0 - O/P Voltage - Dummy
    ChSel [1] = 0;  //B     // A0 - O/P Voltage
    ChSel [2] = 2;          // A2 - Transformer Primary Current
    ChSel [3] = 9;          // B1 - I/P Voltage
    ChSel [4] = 0;  //C     // A0 - O/P Voltage

    ChSel [5] = 0;          // A0 - O/P Voltage - Dummy
```

```
    ChSel [6] = 0;  //A   // A0 - O/P Voltage
    ChSel [7] = 0;        // A0 - O/P Voltage - Dummy
    ChSel [8] = 0;  //D   // A0 - O/P Voltage
    ChSel [9] = 11;       // B3 - Iout1
    ChSel [9] = 12;       // B4 - Iout2

    TrigSel[0] = ADCTRIG_EPWM3_SOCA;        // O/P Voltage sampling triggered by EPWM3 SOCA -
 Dummy
    TrigSel[1] = ADCTRIG_EPWM3_SOCA; //B    // O/P Voltage sampling triggered by EPWM3 SOCA
    TrigSel[2] = ADCTRIG_EPWM3_SOCA;        // Transformer Primary Current sampling triggered
by EPWM3 SOCA
    TrigSel[3] = ADCTRIG_EPWM3_SOCA;        // I/P Voltage sampling triggered by EPWM3 SOCA
    TrigSel[4] = ADCTRIG_EPWM3_SOCA; //C    // O/P Voltage sampling triggered by EPWM3 SOCA
    TrigSel[5] = ADCTRIG_EPWM1_SOCA;        // O/P Voltage sampling triggered by EPWM1 SOCA at
CTR = ZRO or PRD - Dummy
    TrigSel[6] = ADCTRIG_EPWM1_SOCA; //A    // O/P Voltage sampling triggered by EPWM1 SOCA at
CTR = ZRO or PRD
    TrigSel[7] = ADCTRIG_EPWM3_SOCB;        // O/P Voltage sampling triggered by EPWM3 SOCB at
CMPB3 - Dummy
    TrigSel[8] = ADCTRIG_EPWM3_SOCB; //D    // O/P Voltage sampling triggered by EPWM3 SOCB at
CMPB3
    TrigSel[9] = ADCTRIG_EPWM2_SOCA;        // Iout triggered by EPWM2 SOCA
    EALLOW;
    AdcRegs.SOCPRICTL.bit.SOCPRIORITY = 9;  // SOC0-8 are high priority
    EDIS;
    ADC_SOC_CNF(ChSel,TrigSel,ACQPS, 16, 0);// ACQPS=8, No ADC channel triggers an interrupt
IntChSel > 15,
                                            // Mode= Start/Stop (0)
                                            // ADC feedback connections
    ADCDRV_4ch_RltPtrA = &Adc_VavgBus[1];
    ADCDRV_4ch_RltPtrB = &Adc_VavgBus[2];
    ADCDRV_4ch_RltPtrC = &Adc_VavgBus[3];
    ADCDRV_4ch_RltPtrD = &Adc_VavgBus[4];
    ADCDRV_1ch_Rlt2 = &Adc_Ifb;
    ADCDRV_1ch_Rlt3 = &Adc_Vfbin;
    ADCDRV_1ch_Rlt9 = &Adc_Iout;
```

5. 打开并检查 HVPSFB-DPL-ISR.asm。 请注意 _DPL_Init 和 _DPL_ISR 部分。 这分别是 DAC 和 ADC 驱动器宏例示分别完成初始化和运行结束的地方。 您可以选择关闭被检查的文件。

4.2.3 建立和加载项目

1. 在位于www.ti.com/controlsuite内的*HVPSFB-Settings.h*文件内将递增构建选项选为 1。

注: 只要您改变了*HVPSFB-Settings.h*内的递增构建选项，那么请始终“重建全部”。

2. 单击 Project → “Rebuild All”按钮并查看构建窗口中运行的工具。

3. 单击 Target → ”Debug Active Project”（目标 → “调试激活项目”）。 如果还未选择，Code Composer Studio 将要求您打开一个新目标配置文件。 如果已经为这个连接创建了一个有效目标配置文件，您可前往步骤 5。 在新目标配置窗口内，为您将使用的目标输入以 .ccxml 为后缀的文件名（例如：xds100-F28027.ccxml）。 选中 "Use shared location"（使用共用位置） 并单击 Finish。
4. 在打开的 .ccxml 文件中，将 Connection（连接）选为 "Texas Instruments XDS100v2 USB Emulator"（德州仪器 (TI) XDS100v2 USB 仿真器），向下滚动并选择 "TMS320F28027" 器件。
5. 单击 Save（保存）。
6. 单击 Target → ”Debug Active Project”。 根据所选择的项目配置，此程序将被载入闪存或 RAM 存储器。 这个项目只随 F2802x_FLASH 配置提供。 您现在应该处于 Main() 的起始位置。

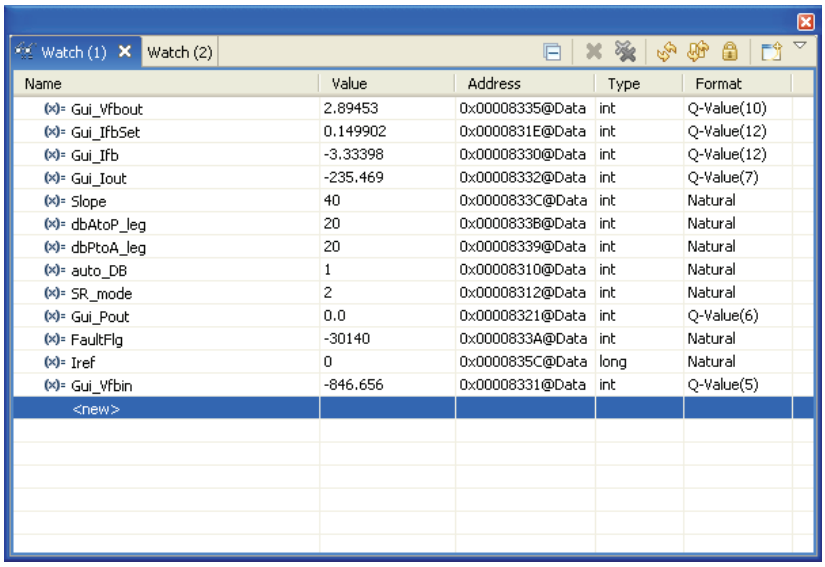
4.2.4 调试环境窗口

在调试代码的同时观察本地和全局变量是标准的调试做法。 在 Code Composer Studio 内有多种不同的方法来进行这个操作，诸如存储器视图和观察视图。 此外，Code Composer Studio 能够生成时（和频）域图。这使您能够使用图形窗口来观察波形。

如果调试环境启动时观察视图没有打开，那么打开一个新的观察视图并通过下面给出的步骤为其添加不同参数。

1. 单击菜单栏上的 View → Watch（视图 → 观察）。
2. 单击 “Watch (1)" 标签页。 您可将任一变量添加至观察视图。
3. 在 "Name" 栏内的空白方框内输入您想观察的变量符号名并且按下键盘上的回车键。 请确保按要求修改了 "Format"（格式）。 观察视图应该与下面所显示的类似。 请注意，主代码中的某些变量此时还未被初始化，有可能包含一些垃圾值。

FaultFlg，如果被置位的话，表示一个过流情况（上面已经讨论过了），此情况关断 PWM 输出。 PWM 输出被保持在这个状态，直到器件复位（请按照步骤 11内的适当流程进行操作）。 *Ipri_trip*变量针对片载比较器 2 设定内部 10 位 DAC 基准电平。请注意，这是一个 Q15 数。



4.2.5 使用实时仿真

实时仿真功能是一种特殊的仿真特性，可以让 Code Composer Studio 中的窗口在 MCU 运行时以 10Hz 的速率刷新。 这不但可实现图形和观察视图更新，而且使您能够改变观察和存储器窗口内的值，并使这些值的变化影响 MCU 运行方式。 例如，当调整控制律参数时，这一功能十分有用。

1. 通过将您的鼠标悬停在水平工具条的按钮上并单击 *Enable Silicon Real-Time Mode*（*启用芯片实时*

模式）（当暂停时处理关键中断，可在运行时访问调试程序）按钮来启用实时模式。

