

宇航矢量伺服控制器在 SiP 中的设计与实现

周 禹, 刘鸿瑾, 张绍林, 施 博, 王鑫鑫, 李 宾
(北京轩宇空间科技有限公司 北京 100190)

摘要: 磁场导向控制 FOC (field-oriented control) 矢量控制算法在伺服驱动控制系统中一般由 CPU 或 DSP 实现, 难以满足航天应用中实时性较高场景下的需求。为提高宇航电机系统控制的实时性与可靠性, 从 FOC 矢量控制算法的硬件加速角度出发, 详细介绍了伺服控制器的设计, 给出了一种全数字、高性能的伺服控制器硬件加速设计方案, 并在具有可编程逻辑功能的宇航级 SiP6117S 芯片上进行了验证。仿真实验表明, 相比于前人的设计, 通过调整观测数据的量化精度来降低硬件加速过程中的处理延时, 能有效改善多级流水延迟并在一定程度上提升算法的实时性。

关键词: 伺服控制; FOC; SiP6117S

中图分类号: TN341 文献标识码: A 文章编号: CN11-1780(2021)05-0013-08

DOI: 10.12347/j.ycyk.20210519001

引用格式: 周禹, 刘鸿瑾, 张绍林, 等. 宇航矢量伺服控制器在 SiP 中的设计与实现[J]. 遥测遥控, 2021, 42(5): 113–120.

Design and implementation of aerospace vector servo controller in SiP

ZHOU Yu, LIU Hongjin, ZHANG Shaolin, SHI Bo, WANG Xinxin, LI Bin
(Beijing Sunwise Space Technology Ltd, Beijing 100190, China)

Abstract: FOC vector control algorithm is usually implemented on CPU or DSP for servo driver control. It is difficult to meet the requirements of real-time scene in aerospace applications. In order to improve the real-time performance and reliability of aerospace servo driver system, we implemented the FOC vector control algorithm using hardware acceleration. The design of servo controller is introduced in details, and a hardware acceleration design scheme of full digital and high-performance servo controller is given. It is verified on aerospace grade SiP6117s chip with programmable logic function. The simulation experiments show that the performance of the servo driver controller is tested to verify the feasibility and effectiveness of the design.

Key words: Servo control; FOC; SiP6117S

DOI: 10.12347/j.ycyk.20210519001

Citation: ZHOU Yu, LIU Hongjin, ZHANG Shaolin, et al. Design and implementation of aerospace vector servo controller in SiP[J]. Journal of Telemetry, Tracking and Command, 2021, 42(5): 113–120.

引 言

伴随着计算机控制、电力电子、传感器、电机控制等技术的进步, 伺服驱动技术也得到了迅猛发展, 伺服驱动器的性能获得极大的提高。伺服驱动器在经历了模拟式、模数混合式的发展后, 如今已进入全数字化的发展时期。全数字伺服弥补了模拟式伺服分散性大、零漂、低稳定性等缺点, 发挥了数字控制在控制精度上的优势和控制方法的灵活性^[1], 使伺服驱动器不仅结构简单, 而且性能更加可靠, 其中 FOC 矢量控制算法是伺服驱动控制系统的重要组成部分。目前, 广泛使用的伺服驱动器大多基于 CPU 或 DSP 等微处理器纯软件的方式实现, 受固有代码串行执行特点和微处理器频率限制, 很难减小一次采样控制时间, 大大降低了电机控制的实时性。传统基于 CPU 或 DSP 纯软件控制完成一次电流控制运算需要将近 $20\ \mu\text{s}$ ^[2], 难以满足航天应用中实时性较高场景下的需求。

为了提高控制的实时性与稳定性, 本文从硬件加速的角度考虑, 设计一种全数字的高性能伺服控制器。FPGA 作为现场可编程门阵列, 具有并行处理数据的特性, 可提高控制系统的实时性。SiP6117S 内部集成 1 片 V2 系列 FPGA、1 片低速串行 ADC、2 片高速并行 ADC、1 片高速并行 DAC、1 片刷新芯片 BSV2CQRH 及多颗电容, 产品性能满足宇航级抗辐照要求。

1 FOC 矢量控制原理分析

永磁同步电机的定子为三相对称绕组, 转子上装有永久磁钢。忽略铁芯饱和, 不计涡流和磁滞损耗, 在 d - q 同步旋转坐标系下的数学模型可以表示为以下部分。

磁链方程:

$$\begin{cases} \psi_d = L_d i_d + \psi_r \\ \psi_q = L_q i_q \end{cases} \quad (1)$$

其中, ψ_d 、 ψ_q 为 d 、 q 轴上的定子磁链分量; L_d 、 L_q 为 d 、 q 轴上的等效定子电感; i_d 、 i_q 为 d 、 q 轴上的定子电流分量; ψ_r 为永磁体产生的磁链。

