

使用 **CLB** 扩展 **PWM** 实现 **F280049C** 驱动三个电机

TERRY DENG/ EP FAE

摘要

该文档详细描述如何利用 CLB 模块扩展 C2000 芯片的 EPWM 通道，并且满足电机控制的时序需要，可以兼容各种现成的电机 FOC 算法。

该方案以 F280049C 芯片为例子开发了一套软件参考，把原有的 16 通道 EPWM 扩展到 18 通道，从而可以单芯片同时驱动三个电机。

该方案可以广泛应用在三个电机的驱动系统里面，比如商用空调压缩机+2 个风扇，比如自平衡摄像头云台的三轴电机等等。

Contents

1	背景介绍	3
2	使用 CLB 实现 PWM 保护的方法	4
	2.1 电机驱动 PWM 的要求	4
	2.2 CLB 的硬件设计逻辑.....	5
	2.3 CLB 的具体设计代码.....	6
	2.4 CPU 的具体设计代码	9
3	测试结果和注意事项	11
4	参考文献	12

Figures

Figure 1.	电机驱动信号示意图	3
Figure 2.	商用空调应用示意图	3
Figure 3.	F280049C 的 PWM 规格书	3
Figure 4.	电机 6 路 PWM 的波形	4
Figure 5.	CLB 实现 PWM 扩展的逻辑设计	5
Figure 6.	测试结果波形	11
Figure 7.	PWM 模块同步关系	11

1 背景介绍

在三相电机变频控制应用中，通常 MCU 需要 6 个 PWM 信号来驱动三相全桥电路，如下图所示。

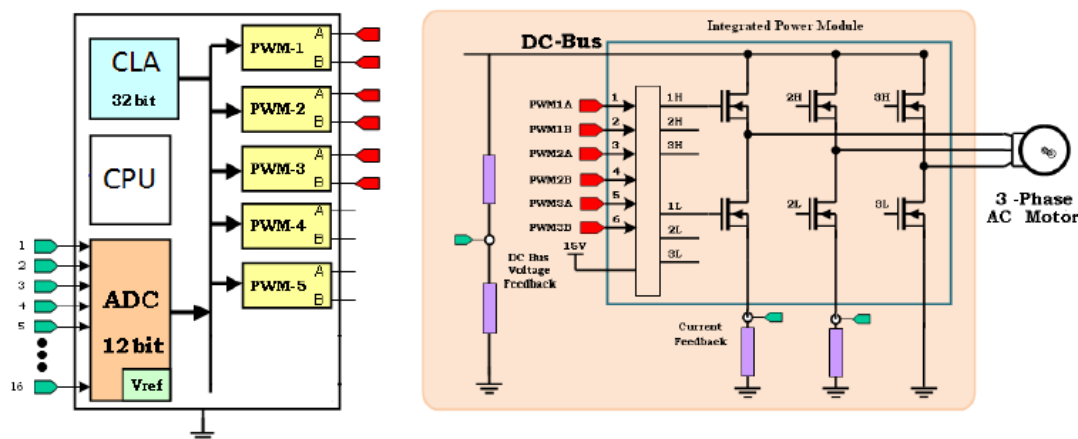


Figure 1. 电机驱动信号示意图

某些应用场合会需要控制三个电机，比如下图的商用空调就是压缩机+2 个风扇。这样的应用就需要 $6 \times 3 = 18$ 个 PWM 信号。



Figure 2. 商用空调应用示意图

TMS320F280049C 是一颗在电机驱动领域的主流 MCU，广泛应用在各种电机控制场合。如下图显示这颗芯片原生总共有 16 个 PWM 信号，所以通常来说它只能同时驱动 2 个电机。但是我们新设计方案，利用芯片内部的 CLB 模块可以纯硬件扩展出更多 2 路 PWM，从而实现总共 $16+2=18$ 路 PWM 满足同时驱动 3 个电机的要求。