2. 如果出现一个消息框，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。
3. 现在单击同一水平工具条上的  *Enable Polite Real-Time Mode（启用礼貌实时模式）*。
4. 当打开大量窗口时，由于仿真连接上的带宽是有限的，以持续的刷新率更新太多窗口和变量会导致刷新频率停顿。 右键单击观察视图内的  按钮并选择“*Customize Continuous Refresh Interval.（定制持续刷新间隔。）*”。 您可以通过改变*Continuous refresh interval (milliseconds)（持续刷新间隔（毫秒））*值来使针对观察视图的刷新率降下来。 通常 4000ms 的刷新率可满足这些运用的要求。
5. 单击观察视图中的  Continuous Refresh（持续刷新）按钮。

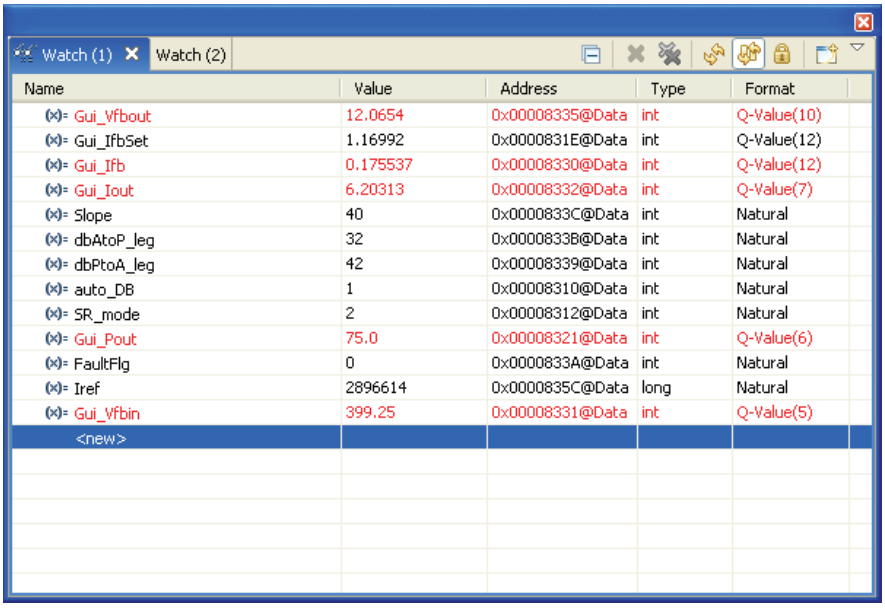
4.2.6 运行代码：

1. 使用 <F8> 键来运行代码，或者使用工具条上的 RUN（运行）按钮，或者使用菜单栏上的 Target → Run（目标 → 运行）。
2. 在观察视图中，变量*Gui_IfbSet*应该被设定为 0.15 (Q12)。 这个变量以安培为单位显示峰值电流基准指令并且将*Iref*驱动至*DACDRV_RAMP*模块。 *Gui_IfbSet*的值不应小于 0.15。
3. 在 DC 输出上，将一个适当的阻性负载施加到 PSFB 系统。 12V 输出时，以汲取大约 3A-6A 电流的负载作为开始比较好。

注: 出于安全考虑，建议使用一个隔离式 DC 电源来为电路板提供 400V DC 输入。

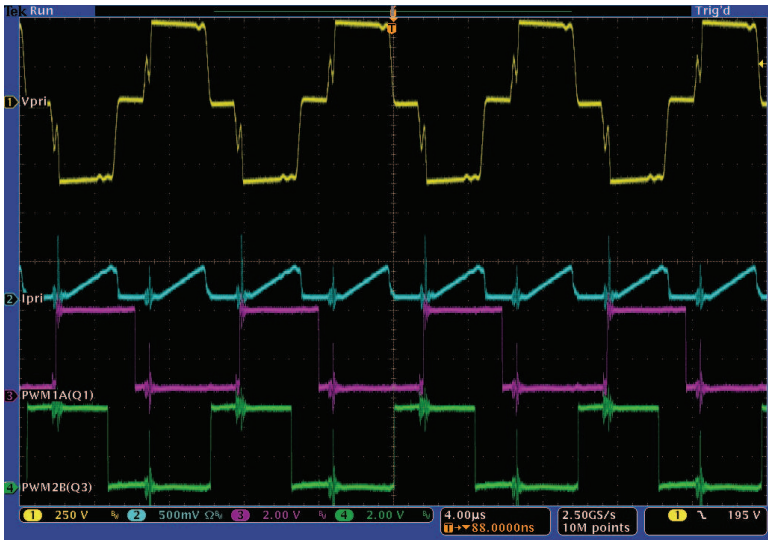
---

4. 在 J1，J2 上为输入提供 400V DC 电源。
5. 通过在观察窗口中将*Gui_IfbSet*设定为一个更高的值 0.25（比如说）的值可增加峰值电流基准命令。 输出电压应该增加。 仔细观察输出电压，不应该让这个值超过电路板的承受能力。 请牢记，当您使用一个特定的*Gui_IfbSet*值进行操作时，如果负载突然减少，输出电压将上升。 因此，当在构建 1 中运行时，不要突然改变负载或大幅增加*Gui_IfbSet*指令。
6. 对于不同的*Gui_IfbSet*值，请在观察视图中观察不同的 ADC 结果。 这个观察视图是与一个*Gui_IfbSet*指令为 1.17A 的系统相对应的观察视图，此系统的输入电压大约为 400V，输出为 12V 时的负载大约为 6A。



Name	Value	Address	Type	Format
Gui_Vfbout	12.0654	0x00008335@Data	int	Q-Value(10)
Gui_IfbSet	1.16992	0x0000831E@Data	int	Q-Value(12)
Gui_Ifb	0.175537	0x00008330@Data	int	Q-Value(12)
Gui_Iout	6.20313	0x00008332@Data	int	Q-Value(7)
Slope	40	0x0000833C@Data	int	Natural
dbAtoP_leg	32	0x0000833B@Data	int	Natural
dbPtoA_leg	42	0x00008339@Data	int	Natural
auto_DB	1	0x00008310@Data	int	Natural
SR_mode	2	0x00008312@Data	int	Natural
Gui_Pout	75.0	0x00008321@Data	int	Q-Value(6)
FaultFlg	0	0x0000833A@Data	int	Natural
Iref	2896614	0x0000835C@Data	long	Natural
Gui_Vfbin	399.25	0x00008331@Data	int	Q-Value(5)
<new>				

7. 以下的示波器捕捉显示了变压器初级电压、初级感测电流以及驱动两个对角相对开关（Q1 和 Q3）的PWM 波形（在上面已描述的条件下可见）。



8. 缺省情况下，同步整流器运行在模式 2 下。您可以通过在观察视图中将*SR_mode*变量改为 0，1 或 2 来改变它们的运行模式。 请观察被汲取的输入电流的变化以及使用不同 SR 模式时，输出电压变化。 您还可以探测驱动同步整流器开关的 PWM 波形。 当运行在极低负载或者输出电压极低（少于 6V）时，不要在不同的 SR 模式间进行改变。 在这些情况下，使用缺省 SR 模式 2。
9. 尝试不同的*Gui_IfbSet*值并观察相应的 ADC 结果。 以小步长增加*Gui_IfbSet*。 始终小心观察输出电压，这个值不应超过电路板的处理能力。 诸如 PWM 栅极驱动信号、输入电压和电流以及输出电压等的不同波形也可使用一个示波器进行探测。 在探测这些用于隔离式 DC-DC 转换器的高压和高电流时应该采取适当的安全防护措施并考虑合适的接地要求。

10. 实时模式时完全停止是一个两步过程。 当 400V DC 输入关闭时，等待几秒钟。 首先，通过使用工具条上的 Halt（暂停）按钮，或者使用 Target → Halt（目标 → 暂停）来暂停处理器。 然后再次单击 
此按钮来使 MCU 离开实时模式，然后复位 MCU。
11. 您可选择使 Code Composer Studio 继续运行，用于下一个应用，或者选择将其关闭。

4.3 构建 2：整个 HVPSFB

4.3.1 目的

这个构建的目的在于验证整个基于 PCMC 的 HVPSFB 项目在 Code Composer Studio 环境中的运行。

4.3.2 概述

图 21显示了这个构建中所使用的软件区块。 一个 2 极 2 零控制器用于电压环路。 根据应用的控制环路需求的不同，有可能使用诸如一个 PI，一个 3 极 3 零等其它控制器区块。 如图 21中所示，电压环路区块的执行频率为 100Hz CNTL2P2Z 是一个由无限脉冲响应 (IIR) 滤波器结构实现的二阶比较器。 这个函数与任何外设无关，因此就不需要 CNF 函数调用。

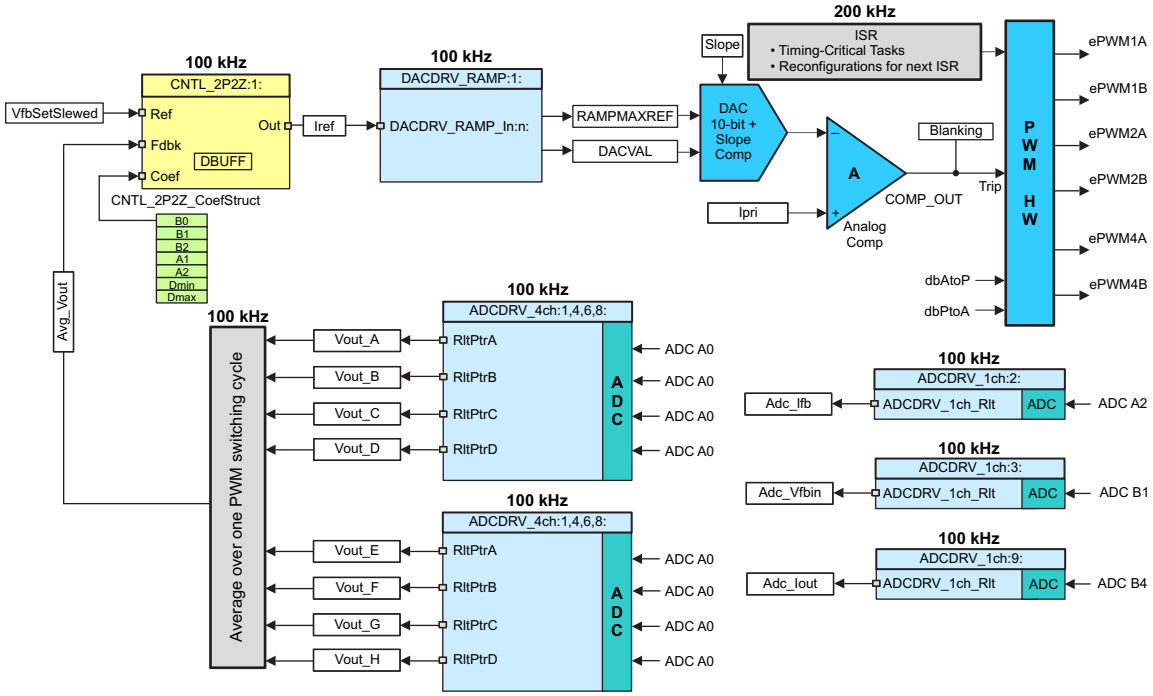


图 21. 构建 2 软件区块

将被修改的 5 个系数被存储为结构 CNTL_2P2Z_CoefStruct1 的元素组件，此结构的其它元素被用来钳制控制器输出。 如果系统需要多个环路，CNTL_2P2Z 区块可被实例化多次。 每个实例可具有单独的系数集合。 通过反复试验直接独立操纵这 5 个系数几乎是不可能的，并且要求数学分析或诸如 matlab，mathcad 等工具的帮助。 这些工具提供伯德图、根轨迹法和其它特性来确定相位裕量度、增益裕度等。

为了简化环路调整且无需复杂数学或分析工具，通过将 P，I 和 D 的更加直观的系数增益简便映射为 B0，B1，B2，A1 和 A2，系数选择问题已经从自由度 5 减为自由度 3。 这样可实现对 P，I 和 D 的独立且逐步的调节。 下面给出了这些映射等式。

补偿器区块 (CNTL_2P2Z) 有两个极和两个零，并基于普通 IIR 滤波器结构。 它有一个基准输入和一个反馈输入。 对于电路环路，此反馈是在一个 PWM 周期内计算得出的平均输出电压 (*Avg_Vout*)，而到控制器的基准输入是输出电压基准指令 (*Vref*) 的已转换版本 (*VfbSetSlewed*)。 传输函数给出如下：

$$\frac{U(z)}{E(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

PID 控制器的递归形式由不同的等式给出：

$$u(k) = u(k-1) + b_{0e}(k) + b_{1e}(k-1) + b_{2e}(k-2)$$

其中：

$$b_0 = K_p' + K_i' + K_d'$$
$$b_1 = K_p' + K_i' - 2K_d'$$
$$b_2 = K_d'$$

这个的 z 域传输函数形式为：

$$\frac{U(z)}{E(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 - z^{-1}} = \frac{b_0 z^2 + b_1 z + b_2}{z^2 - z}$$

将它与普通形式相比较，您可发现 PID 只是 CNTL_2P2Z 控制的一个特殊情况，其中：

$$a1 = -1 \text{ and } a2 = 0$$