电压方程:

$$\begin{cases} u_d = R_s i_d + p\psi_d - P\omega_m \psi_q \\ u_q = R_s i_q + p\psi_q + P\omega_m \psi_d \end{cases} \quad (2)$$

其中, u_d 、 u_q 为 d 、 q 轴上的定子电压分量; R_s 为定子电阻; p 为微分算子; P 为电机的极对数; ω_m 为电机的机械转速。

电磁转矩方程:

$$T_e = P(\psi_d i_q - \psi_q i_d) \quad (3)$$

其中, T_e 为电机产生的电磁转矩。

机械运动方程:

$$T_e = T_l + p\omega_m + B\omega_m \quad (4)$$

其中, T_l 为负载转矩, B 为与转速成正比的摩擦系数。

矢量控制的基本思想是将交流电动机的物理模型等效地变换为直流电动机的结构, 在同步旋转坐标系下将电流空间矢量分解成正交的励磁分量和转矩分量。定子电流矢量被分解成两个独立的直流分量 i_d 和 i_q , 当永磁体的磁链 ψ_r 和 d 、 q 轴电感 L_d 、 L_q 确定以后, 保持 i_d 为一定值, 控制 i_q 就可以实现对输出电磁转矩的线性化控制, 从而实现对永磁同步电机的控制。

伺服控制器根据用途不同, 其电流矢量控制方法也不一样, 可以采用的控制方法主要有: $i_d=0$ 控制、功率因数控制、恒磁链控制、最大转矩电流控制、弱磁控制和最大输出功率控制。 $i_d=0$ 控制实现了伺服控制器的解耦控制, 最简单也最常用^[3]。FOC 矢量变换通过对外部采集输入的电流值进行 clack 变换, 将 (a,b,c) 坐标系电流值转化为 (α,β) 坐标系的电流值, 然后进行 Park 变换, 将 (α,β) 电流值转换为 (d,q) 坐标系下的电流值。之后进入电流环控制模块, 进行 PI 调节控制, 将电流值转换为电压值。对 (d,q) 坐标系电压值进行 Park 逆变换, 得到 (α,β) 坐标系下的电压值。最后将得到的电压值进行 SVPWM 变换, 输出所在扇区标志信号以及作用时间 T_1 、 T_2 及 T_0 。在 PWM 产生模块输出相应的脉冲。

2 矢量伺服控制器设计

2.1 设计原理

伺服控制器在代码设计上采用模块化、寄存器全同步设计, 模块设计按照组合逻辑和时序逻辑分开描述的方法进行设计。主要实现 Clack 变换、Park 变换、电流环 PI 控制、Park 逆变换、SVPWM 运算控制、Cordic 正余弦计算模块及 PWM 波形产生等功能。其设计功能框图如图 1 所示。

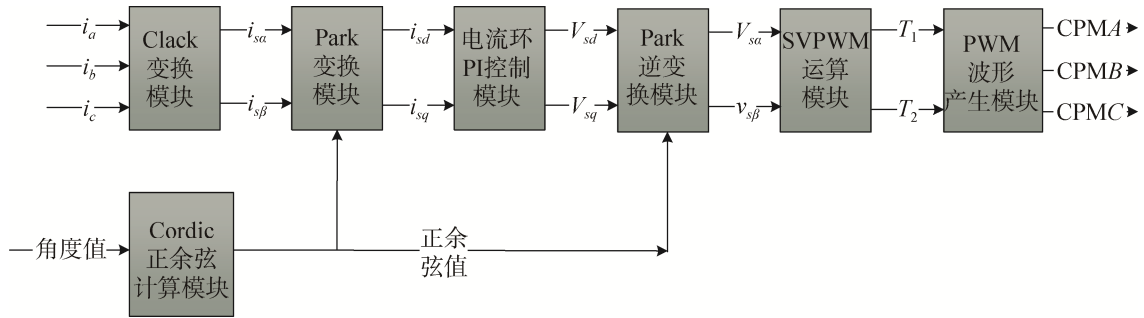


图 1 伺服控制器功能框图

Fig. 1 Function block diagram of servo controller

FOC 矢量运算涉及到乘法、除法、加减、正余弦等操作。FPGA 不能直接对 FOC 矢量算法中的浮点数进行操作,需要采用定标的方法将浮点数转换成定点数,此时的定点数将会是一个位宽特别大的整数。而除法器一般是通过移位等操作来实现的,数据计算的时间与被除数位宽呈线性关系。因此,在不影响功能的前提下,逻辑设计时要对数据的位宽进行适当的截取,通过损失一定的数据精度减少计算的延迟时间。