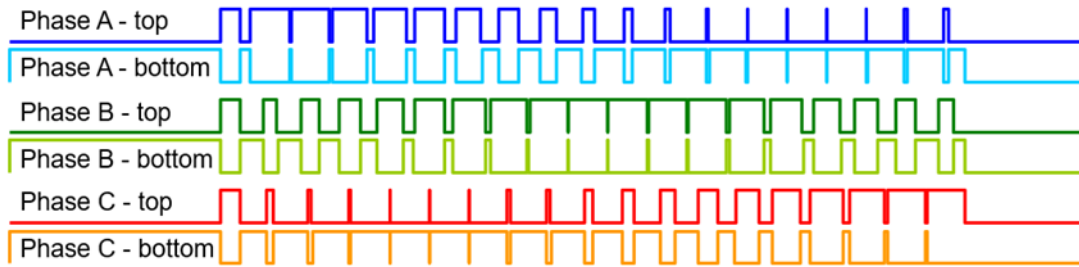
FEATURE ⁽¹⁾	F280049 F280049C	F280048 F280048C	F280045	F280041 F280041C	F280040 F280040C
CONTROL PERIPHERALS ⁽⁴⁾					
eCAP/HRCAP modules - Type 1	7 (2 with HRCAP capability)				
ePWM/HRPWM channels - Type 4	16				

Figure 3. F280049 的 PWM 规格书

2 使用 CLB 实现 PWM 保护的方法

2.1 电机驱动 PWM 的要求

为了产生互差 120 度的三相正弦波来驱动电机，6 个 PWM 信号是需要满足一定的逻辑关系，如下图所示。这个逻辑关系通常要求包括：6 个 PWM 都是相同的周期；所有周期的起始时刻始终同步；分成 3 组而每组 2 个；其中每组信号互相取反并保持死区；具有统一的故障信号同时关断。



Over time, under steady-state conditions, the correction voltages v_a , v_b , and v_c will be sine waves phase shifted by 120°.

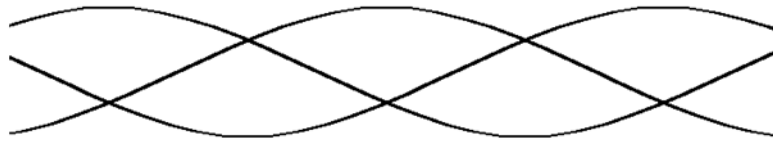


Figure 4. 电机 6 路 PWM 的波形

2.2 CLB 的硬件设计逻辑

为了满足章节 2.1 所描述的 PWM 要求，CLB 的硬件逻辑设计如下图：

- ✓ 原 EPWM 模块内部的 EPWMA 信号通过原生的 Dead Band 模块拆分成 PWMA-u 和 PWMA-d 信号，并通过原生的 PWM 功能引脚输出到芯片外部。
- ✓ 原 EPWM 模块内部的 EPWMB 信号则导入 CLB 模块 in1 输入通道，经过 CLB 的硬件逻辑拆分为 PWMB-u 和 PWMB-d 信号，并通过 XBAROUT 功能引脚输出到芯片外部，实现额外的 2 路新 PWM 信号。
- ✓ 原 EPWM 模块内部的 TZ 故障信号则导入 CLB 模块 in0 输入通道，经过 CLB 的硬件逻辑对 2 路新 PWM 信号实现统一的保护处理。
- ✓ 扩张的 2 路新 PWMB 信号将和原生态的 PWMA 信号共用同一组 EPWM 模块的 Time-Base，从而严格保证相同的周期和起始时刻