对于电压环路，这些 P，I 和 D 系数为：Pgain，Igain 和 Dgain。 这些 P，I 和 D 系数所采用的格式为 Q26。 为了简化 GUI 环境（或者 Code Composer Studio 观察视图）的调整，这 3 个系数被进一步调整为 0 至 999 之间的值（Pgain_Gui，Igain_Gui 和 Dgain_Gui）。

在 GUI 环境中，可使用 2 极 (*fp1*，*fp2*)，2 零 (*fz1*，*fz2*) 和增益 (*Kdc*) 来调整电压环路。 这些参数提供格式为 I5Q10 的*b2_Gui*，*b1_Gui*，*b0_Gui*，*a2_Gui*和*a1_Gui*系数，然后这些系数被转换为用于 2P2Z 控制器的 5 个 Q26 系数。 虽然不建议这么做，但是*b2_Gui*，*b1_Gui*，*b0_Gui*，*a2_Gui*和*a1_Gui*值也可使用观察视图从 Code Composer Studio 环境直接转换。 要获得与这些计算结果相关的细节，请参见www.ti.com/controlsuite内的 HVPSFB-Calculations.xls 文件。 对于根据极，零和增益以及开关频率推导出系数值的等式也在 GUI 源文件中明确给出。

这个项目通过在执行期间提供系数间的轻松切换，实现对两个环路调整方法的简便评估。 可通过简单单击 GUI 上的*2P2Z(On)/PID(Off)*按钮或在 Code Composer Studio 的观察视图中将*pid2p2z_GUI*变量从 0 改为 1 来完成这一操作。 GUI 环境内基于 PID 的环路调整 (*pid2p2z_GUI=0*) 曾被用作一个起始点。 然后，与这些 PID 经调整系数相对应的极、零和增益被用作一个起始点，以根据第二个方法 (*pid2p2z_GUI=1*) 对环路进行进一步调整。 然后，通过在 GUI 环境内改变极、零和增益来获得更佳结果来调整为最优动态性能。 缺省情况下，使用基于这些经调整的极、零和增益值的系数（*pid2p2z_GUI= 1*- 缺省值）。

注： 当使用 2 极 2 零控制器系数调节来调整系统时，如果在 GUI 中选择极、零和 Kdc 值，那么这些系数 (b2，b1，b0，a2，a1) 中的任何一个的大小将大于或等于 32，那么这些系数值不由 GUI 发送给控制器。

4.4 恒定电流 (CC) 和恒定功率 (CP)

试验恒定电流和恒定功率功能性的简单执行已经包含在这个项目中。 缺省情况下，这个功能性是被禁用的。 通过改变电压环路控制器的钳位值来实现恒定电流功能。 通过根据负载调整电压环路基准指令以保持输出上的恒定功率来实现恒定功率功能。图 22提供了针对这些功能的完整软件流程图。

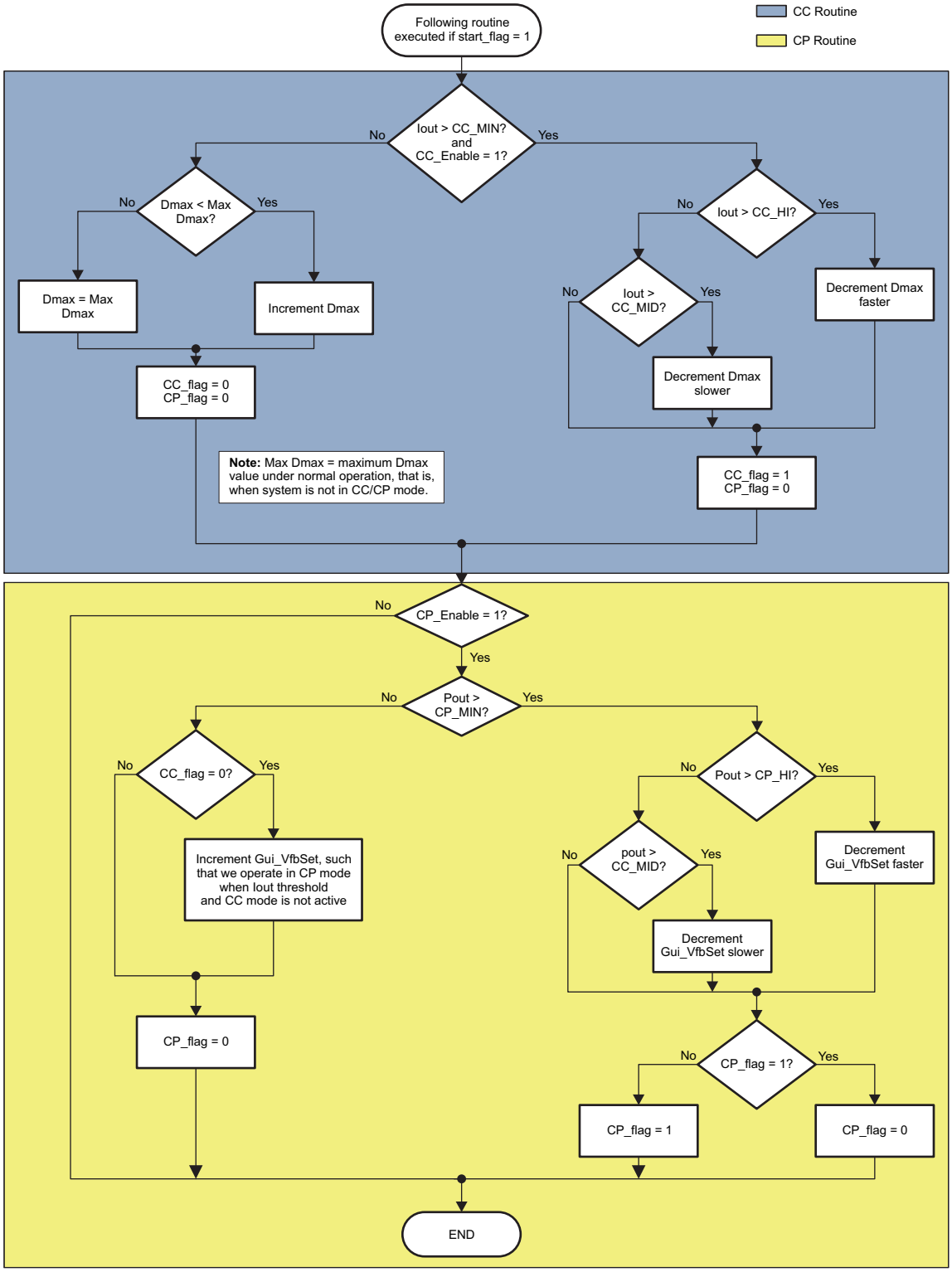


图 **22.** 恒定电流和恒定功率软件流程图

4.5 过程

构建和加载项目

使用以下步骤来快速执行这个使用预先配置的工作环境的构建：

1. 按照构建 1 中的步骤 1至 2。如果您在 Code Composer Studio 的最后一次使用中操作构建 1，同一个工作区应该与项目一同打开。
2. 如果情况不是这样，您可以通过单击 File → Switch Workspace（文件 → 切换工作区）来打开用于构建 1 的工作区，然后前往至正确工作区。 如果一个工作区未被保存或被删除，请按照与构建 1 中相同的步骤 3和 4。
3. 找到并检查主文件中专门用于构建 2 的初始化代码。 这就是控制流程中，所有控制区块被配置、初始化并被连接的地方。
4. 在*HVPSFB-Setting.h*中将递增构建选项选为 2。

注: 只要您改变了*HVPSFB-Setting.h*中的递增构建选项，请始终进行“重建全部”操作。

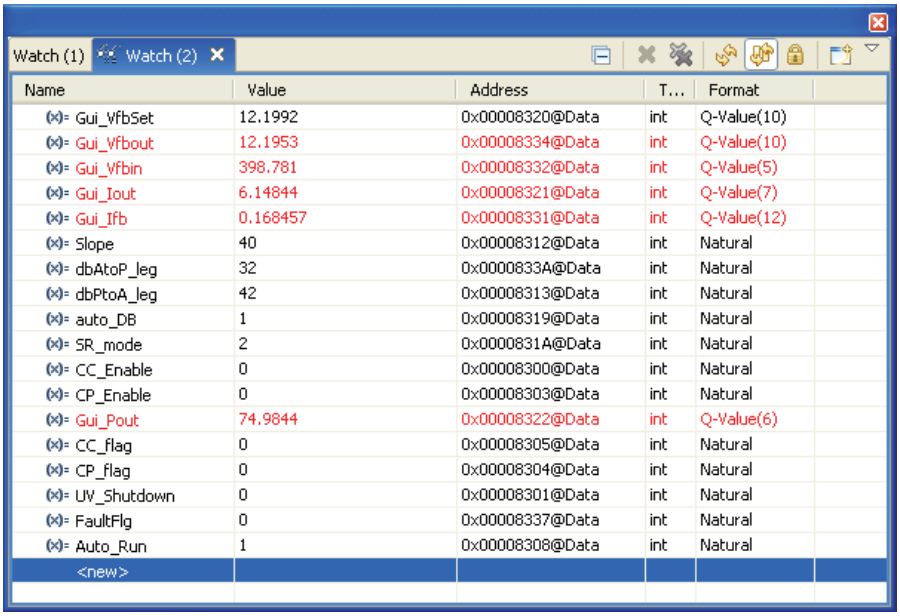
5. 单击 Project → "Rebuild All" 按钮并观察在构建窗口中运行的工具。
6. 单击 Target → "Debug Active Project"。 根据项目配置，此程序将被载入闪存或 RAM 存储器。 这个项目只随 F2802x_FLASH 配置一同提供。 现在您应该在 Main() 的开始位置。

4.5.1 调试环境窗口

在调试代码时观察本地和全局变量是标准调试做法。 在 Code Composer Studio 内有多种不同的方法来进行这个操作，诸如存储器视图和观察视图。 此外，Code Composer Studio 能够生成时（和频）域图。 这使您能够使用图形窗口来观察波形。

如果调试环境启动时观察视图没有打开，那么打开一个新的观察视图并通过下面给出的步骤为其添加不同参数。

1. 单击菜单栏上的 View → Watch（视图 → 观察）。
2. 单击 “Watch (1)" 标签页。
3. 如果已经从为构建 1 保存的调试环境中打开了一个观察视图，请单击*New Watch View（新观察视图）*  按钮。一个 "Watch (2)" 标签页将打开，然后您可以将其拖拽至您选择的窗口中进行检查。 您可将任一变量添加至这个观察视图标签。
4. 在 "Name" 栏内的空白方框内输入您想观察的变量符号名并且按下键盘上的回车键。 请确保按要求修改了 "Format"。 观察视图应该与下面所显示的类似。 请注意，主代码中的某些变量此时还未被初始化，有可能包含一些垃圾值。



请注意，在观察视图中有额外变量。

5. Gui_VfbSet 被用来设定输出电压指令。

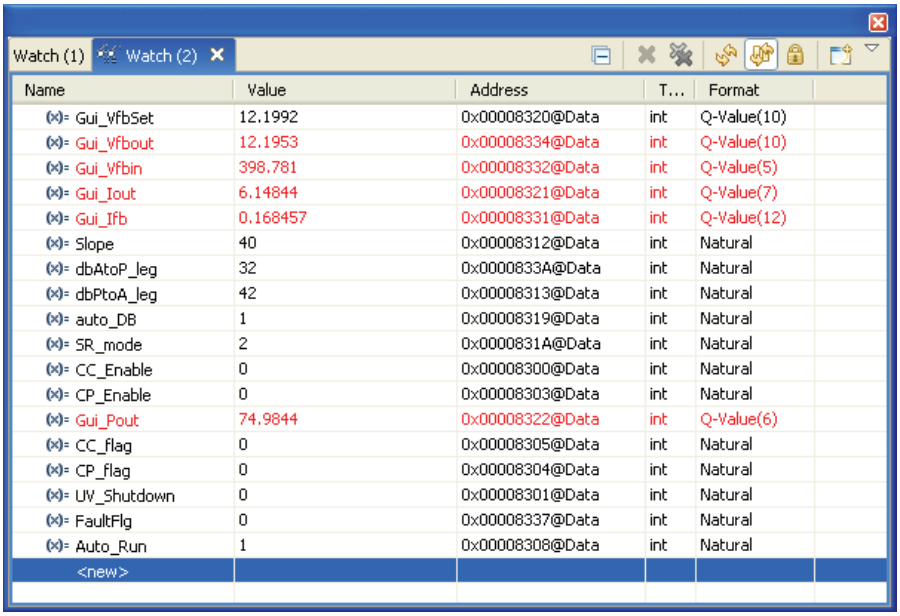
4.5.2 运行代码：

1. 按照构建 1 过程步骤 1至 2来启用实时模式和针对观察视图的持续刷新，也可在需要时改变观察视图的持续刷新间隔。
2. 使用 <F8> 键来运行代码，或者使用工具条上的 RUN（运行）按钮，或者使用菜单栏上的 Target → Run（目标 → 运行）。
3. 在 DC 输出上，将一个适当的阻性负载施加到 PSFB 系统。 12V 输出时，以汲取大约 3A-6A 电流的负载作为开始比较好。

> **注:** 出于安全考虑，建议使用一个隔离式 DC 电源来为电路板提供 400V DC 输入。

4. 在 J1，J2 上为输入提供 400V DC 电源。
5. 缺省情况下，*Auto_Run*被置位为 1。如果不为 1，请在观察视图中将其改为 1。 现在，输出电压应该开始斜升至 12V。可通过改变变量*VfbSlewRate*来更改此输出电压斜升率。

   下图显示了一个观察视图，此视图与输出上为 12.2V 的系统运行相对应，此时系统的输入电压为大约 400V，负载大约为 6A 输出。



6. 图 23显示了在上面已描述的情况下可见的变压器初级电压、初级感测电流以及驱动两个对角相对开关（Q1 和 Q3）的 PWM 波形。 还显示了这些情况下开关 Q3 的 ZVS 切换以及开关 Q4 的 LVS 切换。

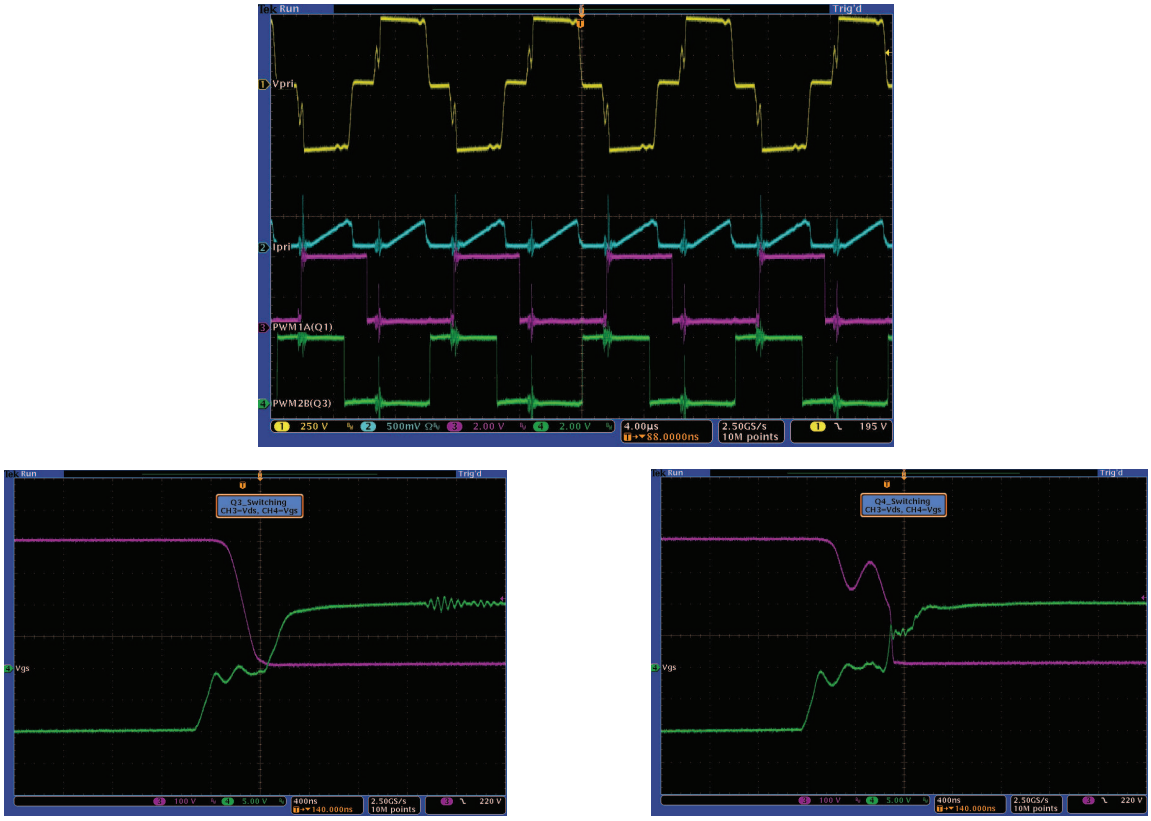


图 **23.** 变压器初级电压、初级感测电流以及驱动两个对角相对开关（**Q1** 和 **Q3**）的 **PWM** 波形

7. 缺省情况下，同步整流器运行在模式 2 下。您可以通过在观察视图中将 SR_mode 变量改为 0，1 或 2 来改变它们的运行模式。 请观察被汲取的输入电流的变化以及使用不同 SR 模式时，输出电压变化。 您还可以探测驱动同步整流器开关的 PWM 波形。 当运行在极低负载时（低于 3A），不要在不同的 SR 模式中进行改变。 在这些情况下，使用缺省 SR 模式 2。
8. 观察负载变化对输出电压和输入电流的影响。 实际上不应对输出电压产生影响。 相似地，观察改变输入

电压的影响。 实际上，仍然不应对输出电压产生影响。

注: 请确保做出的这些变化在本文档技术规格中所列出的电路板处理能力范围内。

9. 诸如 PWM 栅极驱动信号、输入电压和电流以及输出电压等的不同波形也可使用一个示波器进行探测。 在探测这些用于隔离式 DC-DC 转换器的高压和高电流时应该采取适当的安全防护措施并考虑合适的接地要求。
10. 也可在这个项目中执行简单‘恒定电流 (CC)’和‘恒定功率 (CP)’功能，并且可通过启用和禁用它们的相应标志（*CC_Enabe 和 CP_Enable*）来试验这些功能。
11. 当运行在实时模式时，完全停止 MCU 是一个两步过程。 等待几秒钟，400V DC 输入被关闭。 首先，通过使用工具条上的 Halt（暂停）按钮，或者使用 Target → Halt（目标 → 暂停）来暂停处理器。 然后再次单击此  按钮来使 MCU 离开实时模式，然后复位 MCU。
12. Close Code Composer Studio。

5 软件概述 - VMC

5.1 软件控制流程

HVPSFB_VMC 项目使用 “C-background/ASM-ISR” 架构。 它使用 C 语言代码作为针对应用的主要支持程序，并负责所有系统管理任务、决策制定、智能和主机交互。 汇编代码被严格限制为 ISR，此例程运行所有关键控制代码，这代码通常包括 ADC 读取、控制计算和 PWM 与 ADC 更新。

图 24描述了针对这个项目的一般软件流程。

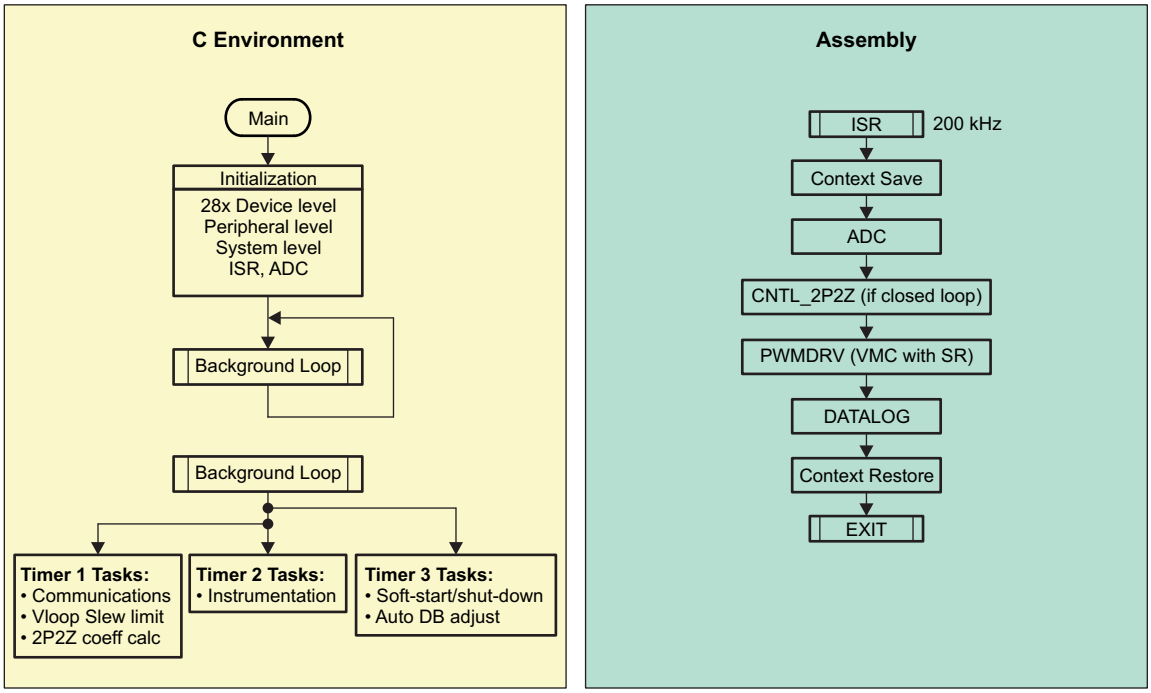


图 **24. VMC** 软件流程

本项目中使用的关键框架 C 语言文件为：

- HVPSFB-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”。
- HVPSFB-DevInit.c - 这个文件负责 F280x 器件的一次初始化和配置，并且包括诸如设置时钟，PLL，GPIO 等功能。

ISR 包括一个单个文件：

- HVPSFB-DPL-ISR.asm - 这个文件包含所有时间关键“控制类型”代码。 这个文件有一个初始化部分（一次执行）和一个运行时间部分，其执行速率（通常情况下）与用来触发它的 PWM 时基相同。

电源库函数（模块）从这个框架被调用。

库模块也许具有一个 C 语言和一个汇编组件。 在这个项目中，使用了以下库模块。 C 语言和相应的汇编模块名称为：

表 **5.** 库模块

C 配置函数	**ASM** 初始化宏	**ASM** 运行时间宏
PWMDRV_PSFB_VMC_SR_CNF	PWMDRV_PSFB_VMC_SR_INIT n,m,p	PWMDRV_PSFB_VMC_SR n,m,p
ADC_SOC_Cnf.c	ADCDRV_4CH_INIT m,n,p,q	ADCDRV_4CH m,n,p,q
	CNTL_2P2Z_INIT n	CNTL_2P2Z n

控制区块也以图形的方式显示在图 25中。

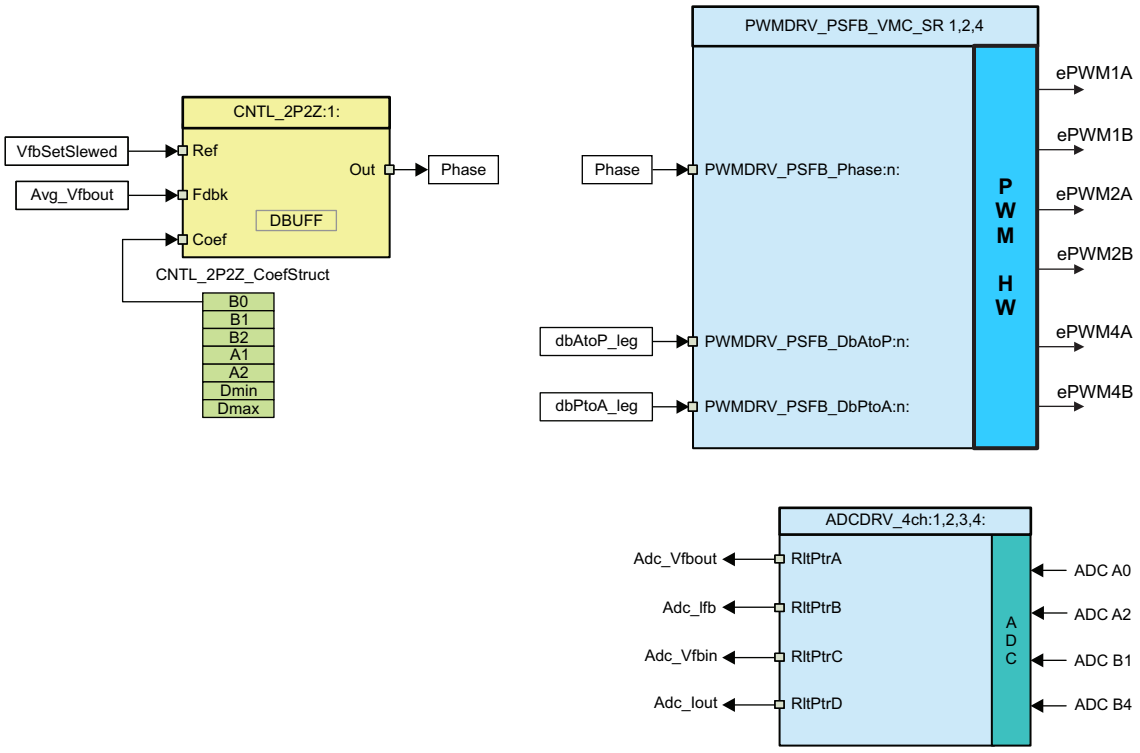


图 25. VMC 软件区块

请注意图 25中针对模块的颜色编码。 ‘深蓝色’的区块代表 C2000 微控制器控制器上的硬件模块。 ‘蓝色’区块是这些模块的软件驱动程序。 ‘黄色’区块是用于控制环路的控制器区块。 虽然在这里使用了一个 2 极 2 零控制器，对于这个应用，一个 PI 和 PID，一个 3 极 3 零或任何其它控制器也同样适用。 这样一个模块库结构使得图形化和理解图 26中显示的整个系统软件流程变得更加方便。 它还可多种功能性的简单使用、添加和删除。

通过执行一个递增构建方法可在这个项目中充分证明这一事实。 这在下一部分进行了更加详细的讨论。

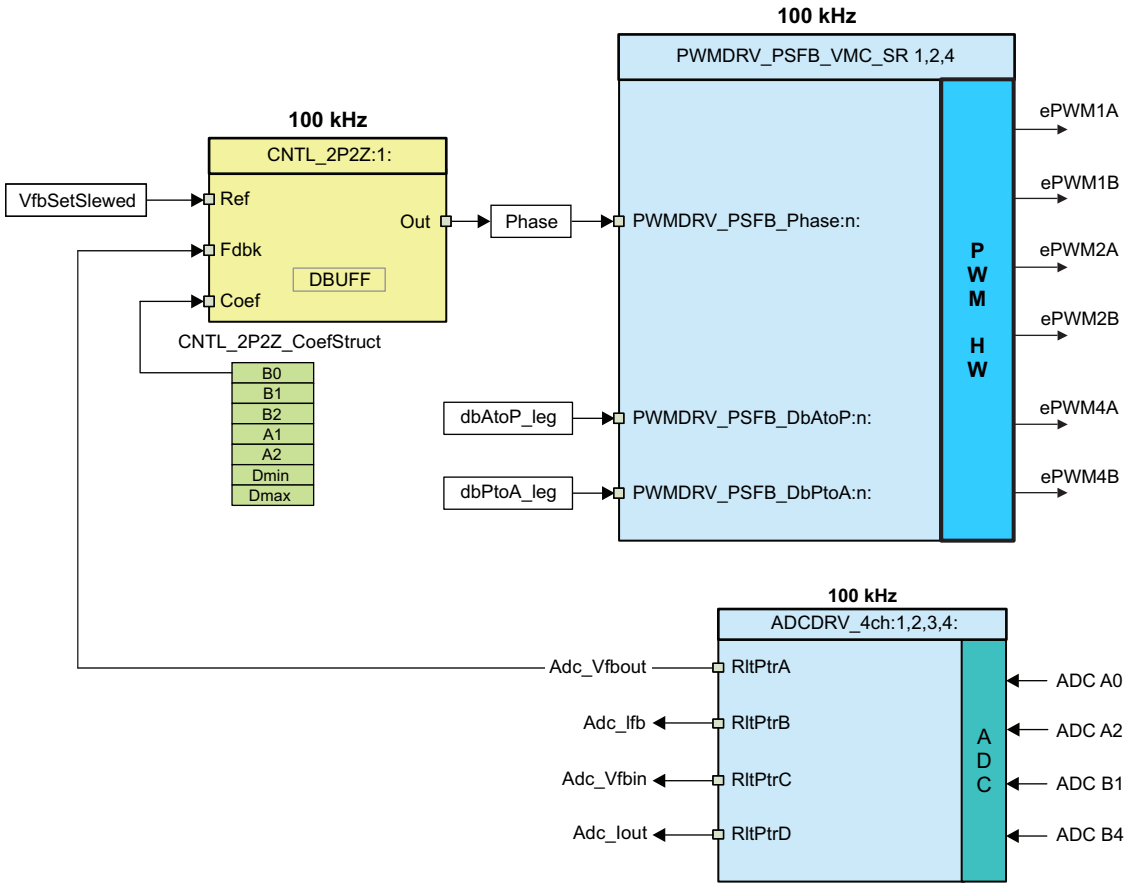


图 **26. VMC** 控制流程

此系统通常由一个电压反馈环路控制。图 26还给出了控制区块的执行速率。 例如，电压控制器的执行速率为 100kHz（与 PWM 频率一样）。 下面解释了上面执行的控制。

感测到的输出电压 (*Adc_Vfbout*) 与电压控制器中的电压基准指令 (*Vref*) 的已转换版本 (*VfbSetSlewed*) 相比较。 此电压控制器输出直接控制驱动全桥的两个桥臂的 PWM 信号间的相移。 这控制全桥的两个桥臂间的相位重叠量，以调节输出电压。 *dbAtoP_leg*和*dbPtoA_leg*值分别为全桥的'有源到无源'和'无源到有源'桥臂提供死区时间值。 这些值被用来在负载范围内实现 ZVS 和 LVS。

5.2 递增构建

这个项目被分成两个递增构建。 这使得学习和熟悉此电路板和软件变得更加容易。 这个方法对也有利于调试和测试电路板。 此构建选项显示如下。 要选择一个特定的构建选项，

1. 将 HVPSFB-Settings.h 文件中的宏 INCR_BUILD 设定为下面显示的相应构建选项。
2. 一旦选择了构建选项，则通过选择 rebuild-all compiler option（重建所有编译器选项）来编译整个项目。

6 节提供了运行每个构建选项的更多细节。

表 **6.** 针对 **VMC** 的递增构建选项

递增构建选项	
INCR_BUILD = 1	具有 ADC 反馈（检查 PWM 驱动电路和感测电路）的开环 PSFB 驱动
INCR_BUILD = 2	闭合电压环路（VMC 模式中的全 PSFB）

6 运行递增构建的过程 - VMC

提出 PSFB 系统的主源文件，ISR 汇编文件和针对 C 语言框架的项目文件位于以下目录中（请使用软件包的最新版本。 版本 1.1 是截止 2012 年 3 月的最新软件版本）..\controlSUITE\development_kits\HVPSFB_v1.1\HVPSFB_VMC. 包含这个软件的项目针对 Code Composer Studio v4。

WARNING

在电路板上会有高压出现。 它应该只在实验室环境中由经验丰富的电源专业人员处理。 为了安全评估这个电路板，应该使用一个适当隔离的高压 **DC** 电源。 在 **DC** 电源被施加到电路板上之前，必须将一个电压计和一个适当的阻性或电力负载连接至输出。 当此器件加电时，一定不要触摸此器件。

请按照以下步骤来构建和运行包括在*HVPSFB_VMC*软件内的示例。

6.1 构建 1：使用 ADC 反馈的开环检查

6.1.1 目的

这个构建的目的是为了评估系统的开环运行、验证 PWM 和 ADC 驱动器模块、验证电路板上的 MOSFET 驱动器电路和感测电路并逐渐熟悉 Code Composer Studio 的操作。 由于这个系统正在开环路运行，ADC 测量值只用于这个构建内的仪器仪表用途。 研究了构建并运行一个项目所需的步骤。

6.1.2 概述

已经配置了 Build1 中的软件，这样您就能够通过查看示波器上的输出波形并且通过交互地调整 Code Composer Studio 上的相位来观察相位变化对输出电压的影响，从而快速地评估移相全桥 PWM 驱动器模块。 此外，您可以通过在观察视图中查看 ADC 采样数据来评估 ADC 驱动器模块。

如5 节中所提到的那样，PWM 和 ADC 驱动器宏例示在 _DPL_ISR 内部执行。图 27显示了这个构建中使用的区块。ePWM1A 和 ePWM1B 驱动 Q2 和 Q3 全桥开关，而 ePWM2A 和 ePWM2B 驱动 Q1 和 Q4 全桥开关。 ePWM4A 和 ePWM4B 驱动 Q6 和 Q5 同步整流器开关。

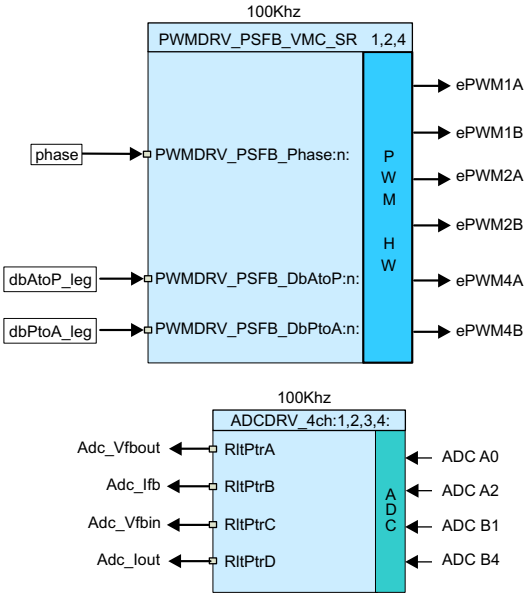


图 **27.** 构建 **1** 软件区块

这些 PWM 信号所需的生成频率为 100kHz（一个 10µs 周期）。 当 MCU 运行在 60MHz 时，ePWM1，ePWM2 或 ePWM4 的时基计数器的一个计数的相应值为 16.667ns。 这意味着一个 10µs 的 PWM 周期将等于时基计数器（TBCNT1，TBCNT2 和 TBCNT4）的 600 次计数。 ePWM1 和 ePWM2 模块被配置成运行在上数模式，而 ePWM4 运行在上-下计数模式。ePWM1A 和 ePWM1B 输出运行在 50% 占空比并且互相补充。 相似地，ePWM2A 和 ePWM2B 运行在 50% 占空比并且互相补充。 针对 ePWM2 时基的相位可相对于 ePWM1 相位动态变化。 图 14中显示了这些 PWM 波形。

到 PWM 驱动器模块的*相位*输入决定了 PWM1 和 PWM2 时基之间的相移。 *相位*值控制 PWM 信号之间的重叠量，这些信号驱动全桥对角相对开关对。 随着*相位*增加，重叠量增加，这增加了传输至次级的能量。 TBPHS2 值由输入*相位*指令生成。

表 7给出了针对一个值为 599 的 TBPRD 值所推导出的示例 TBPHS2 值。

表 **7.** 针对基准的相位值

相位 **(Q24)**	**TBPHS2 =（相位 * TBPRD/2^25）**	相移度数
2097152d	37	22.5
8388608d	149	90
16776704d	299	180

请注意，全桥的每对对角开关在一个 PWM 周期内重叠一次。 这意味着最大的重叠出现在相移接近 180° 时。

汇编 ISR 在 ePWM1 的一个 ZRO (TBCNT1 = 0) 事件上触发。 这是执行控制驱动器宏，以及更新时基相位寄存器（TBPHS2 和 TBPHS4）的位置。

在何处对 ADC 输入进行采样是一个重要考虑因素。 ADC 输入信号的完整性非常重要，这是因为它决定了模拟和数字域接口信号的输出。 在功率级中将一个开关调整为 ON 或 OFF 会导致某些噪声或对信号的干扰，这些信号此时在这一点上被感测。 即使使用所提供的全部信号滤波以避免出现在 ADC 输入上的噪声，每次采样 ADC 输入时应该小心谨慎，这样可避免这个干扰。