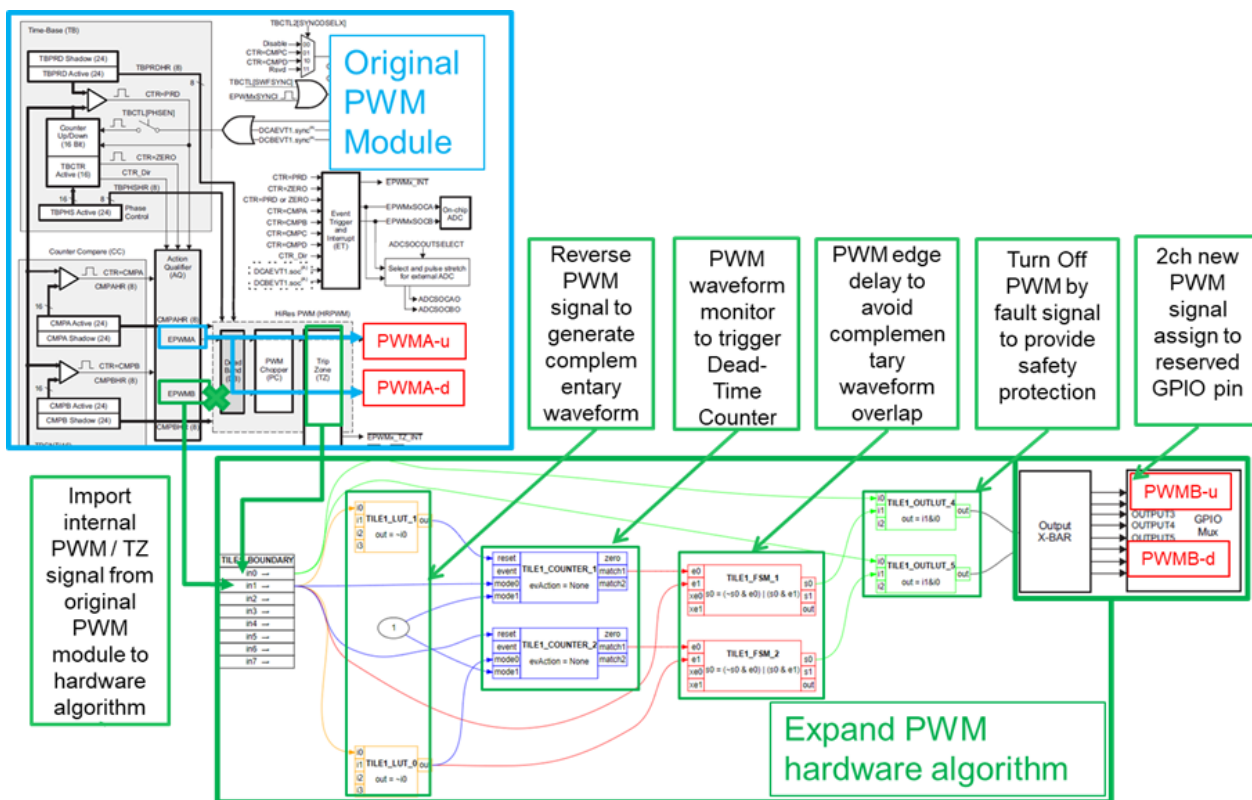


Figure 5. CLB 实现 PWM 扩展的设计逻辑

2.3 CLB 的具体设计代码

- i. LUT0 和 LUT1 模块实现把输入的 EPWMB 进行反转产生互补的扩展波形。

LUT_0		▼
eqn	~i0	
i0	BOUNDARY.in1	▼
i1	0	▼
i2	0	▼
i3	0	▼

LUT_1		▼
eqn	~i0	
i0	BOUNDARY.in1	▼
i1	0	▼
i2	0	▼
i3	0	▼

ii. COUNTER_1 和 COUNTER_2 模块实现产生扩展 PWM 波形上升沿或下降沿的死区时间

COUNTER_1		▼
reset	LUT_1.OUT	▼
event	0	▼
mode0	BOUNDARY.in1	▼
mode1	1	▼
match1_val	200	
match2_val	0	
event_load_val	0	
What action should be taken on an event trigger?	None	▼