而且，如2.5 节中讨论的那样， 最好在开关周期内的适当位置上对感测到的输出电压进行采样，在这一点上，输出电压值接近于它的平均值。 为了实现这一操作，在尽可能获得无噪声采样，并且还可采样平均输出电压时，对 ADC 输入信号进行采样。 对于全桥，通过在两个对角开关间的重叠中点进行采样可实现这一操作（在这里，重叠是指两个开关同时 ON 时的周期 - 尽可能地远离 MOSFET 开关）。 这避免了反应在 ADC 结果上的任何开关噪声。 C2000 器件上 ADC 和 PWM 模块的灵活性可实现这样的 ADC 转换的精准且灵活触发。 ADC 驱动器模块被用来读取 12 位 ADC 结果并将它们转换为 Q24 值。 在每个 PWM 周期内，PWM2 SOCA（转换 A 开始）被用来触发 5 个 ADC 转换。

6.1.3 保护

在这个阶段，是时候介绍这个项目所使用的关断机制了。 在这里，使用片载模拟比较器 1 来执行针对变压器初级电流的过流保护。使用内部 10 位 DAC 来设定基准触发电平并将其馈入这个比较器的反相端子。 这些比较器输出被配置为，只要感测到的电流大于设定的限值，就在 ePWM1 和 ePWM2 上生成一个单次触发操作。 这个 C2000 器件上触发机制的灵活性为在不同的触发事件上采取不同的操作提供了可能性。 在这个项目中，为了保护功率级，ePWM1A，ePWM2A 和 ePWM2B 输出被立即驱动为低电平。 在执行器件复位前，这些输出保持这个状态。 要获得与这些计算结果相关的更多细节，请参见www.ti.com/controlsuite内的 HVPSFB-Calculations.xls 文件。

软件中还执行输入欠压和过压闭锁。

6.1.4 资源映射

在表 8中总结了 C2000 微控制器和 HVPSFB 级之间的关键信号连接。 请注意，VMC 和 PCMC 项目的 PWM 映射是不同的。 控制器卡上的跳线（J2，J3 - PCMC 和 VMC PWM 驱动跳线使能）需要针对 VMC 模式正确配置（缺省跳线位置被设定用于 PCMC 运行）。 下面是针对这两个模式的跳线配置：

- PCMC：J2(1) → J3(1)，J2(2) → J3(2)，J2(3) → J3(3)，J2(4) → J3(4)，J2(7) → J3(7)，J2(8) → J3(8)
- VMC：J2(1) → J3(3)，J2(2) → J3(4)，J2(3) → J3(1)，J2(4) → J3(2)，J2(7) → J3(8)，J2(8) → J3(7)

表 8. HVPSFB 信号接口基准 - VMC

信号名	说明	连接至 C2000 控制器
ePWM-1A	针对全桥开关 Q2 的 PWM 驱动	GPIO-00
ePWM-1B	针对全桥开关 Q3 的 PWM 驱动	GPIO-01
ePWM-2A	针对全桥开关 Q1 的 PWM 驱动	GPIO-02
ePWM-2B	针对全桥开关 Q4 的 PWM 驱动	GPIO-03
ePWM-4A	针对同步整流器开关 Q6 的 PWM 驱动	GPIO-06
ePWM-4B	针对同步整流器开关 Q5 的 PWM 驱动	GPIO-07
Vout	PSFB 输出电压	ADC-A0
Ifb	变压器初级电流	ADC-A2 和 COMP1A
Ifb	变压器初级电流	ADC-A4 和 COMP2A (1)
Vfbin	PSFB 输入电压	ADC-B1
Iout1	PSFB 输出电流	ADC-B3
Iout2	PSFB 输出电流（被严重过滤）	ADC-B4

(1) 控制器卡上的缺省跳线 J4 配置。 输入可由跳线 J4 选择配置。

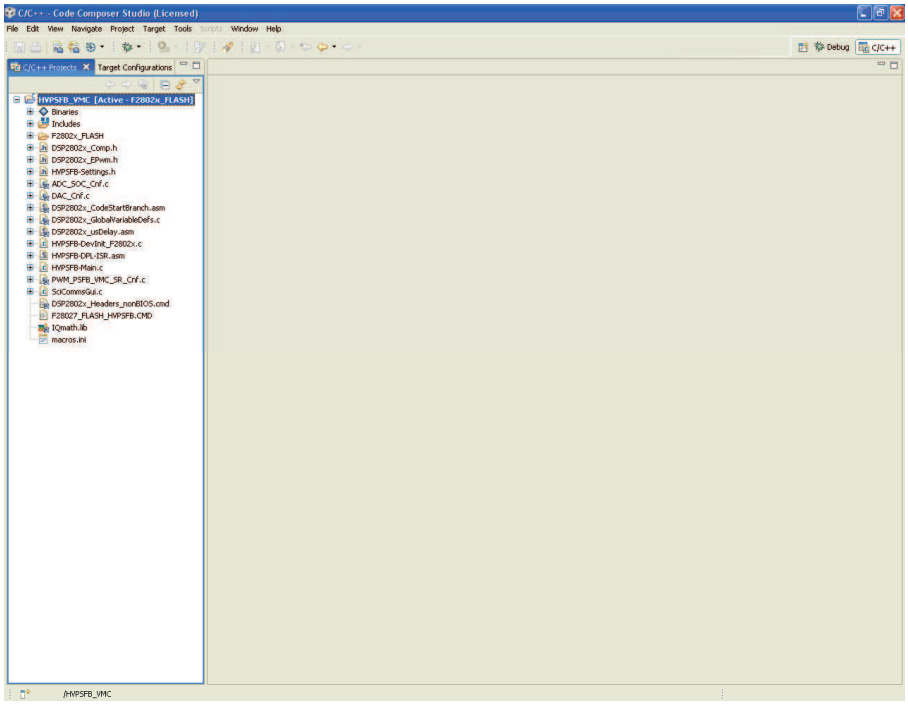
注: HVPSFB_PCMC 和 HVPSFB_VMC 项目使用位于它们自己项目目录内的 DSP2802x_Comp.h 和DSP2802x_EPWM.h 头文件，而不是那些位于device_support 目录内的文件。 这两个 device_support 目录内的文件将在*ControlSuite*更新时升级，到那时这些更新过的文件可用于这个两个项目。

6.2 过程

6.2.1 启动 Code Composer Studio 并打开一个项目

使用以下步骤来快速执行这个构建：

1. 请确保未插装基板上的跳线 J6，而插装了跳线 J8。
2. 缺省情况下，控制器卡的 Piccolo 宏上的电阻器 R6 和跳线 J1 被移除以启用 - 闪存引导。 重新插装电阻器 R6 和跳线 J1 来运行并编辑 RAM 或编辑闪存。
3. 为了实现仿真，将 USB 连接器接至 Piccolo 控制器卡。 建议使用一个输出被设定为大约 400V DC 的合适的隔离式 DC 电源。 在它被连接至主电路板的 J1 和 J2 之前，DC 电源应该保持关闭状态。
4. 使用一条 20AWG 600V 电线将电源接至 J1 和 J2。 请确保这个连接的极性是正确的。 在 J3 和 J4 的 DC 输出上施加一个适当的阻性 或 DC 电力负载至移相全桥系统。
5. 此时不要接通 400V DC 电源。
6. 将 TP1 和 TP2 之间的偏置电源加电至大约 11V DC（这个电压必须小于 12V）。
7. 双击桌面上的 Code Composer Studio 图标。
8. 将 Code Composer Studio 扩展到您的屏幕大小。
9. 如果欢迎屏幕打开的话，请关闭它。
10. 一个项目包含开发一个可执行输出文件 (.out) 所需的全部文件和构建选项，此可执行输出文件能够在 MCU 硬件上运行。 在菜单栏上，单击 → Project Import Existing CCS/CCE Eclipse Project（导入现有 CCS/CCE Eclipse 项目）并在*Select root directory（选择根目录）*下导航至并选择*..\controlSUITE\development_kits\HVPSFB_v1.1\HVPSFB_VMC*目录。 请确保 Project 标签页 HVPSFB_VMC 被选中。
11. 单击 Finish。 这个项目将调用所有需要的工具（编译器、汇编程序、连接程序）来建立项目。
12. 单击 Project 左侧的加号符号 (+)（在左侧的项目窗口中）。 您的项目窗口与以下的图形类似。



6.2.2 器件初始化、主和 ISR 文件

注: 不要对源文件做出任何改动 - 检查即可。

1. 通过双击项目窗口中的文件名可打开并检查HVPSFB-DevInit_F2802x.c。 请注意，系统时钟、外设时钟

预分频和外设时钟使能已经被设置。 下一步，请注意共用 GPIO 引脚已经被配置。

2. 打开并检查 HVPSFB-Main.c。 请注意对 DeviceInit() 函数的调用和其它变量初始化。 此外，请注意针对不同递增构建选项（特别是您现在打算编译的构建），ISR 初始化以及 (;;) 环路所用背景的代码。
3. 确定并检查以下专门针对构建 1 的初始化代码下，主文件内的代码。这是控制流程内*PWMDRV_PSFB_VMC_SR*区块被连接和初始化的地方。

```
PWMDRV_PSFB_VMC_SR_CNF(1, PWM_PRD, 1, 1); // ePWM1 and ePWM2, Period=PWM_PRD,
                                          // SR_Enable=1 (ePWM4), Comp1_Prot=1
                                          // Connect the PWMDRV_PSFB_VMC_SR driver block
PWMDRV_PSFB_Phase1  = &phase              // Point to the phase net
PWMDRV_PSFB_DbAtoP1 = &dbAtop_leg;        // Point to the left leg dead band adjust
PWMDRV_PSFB_DbPtoA1 = &dbPtoA_leg:        // Point to the right leg dead band adjust
```

4. 确定并检查以下专门针对构建 1 的初始化代码下，主文件内的代码。这是控制流程内 ADCDRV_4CH 区块被配置、初始化和连接的地方。

```
#define    Vfb_outR      AdcResult.ADCRESULT1 //
#define    IfbR          AdcResult.ADCRESULT2 //
#define    Vfb_inR       AdcResult.ADCRESULT //

#define    IoutR        AdcResult.ADCRESULT4 //

// Channel Selection for Cascaded Sequencer
     ChSel[0] = 0;    // A0 - O/P Voltage - Dummy
     ChSel[1] = 0;    // A0 - O/P Voltage
     ChSel[2] = 2;    // A2 - Transformer Primary Current
     ChSel[3] = 9;    // B1 - I/P Voltage

     ChSel[4] = 12;   // B4 - Iout2

     TrigSel[0] = 7;  // O/P Voltage sampling triggered by EPWM2 SOCA - Dummy
     TrigSel[1] = 7;  // O/P Voltage sampling triggered by EPWM2 SOCA
     TrigSel[2] = 7;  // Transformer Primary Current sampling triggered by EPWM2 SOCA
     TrigSel[3] = 7;  // I/P Voltage sampling triggered by EPWM2 SOCA

     TrigSel[4] = 7;  // Iout sampling triggered by EPWM2 SOCA

     EALLOW;
     AdcRegs.SOCPRICTL.bit.SOCPRIORITY = 4; // SOC0-3 are high priority
     EDIS;
ADC_SOC_CNF(ChSel,TrigSel,ACQPS, 16, 0);    // ACQPS=8, No ADC channel triggers an interrupt
IntChSel > 15,
                                            // Mode= Start/Stop (0)
                                            // ADC feedback connections
        ADCDRV_4ch_RltPtrA = &Adc_Vfbout;
```

5. 打开并检查 HVPSFB-DPL-ISR.asm。 请注意 _DPL_Init 和 _DPL_ISR 部分。 这分别是 PWM 和 ADC 驱动器宏例示完成初始化和运行结束的地方。 您可以选择关闭被检查的文件。

6.2.3 建立和加载项目

1. 在 HVPSFB-Setting.h 中将递增构建选项选为 1。

注: 只要您改变了*HVPSFB-Setting.h*中的递增构建选项，请始终进行“重建全部”操作。

2. 单击 Project → "Rebuild All"（项目 → “重建全部”）按钮并观察在构建窗口中运行的工具。

3. 单击 Target → "Debug Active Project"（目标 → “调试激活项目”）。 如果还未选择，Code Composer Studio 将要求您打开一个新目标配置文件。 如果已经为这个连接创建了一个有效目标配置文件，您可前往。 在 New targe Configuration Window（新目标配置窗口）内为您将操作的目标键入以 .ccxml 为后缀的文件名（例如：xds100-F28027.ccxml）。 选中 "Use shared location"（使用共用位置） 并单击 Finish。
4. 在打开的 .ccxml 文件中，将 Connection（连接）选为 "Texas Instruments XDS100v2 USB Emulator"（德州仪器 (TI) XDS100v2 USB 仿真器），向下滚动并选择 "TMS320F28027" 器件。 单击 Save（保存）。
5. 单击 Target → "Debug Active Project"。 根据所选择的项目配置，此程序将被载入闪存或 RAM 存储器。 这个项目只随 F2802x_FLASH 配置提供。 您现在应该处于 Main() 的起始位置。

6.2.4 调试环境窗口

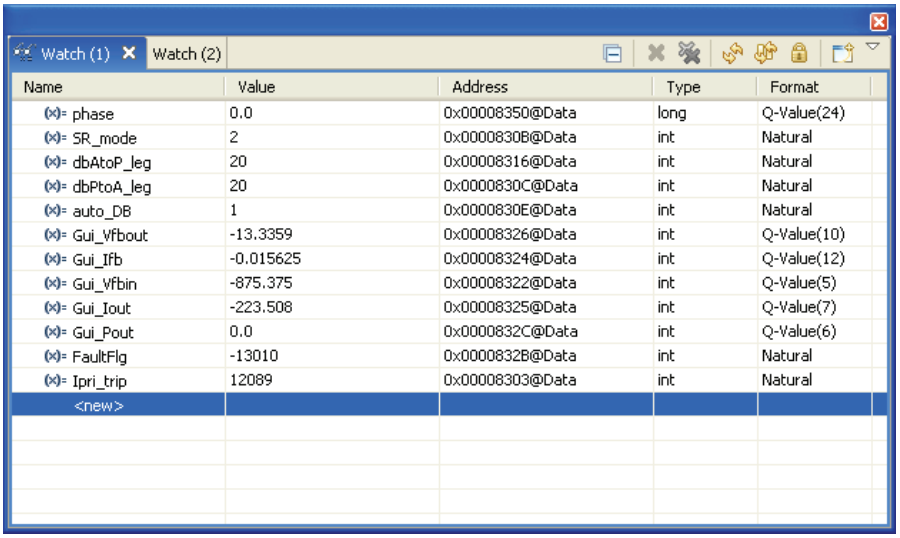
在调试代码时观察本地和全局变量是标准调试做法。 在 Code Composer Studio 内有多种不同的方法来进行这个操作，诸如存储器视图和观察视图。 此外，Code Composer Studio 能够生成时（和频）域图。 这使您能够使用图形窗口来观察波形。

如果调试环境启动时观察视图没有打开，那么打开一个新的观察视图并通过下面给出的步骤为其添加不同参数。

1. 单击菜单栏上的 View → Watch（视图 → 观察）。
2. 单击 “Watch (1)" 标签页。 您可将任一变量添加至观察视图。

3. 在 "Name" 栏内的空白方框内输入您想观察的变量符号名并且按下键盘上的回车键。 请确保按要求修改了 "Format"（格式）。 观察视图应该与下面所显示的类似。 请注意，主代码中的某些变量此时还未被初始化，有可能包含一些垃圾值。

   FaultFlg，如果被置位的话，表示一个过流情况（上面已经讨论过了），此情况关断 PWM 输出。 PWM 输出被保持在这个状态，直到器件复位（请按照步骤内的适当流程进行操作）。 *Ipri_trip*变量针对片载比较器 1 设定内部 10 位 DAC 基准电平。请注意，这是一个 Q15 数。



6.2.5 使用实时仿真

实时仿真功能是一种特殊的仿真特性，可以让 Code Composer Studio 中的窗口在 MCU 运行时以高达 10Hz 的速率刷新。 这不但可实现图形和观察视图更新，而且使您能够改变观察和存储器窗口内的值，并使这些值的变化影响 MCU 运行方式。 例如，当调整控制律参数时，这一功能十分有用。

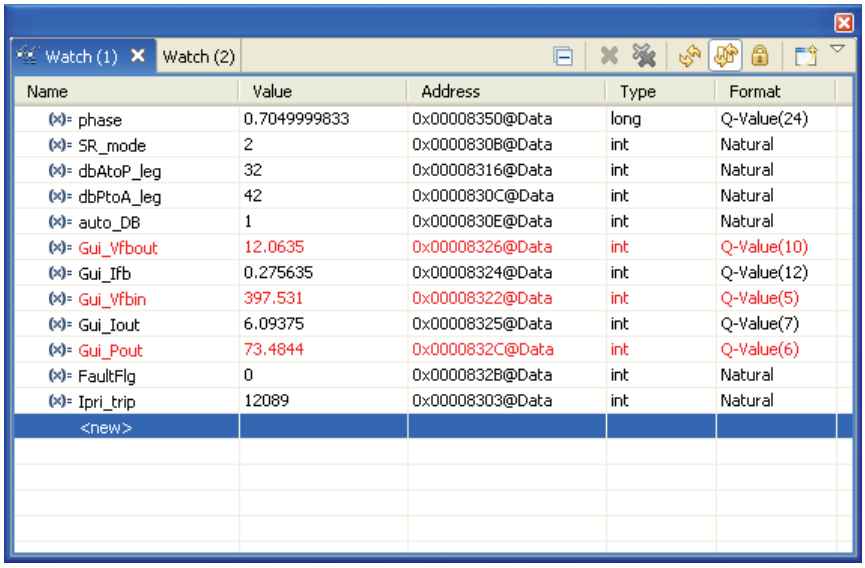
1. 通过将您的鼠标悬停在水平工具条的按钮上并单击  *Enable Silicon Real-Time Mode（启用芯片实时模式）*（当暂停时处理关键中断，可在运行时访问调试程序）按钮来启用实时模式。
2. 如果出现一个消息框，选择 YES 来启用调试事件。 这将把状态寄存器 1 (ST1) 的位 1（DGBM 位）置位为“0”。 DGBM 是调试使能屏蔽位。 当 DGBM 位被置位为“0”时，为了更新调试器窗口，存储器和寄存器值可被传递到主机处理器。
3. 现在单击同一水平工具条上的  *Enable Polite Real-Time Mode（启用礼貌实时模式）*。
4. 当打开大量窗口时，由于仿真连接上的带宽是有限的，以持续的刷新率更新太多窗口和变量会导致刷新频率停顿。 右键单击观察视图内的  按钮并选择“*Customize Continuous Refresh Interval.（定制持续刷新间隔。）*..”。 您可以通过改变*Continuous refresh interval (milliseconds)（持续刷新间隔（毫秒））*值来使针对观察视图的刷新率降下来。 通常 4000ms 的刷新率可满足这些运用的要求。
5. 对于观察视图，单击*Continuous Refresh（持续刷新）*按钮  。

6.2.6 运行代码：

1. 使用 <F8> 键来运行代码，或者使用工具条上的 RUN（运行）按钮，或者使用菜单栏上的 Target → Run（目标 → 运行）。
2. 在观察视图中，将变量相位设定为 0.015625 (Q24)。 这个变量表示*PWMDRV_PSFB_VMC_SR*模块的相移指令。 不要使用小于 0.005*的相位值*。
3. 在 DC 输出上，将一个适当的阻性负载施加到 PSFB 系统。 12V 输出时，以汲取大约 3A-6A 电流的负载作为开始比较好。

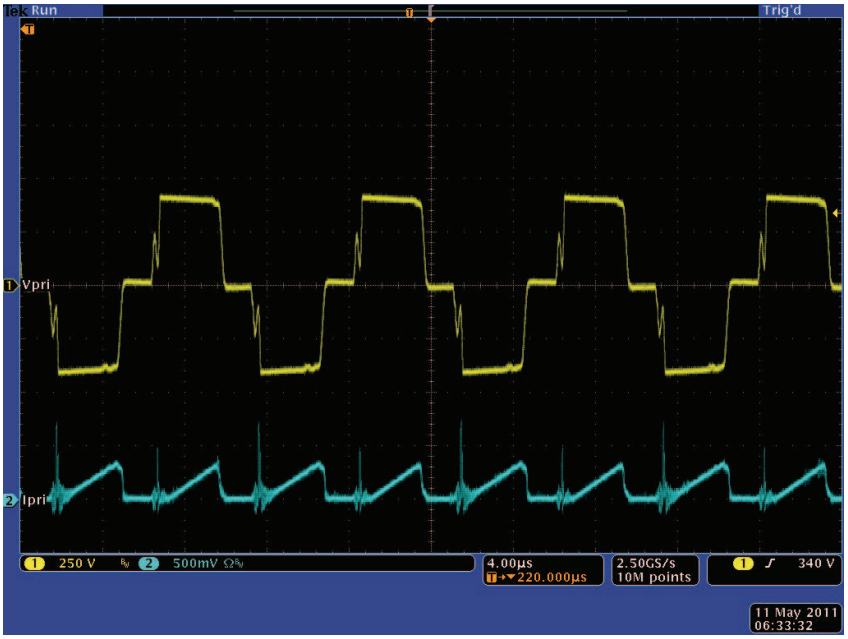
> 注: 出于安全考虑，建议使用一个隔离式 DC 电源来为电路板提供 400V DC 输入。

4. 在 J1，J2 上为输入提供 400V DC 电源。
5. 在观察视图中，通过将*相位*设定为一个更高的值 0.1（比如说）来增加相位指令。 输出电压应该增加。仔细观察输出电压，不应该让这个值超过电路板的承受能力。 请牢记，当您使用一个特定的*相位*值进行操作时，如果负载突然减少，输出电压将上升。 因此，当在构建 1 中运行时，不要突然改变负载或大幅增加相位指令。
6. 对于不同的*相位*值，请在观察视图中观察不同的 ADC 结果。