COUNTER_2		▼
reset	BOUNDARY.in1	▼
event	0	▼
mode0	LUT_0.OUT	▼
mode1	1	▼
match1_val	200	
match2_val	0	
event_load_val	0	
What action should be taken on an event trigger?	None	▼

iii. FSM_1 和 FSM_2 模块实现在扩展 PWM 的边沿插入死区保证互补波形不会重叠

FSM_1	
eqn_out	_____
eqn_s0	(~s0 & e0) (s0 & e1)
eqn_s1	_____
e0	COUNTER_1.count_match1
e1	BOUNDARY.in1
xe0	0
xe1	0

FSM_2	
eqn_out	_____
eqn_s0	(~s0 & e0) (s0 & e1)
eqn_s1	_____
e0	COUNTER_2.count_match1
e1	LUT_0.OUT
xe0	0
xe1	0

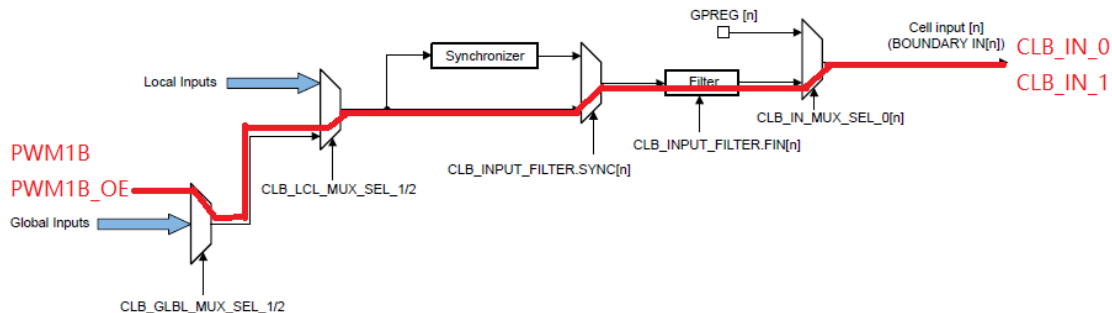
iv. OUTPUT_4 和 OUTPUT_5 模块实现根据 TZ 故障信号关断或使能扩展 PWM

OUTLUT_4	
eqn	i1&i0
i0	BOUNDARY.in0
i1	FSM_1.S0
i2	0

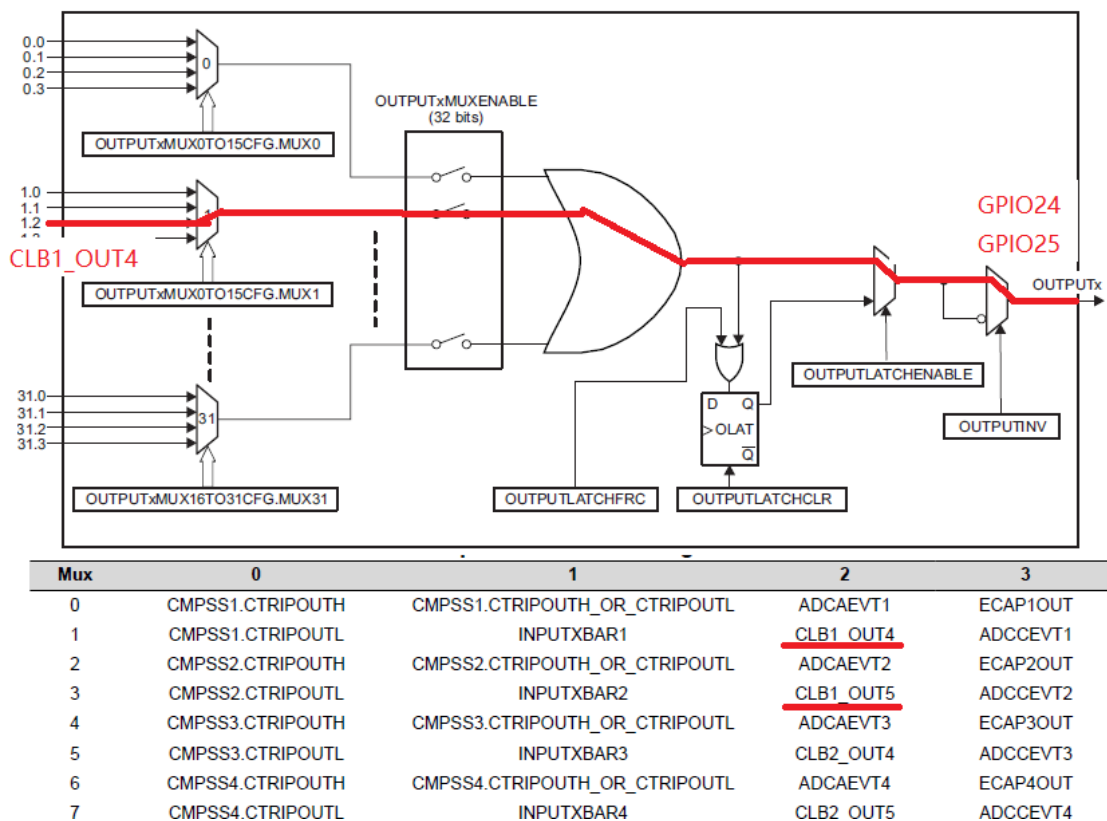
OUTLUT_5	
eqn	i1&i0
i0	BOUNDARY.in0
i1	FSM_2.S0
i2	0

2.4 CPU 的具体设计代码

CPU 主要负责设置把相应的信号输入到 CLB 模块，如下图的例子，我们需要选择 PWM1B 和 PWM1B_OE 信号从源头进入到 CLB_IN0 和 CLB_IN1 通道。



以及把扩展的 PWM 信号输出到 XBAROUT 引脚。如下图的例子，在 XBAR1 的模块里面选择 Mux1.2 做为 CLB1_OUT4 信号，然后 GPIO24 引脚的复用功能就可以选择 XBAR1 的 OUTPUT。



以下是 CPU 中选择 CLB 信号输入和输出功能的代码

```
// initialize all CLB tiles
CLB_enableCLB(CLB1_BASE);

initTILE1(CLB1_BASE);

// CLB tile 1 configuration
CLB_configGlobalInputMux(CLB1_BASE, CLB_IN0, CLB_GLOBAL_IN_MUX_EPWM1B_OE);
CLB_configGlobalInputMux(CLB1_BASE, CLB_IN1, CLB_GLOBAL_IN_MUX_EPWM1B);

CLB_configLocalInputMux(CLB1_BASE, CLB_IN0, CLB_LOCAL_IN_MUX_GLOBAL_IN);
CLB_configLocalInputMux(CLB1_BASE, CLB_IN1, CLB_LOCAL_IN_MUX_GLOBAL_IN);

CLB_configGPInputMux(CLB1_BASE, CLB_IN0, CLB_GP_IN_MUX_EXTERNAL);
CLB_configGPInputMux(CLB1_BASE, CLB_IN1, CLB_GP_IN_MUX_EXTERNAL);

CLB_selectInputFilter(CLB1_BASE, CLB_IN0, CLB_FILTER_NONE);
CLB_selectInputFilter(CLB1_BASE, CLB_IN1, CLB_FILTER_NONE);

GPIO_setPinConfig(GPIO_24_OUTPUTXBAR1);
GPIO_setDirectionMode(24, GPIO_DIR_MODE_OUT);
GPIO_setPadConfig(24, GPIO_PIN_TYPE_STD);
XBAR_setOutputMuxConfig(XBAR_OUTPUT1, XBAR_OUT_MUX01_CLB1_OUT4);
XBAR_enableOutputMux(XBAR_OUTPUT1, XBAR_MUX01);

GPIO_setPinConfig(GPIO_25_OUTPUTXBAR2);
GPIO_setDirectionMode(25, GPIO_DIR_MODE_OUT);
GPIO_setPadConfig(25, GPIO_PIN_TYPE_STD);
XBAR_setOutputMuxConfig(XBAR_OUTPUT2, XBAR_OUT_MUX03_CLB1_OUT5);
XBAR_enableOutputMux(XBAR_OUTPUT2, XBAR_MUX03);
```