7. 这个观察视图是与一个*相位*指令为 0.705 (Q24) 的系统相对应的观察视图，此系统的输入电压大约为 400V，输出为 12V 时的负载大约为 6A。



Name	Value	Address	Type	Format
phase	0.7049999833	0x00008350@Data	long	Q-Value(24)
SR_mode	2	0x0000830B@Data	int	Natural
dbAtoP_leg	32	0x00008316@Data	int	Natural
dbPtoA_leg	42	0x0000830C@Data	int	Natural
auto_DB	1	0x0000830E@Data	int	Natural
Gui_Vfbout	12.0635	0x00008326@Data	int	Q-Value(10)
Gui_Ifb	0.275635	0x00008324@Data	int	Q-Value(12)
Gui_Vfbin	397.531	0x00008322@Data	int	Q-Value(5)
Gui_Iout	6.09375	0x00008325@Data	int	Q-Value(7)
Gui_Pout	73.4844	0x0000832C@Data	int	Q-Value(6)
FaultFlg	0	0x0000832B@Data	int	Natural
Ipri_trip	12089	0x00008303@Data	int	Natural
<new>				

8. 下面的示波器捕捉显示了上面所描述的情况下可见的变压器初级电压和感测到的初级电流。



9. 缺省情况下，同步整流器运行在模式 2 下。您可以通过在观察视图中将 SR_mode 变量改为 0，1 或 2 来改变它们的运行模式。 请观察被汲取的输入电流的变化以及使用不同 SR 模式时，输出电压的变化。 您还可以探测驱动同步整流器开关的 PWM 波形。 当运行在极低负载或者输出电压极低（少于 6V）时，不要在不同的 SR 模式间进行改变。 在这些情况下，使用缺省 SR 模式 2。
10. 尝试不同的相位值并观察相应的 ADC 结果。 以小步长来增加相位。 始终小心观察输出电压，这个值不应超过电路板的处理能力。 诸如 PWM 栅极驱动信号、输入电压和电流以及输出电压等的不同波形也可使用一个示波器进行探测。 在探测这些用于隔离式 DC-DC 转换器的高压和高电流时，应该采取适当的安全防护措施并考虑合适的接地要求。
11. 实时模式时，完全停止 MCU 是一个两步过程。 等待几秒钟，400V DC 输入关闭。 首先，通过使用工具条上的 Halt（暂停）按钮，或者使用 Target → Halt（目标 → 暂停）来暂停处理器。 然后再次单击此  按钮来使 MCU 离开实时模式，然后复位 MCU。
12. 使 Code Composer Studio 继续运行，用于下一个应用，或者选择将其关闭。

6.3 构建 2：闭合电压环路（VMC 模式中的整个 HVPSFB）

6.3.1 目的

这个构建的目的在于验证整个基于 VMC 的 HVPSFB 项目在 Code Composer Studio 环境中的运行。

6.3.2 概述

图 28显示了这个构建中所使用的软件区块。 PWM 和 ADC 驱动器区块的使用方法与在之前构建中的使用方式一样。 一个 2 极 2 零控制器用于电压环路。 根据应用的控制环路需求的不同，可使用诸如一个 PI，一个 3 极 3 零等其它控制器区块。 如图 28中所示，电压环路区块的执行频率为 100KHz。 CNTL2P2Z 是一个由无限脉冲响应 (IIR) 滤波器结构实现的二阶比较器。 这个函数与任何外设无关，因此就不需要 CNF 函数调用。

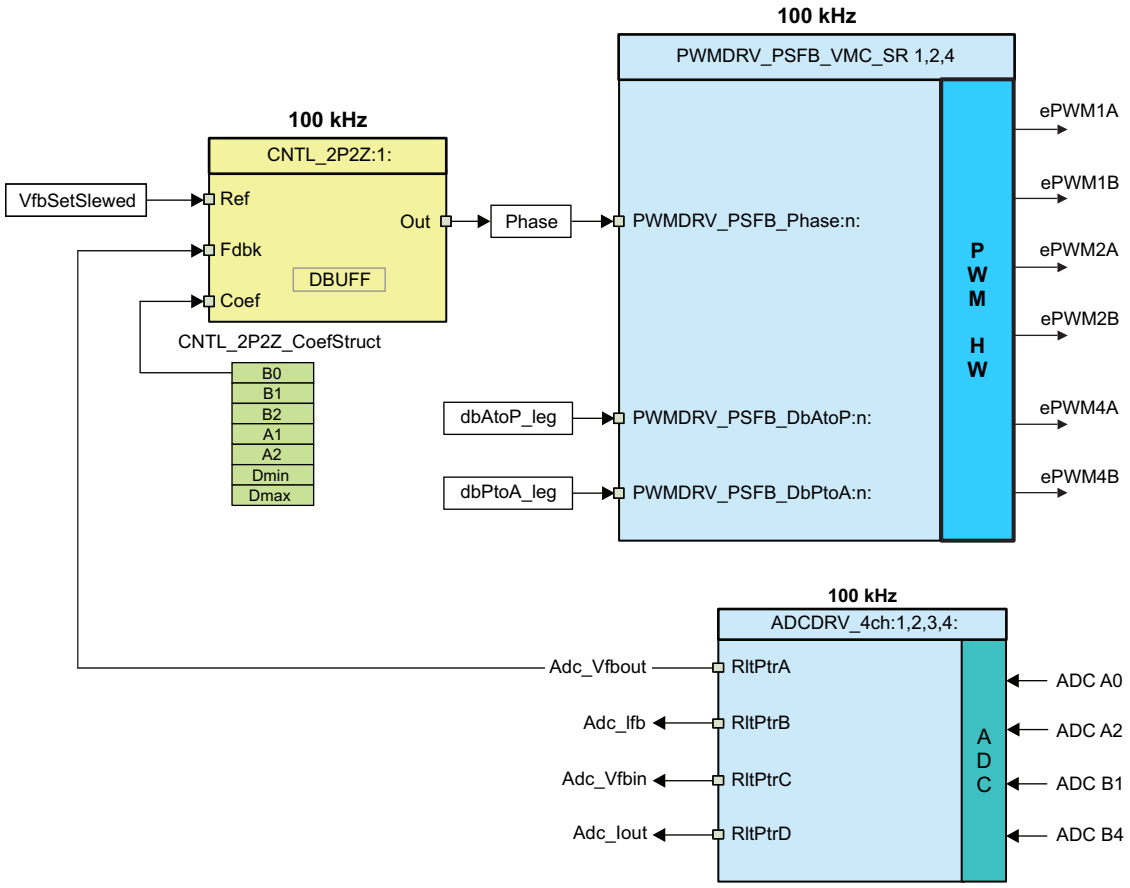


图 **28.** 构建 **2** 软件区块

将被修改的 5 个系数被存储为结构 CNTL_2P2Z_CoefStruct1 的元素组件，此结构的其它元素被用来钳制控制器输出。 如果系统需要多个环路，CNTL_2P2Z 区块可被实例化多次。 每个实例可具有单独的系数集合。 通过反复试验直接独立操纵这 5 个系数几乎是不可能的，并且要求数学分析或诸如 matlab，mathcad 等工具的帮助。 这些工具提供伯德图、根轨迹法和其它特性来确定相位裕量度、增益裕度等。

为了简化环路调整且无需复杂数学或分析工具，通过将 P，I 和 D 的更加直观的系数增益简便映射为 B0，B1，B2，A1 和 A2，系数选择问题已经从自由度 5 减为自由度 3。 这样可实现对 P，I 和 D 的独立且逐步的调节。 下面给出了这些映射等式。

补偿器区块 (CNTL_2P2Z) 有两个极和两个零，并基于普通 IIR 滤波器结构。 它有一个基准输入和一个反馈输入。 对于电压环路，此反馈是感测输出电压 (*Adc_Vfbout*)，而到控制器的基准输入是输出电压基准指令 (*Vref*) 的已转换版本 (*VfbSetSlewed*)。 传输函数给出如下：

$$\frac{U(z)}{E(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

PID 控制器的递归形式由不同的等式给出：

$$u(k) = u(k-1) + b_{0e}(k) + b_{1e}(k-1) + b_{2e}(k-2)$$

其中：

$$b_0 = K_p' + K_i' + K_d'$$
$$b_1 = K_p' + K_i' - 2K_d'$$
$$b_2 = K_d'$$

这个的 z 域传输函数形式为：

$$\frac{U(z)}{E(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 - z^{-1}} = \frac{b_0 z^2 + b_1 z + b_2}{z^2 - z}$$

将它与普通形式相比较，您可发现 PID 只是 CNTL_2P2Z 控制的一个特殊情况，其中：

$$a1 = -1 \text{ and } a2 = 0$$

对于电压环路，这些 P，I 和 D 系数为：Pgain，Igain 和 Dgain。 这些 P，I 和 D 系数所采用的格式为 Q26。 为了简化 GUI 环境（或者 Code Composer Studio 观察视图）的调整，这 3 个系数被进一步调整为 0 至 999 之间的值（Pgain_Gui，Igain_Gui 和 Dgain_Gui）。

在 GUI 环境中，可使用 2 极 (fp1，fp2)，2 零 (fz1，fz2) 和增益 (Kdc) 来调整电压环路。 这些参数提供格式为 I5Q10 的 b2_Gui，b1_Gui，b0_Gui，a2_Gui 和 a1_Gui 系数，然后这些系数被转换为用于 2P2Z 控制器的 5 个 Q26 系数。 虽然不建议这么做，但是 b2_Gui，b1_Gui，b0_Gui，a2_Gui 和 a1_Gui 值也可使用观察视图从 Code Composer Studio 环境直接转换。 要获得与这些计算结果相关的细节，请参见www.ti.com/controlsuite内的 HVPSFB-Calculations.xls 文件。 对于根据极，零和增益以及开关频率推导出系数值的等式也在 GUI 源文件中明确给出。

这个项目通过在执行期间提供系数间的轻松切换，实现对两个环路调整方法的简便评估。 可通过简单单击 GUI 上的 2P2Z(On) 和 PID(Off) 按钮或在 Code Composer Studio 的观察视图中将 pid2p2z_GUI 变量从 0 改为 1 来完成这一操作。 GUI 环境内基于 PID 的环路调整 (pid2p2z_GUI=0) 曾被用作一个起始点。 然后，与这些 PID 经调整系数相对应的极、零和增益被用作一个起始点，以根据第二个方法 (pid2p2z_GUI=1) 对环路进行进一步调整。 然后，通过在 GUI 环境内改变极、零和增益来获得更佳结果来调整为最优动态性能。 缺省情况下，使用基于这些经调整的极、零和增益值的系数（pid2p2z_GUI= 1 - 缺省值）。

注： 当使用 2 极 2 零控制器系数调节来调整系统时，如果在 GUI 中选择极、零和 Kdc 值，那么这些系数 (b2，b1，b0，a2，a1) 中的任何一个的大小将大于或等于 32，那么这些系数值不由 GUI 发送给控制器。

6.4 过程

6.4.1 构建和加载项目

使用以下步骤来快速执行这个使用预先配置的工作环境的构建：

1. 按照构建 1 中的步骤 1至 2。如果您在 Code Composer Studio 的最后一次使用中操作构建 1，同一个工作区应该与项目一同打开。