3 测试结果和注意事项

下图的 in0 是 TZ 故障信号，in1 是原始 EPWMB 信号。两个 out 分别是扩展以后的 PWMB-u 和 PWMB-d 信号。

可以看出 PWMB-u 和 PWMB-d 的波形很好的跟随 EPWMB 的占空比和周期，并实现了互补以及插入死区功能。在 TZ 故障信号为高电平时立即关断了输出，只有 TZ 低电平没有故障时候才会保持输出。

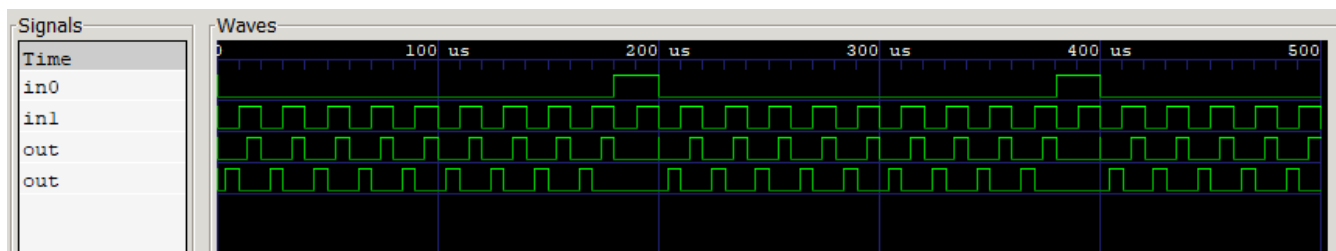


Figure 6. 测试结果波形

请注意：

- ✧ 需要使用带 C 尾缀的芯片型号，这样才会有 CLB 模块可以使用
- ✧ 扩展 PWM 只能从原生 EPWM1~4 模块选择，因为 CLB 的输入信号只支持 EPWM1~4 模块
- ✧ 分配三个电机的 PWM 信号需要考虑模块之间的同步关系，如下图所示：

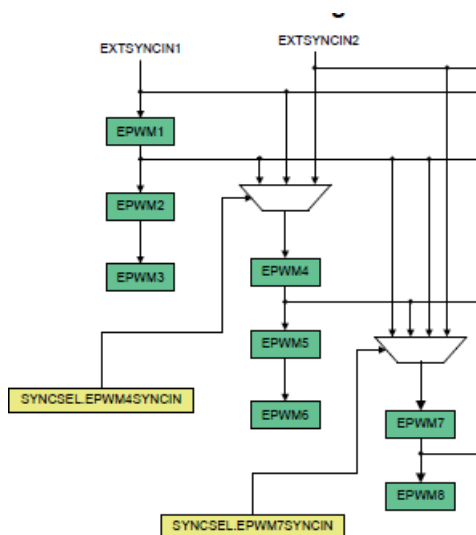


Figure 7. PWM 模块同步关系

4 参考文献

1. TMS320F28004x Datasheet (SPRS945E)
2. TMS320F28004x Technical Reference Manual (SPRUI33C)
3. CLB Tools User Guide (SPRUIR8A)
4. Design with C2000 CLB (SPRACL3)

重要声明和免责声明

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

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他安全、安保或其他要求。这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 TI 的销售条款 (<https://www.ti.com.cn/zh-cn/legal/termsofsale.html>) 或 [ti.com.cn](https://www.ti.com.cn) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

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 © 2021 德州仪器半导体技术（上海）有限公司