2. 如果情况不是这样，您可以通过单击 File → Switch Workspace（文件 → 切换工作区）来打开用于构建 1 的工作区，然后前往至正确工作区。 如果一个工作区未被保存或被删除，请按照与构建 1 中相同的步骤 3和 4。
3. 找到并检查主文件中专门用于构建 2 的初始化代码。 这就是控制流程中，所有控制区块被配置、初始化并被连接的地方。
4. 在*HVPSFB-Setting.h*中将递增构建选项选为 2。

注： 只要您改变了*HVPSFB-Setting.h*中的递增构建选项，请始终进行“重建全部”操作。

5. 单击 Project → "Rebuild All" 按钮并观察在构建窗口中运行的工具。
6. 单击 Target → "Debug Active Project"。 根据项目配置，此程序将被载入闪存或 RAM 存储器。 这个项目只随 F2802x_FLASH 配置提供。 您现在应该处于 Main() 的起始位置。

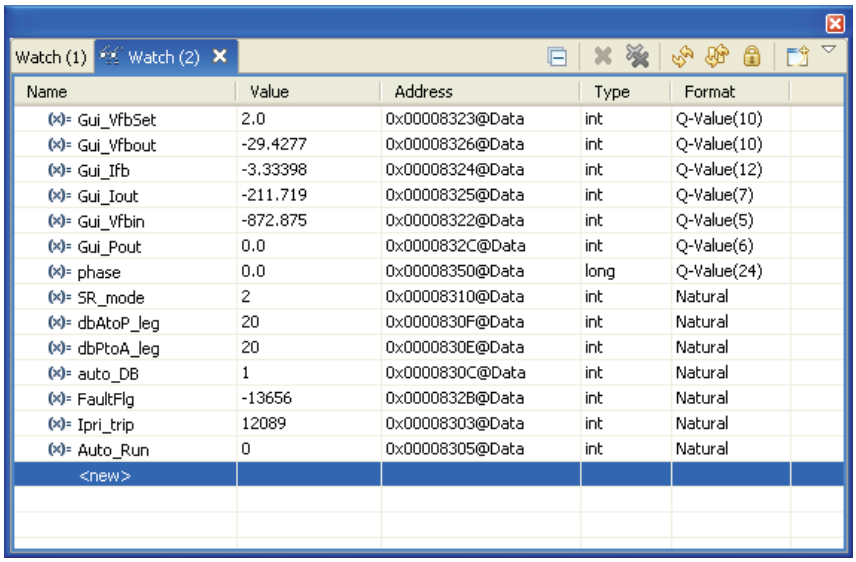
6.4.2 调试环境窗口

在调试代码时观察本地和全局变量是标准调试做法。 在 Code Composer Studio 内有多种不同的方法来进行这个操作，诸如存储器视图和观察视图。 此外，Code Composer Studio 能够生成时（和频）域图。 这使您能够使用图形窗口来观察波形。

如果调试环境启动时观察视图没有打开，那么打开一个新的观察视图并通过下面给出的步骤为其添加不同参数。

1. 单击菜单栏上的 View → Watch（视图 → 观察）。
2. 单击 “Watch (1)" 标签页。
3. 如果已经从为构建 1 保存的调试环境中打开了一个观察视图，请单击  按钮。一个 "Watch (2)" 标签页将打开，然后您可以将其拖拽至您选择的窗口中进行检查。 您可将任一变量添加至这个观察视图标签。
4. 在 "Name" 栏内的空白方框内输入您想观察的变量符号名并且按下键盘上的回车键。 请确保按要求修改了 "Format"（格式）。 观察视图应该与下面所显示的图形类似。

   请注意，主代码中的某些变量此时还未被初始化，有可能包含一些垃圾值。



   请注意，在观察视图中有额外变量。

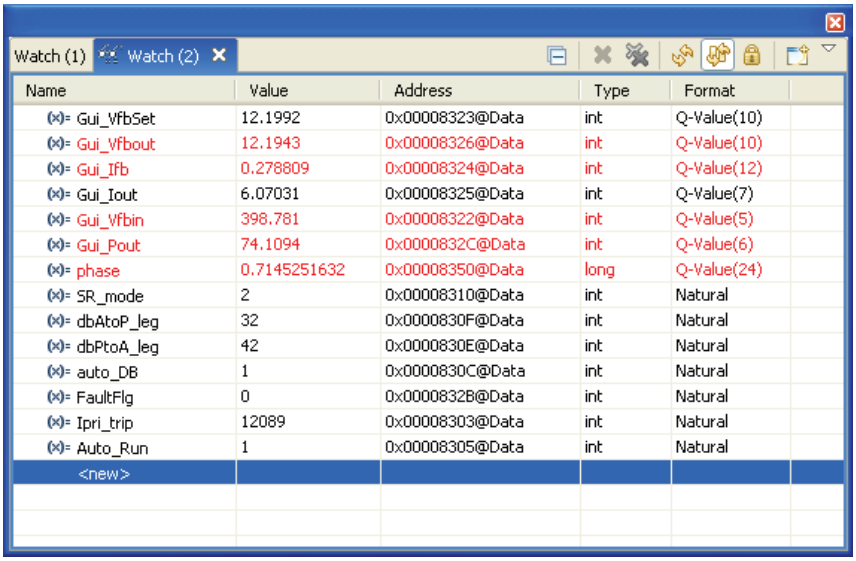
5. *Gui_VfbSet*被用来设定输出电压指令。

6.4.3 运行代码：

1. 按照构建 1 过程步骤 1至 2来启用实时模式和针对观察视图的持续刷新，也可在需要时改变观察视图的持续刷新间隔。
2. 使用 <F8> 键来运行代码，或者使用工具条上的 RUN（运行）按钮，或者使用菜单栏上的 Target → Run（目标 → 运行）。
3. 在 DC 输出上，将一个适当的阻性负载施加到 PSFB 系统。 12V 输出时，以汲取大约 3A-6A 电流的负载作为开始比较好。

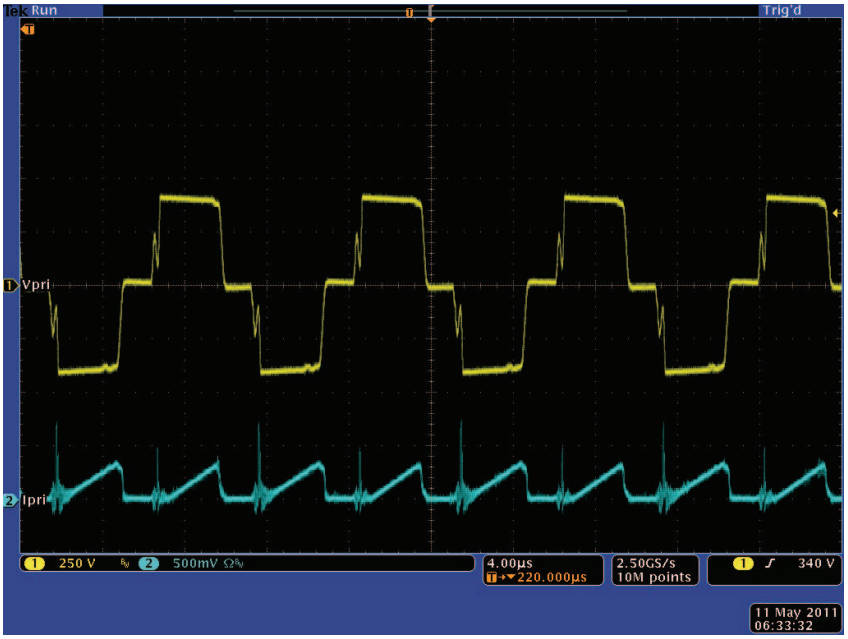
注: 出于安全考虑，建议使用一个隔离式 DC 电源来为电路板提供 400V DC 输入。

4. 在 J1，J2 上为输入提供 400V DC 电源。
5. 缺省情况下，*Auto_Run*被置位为 1。如果不为 1，请在观察视图中将其改为 1。 现在，输出电压应该开始斜升至 12V。可通过改变变量*VfbSlewRate*来更改此输出电压斜升率。
6. 下图显示了一个观察视图，此视图与输出上为 12V 的系统运行相对应，此时系统的输入电压为大约 400V，负载大约为 6A 输出。



Name	Value	Address	Type	Format
Gui_VfbSet	12.1992	0x00008323@Data	int	Q-Value(10)
Gui_Vfbout	12.1943	0x00008326@Data	int	Q-Value(10)
Gui_Ifb	0.278809	0x00008324@Data	int	Q-Value(12)
Gui_Iout	6.07031	0x00008325@Data	int	Q-Value(7)
Gui_Vfbin	398.781	0x00008322@Data	int	Q-Value(5)
Gui_Pout	74.1094	0x0000832C@Data	int	Q-Value(6)
phase	0.7145251632	0x00008350@Data	long	Q-Value(24)
SR_mode	2	0x00008310@Data	int	Natural
dbAtoP_leg	32	0x0000830F@Data	int	Natural
dbPtoA_leg	42	0x0000830E@Data	int	Natural
auto_DB	1	0x0000830C@Data	int	Natural
FaultFlg	0	0x0000832B@Data	int	Natural
Ipri_trip	12089	0x00008303@Data	int	Natural
Auto_Run	1	0x00008305@Data	int	Natural
<new>				

7. 下面的图形显示了在这个条件下捕捉到的波形。



8. 缺省情况下，同步整流器运行在模式 2 下。您可以通过在观察视图中将 SR_mode 变量改为 0，1 或 2 来改变它们的运行模式。 请观察被汲取的输入电流的变化量以及使用不同 SR 模式时，输出电压变化。您还可以探测驱动同步整流器开关的 PWM 波形。 当运行在极低负载时（低于 3A），不要在不同的 SR 模式中进行改变。 在这些情况下，使用缺省 SR 模式 2。
9. 观察负载变化对输出电压和输入电流的影响。 实际上不应对输出电压产生影响。 相似地，观察输入电压改变所产生的影响。 实际上，仍然不应对输出电压产生影响。

注: 请确保做出的这些变化在本文档技术规格中所列出的电路板处理能力范围内。

10. 诸如 PWM 栅极驱动信号、输入电压和电流以及输出电压等的不同波形也可使用一个示波器进行探测。在探测这些用于隔离式 DC-DC 转换器的高压和高电流时应该采取适当的安全防护措施并考虑合适的接地要求。
11. 实时模式时，完全停止 MCU 是一个两步过程。 等待几秒钟，400V DC 输入被关闭。 首先，通过使用工具条上的 Halt（暂停）按钮，或者使用 Target → Halt（目标 → 暂停）来暂停处理器。 然后，再次单击此  按钮来使 MCU 离开实时模式，然后复位 MCU。
12. Close Code Composer Studio。

7 参考书目

1. 《*UCC28950 600W 移相全桥应用报告*》(SLUA560)
2. 《*使用UCC28950EVM-442 用户指南*》(SLUU421)
3. H. Nene，“使用微控制器执行针对移相全桥 DC-DC 转换器的高级控制策略” PCIM 欧洲 2011，纽伦堡，德国。
4. 《*TMS320x2802x, 2803x Piccolo 增强型脉宽调制器 (ePWM) 模块参考指南*》
5. HVPSFB-Calculations.xls - 一个显示 HVPSFB 项目关键计算结果的数据表

   www.ti.com/controlsuite:...\controlSUITE\development_kits\HVPSFB_v1.1\~Docs
6. QSG-HVPSB-Rev1.1.pdf - 给出了如何使用一个直观 GUI 接口快速运行并试验 HVPSFB 项目的概述。

   www.ti.com/controlsuite: ...\controlSUITE\development_kits\HVPSFB_v1.1\~Docs
7. HVPSFB_RevB-HWdevPkg - 一个包含与基板和 Piccolo-A 控制器卡相关的多个文件的文件夹。

   www.ti.com/controlsuite:...\controlSUITE\development_kits\HVPSFB_v1.1\~HVPSFB_HWdevPkg\RevB

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 的直接或隐含权限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在其应用中使用 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/consumer-apps
DLP® 产品	www.dlp.com	能源	www.ti.com/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/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