2.2 模块化设计及实现

2.2.1 Clack 变换模块设计

对采集处理后的电流值进行 Clack 变换,其作用是进行三相坐标系 (a,b,c) 到两相静止坐标系 (α,β) 的转化。

变换方程如下^[4]:

$$i_{s\alpha} = i_a \quad (5)$$

$$i_{s\beta} = \frac{1}{\sqrt{3}}i_a + \frac{2}{\sqrt{3}}i_b \quad (6)$$

$$i_a + i_b + i_c = 0 \quad (7)$$

采用流水线设计,外部输入“clack_start_tick”启动信号,以及电流值 i_a 和 i_b 。当检测到“clack_start_tick”为“1”时,进行状态机的跳转,完成相应的乘法和除法运算。当检测到除法完成标志有效后,输出 (α,β) 坐标系下的电流值 $i_{s\alpha}$ 和 $i_{s\beta}$ 及 Clack 变换完成信号“clack_done”。完成一次 Clack 变换需要 32 个 FPGA 时钟周期。

2.2.2 Cordic 正余弦计算模块设计

该模块用于完成电流环中的正余弦计算。FPGA 采用在线计算的方法,通过 Cordic 算法计算正余弦值,并输出计算完成标志 done 信号。Cordic 算法只能迭代运算角度 Phase 在 $(-99.9^\circ, 99.9^\circ)$ 范围内的正余弦值,为计算 360° 范围内角度的正余弦值,在 FPGA 设计中需要先将电角度 Phase 变换到 $[-90^\circ, 90^\circ]$ 区间,待迭代运算完成后再利用三角恒等式对结果进行转换。采用 16 级流水线设计,输出正余弦值范围在 $(-65535, 65535)$ 之间。当 Cordic 计算结束后,输出完成标志。

2.2.3 Park 变换模块设计

Park 变换完成由两相静止坐标系 (α,β) 到两相旋转坐标系 (d,q) 转化。变换方程如下^[4]:

$$i_{sd} = i_{s\alpha} \cdot \cos(\theta_e) + i_{s\beta} \cdot \sin(\theta_e) \quad (8)$$

$$i_{sq} = -i_{s\alpha} \cdot \sin(\theta_e) + i_{s\beta} \cdot \cos(\theta_e) \quad (9)$$

其中, θ_e 为电机电气角度。

当检测到“clack_done”信号有效后,启动 Park 变换状态机的跳转,完成相应的乘法和移位运算。当完成 Park 变换后,输出 (d,q) 坐标下的电流值 i_{sd} 和 i_{sq} 以及 Park 完成 tick 信号“park_done”。完成一次 Park 变换需要 5 个 FPGA 时钟周期。

2.2.4 电流环 PI 控制模块设计

电流环采用典型 PI 控制律算法。PI 调节器具有算法简单、可靠性高、理论与技术成熟等优点,因此,

被广泛应用于工业控制、机器人控制等领域, 尤其适合具有精确数学模型的系统。离散的 PI 表达式为^[4]:

$$u(n)=u(n-1)+K_p(e(n)-e(n-1))+K_i(e(n)) \quad (10)$$

传递函数如图 2 所示, 积分次数可调。

当检测到“park_done”信号有效后, 启动 PI 调节模块状态机的跳转。对差值进行野值判断、累加求和、逻辑移位、限幅处理等处理, 保证了设计的可靠性。当完成闭环控制后输出完成信号“data_pid_done”。完成一次电流环 PI 控制需要 8 个 FPGA 时钟周期。

2.2.5 Park 逆变换模块设计

Park⁻¹变化的作用是: 两相旋转坐标系(d, q)到两相静止坐标系(α, β)转化。变换方程如下^[4]:

$$V_{sa} = V_{sd} \cdot \cos(\theta_e) - V_{sq} \cdot \sin(\theta_e) \quad (11)$$

$$V_{s\beta} = V_{sd} \cdot \sin(\theta_e) + V_{sq} \cdot \cos(\theta_e) \quad (12)$$

当检测到“pid_done”信号有效后, 启动 Park 逆变换状态机的跳转, 完成相应的乘法和移位运算。当完成 Park 逆变换后输出(α, β)坐标系下的电压值 V_{sa} 和 $V_{s\beta}$ 及 Park 逆变换完成标志“park_rev_done”。完成一次 Park 逆变换需要 5 个 FPGA 时钟周期。

2.2.6 SVPWM 运算模块设计

SVPWM 计算模块主要实现作用扇区的判断及计算作用时间的功能。为了避免死区时间造成电机电流的畸变, SVPWM 实现方式采用混合矢量方式, 共六个扇区, 如图 3 所示。

SVPWM 计算模块当检测到“park_rev_done”信号有效后, 完成信号赋值操作并启动状态机的跳转。根据电压值 U_α 、 U_β 的比例关系进行作用扇区的判断, 并生成对应扇区的 tick 信号(脉冲宽度为一个系统时钟周期), 完成状态机的跳转。详细判断计算过程见表 1。

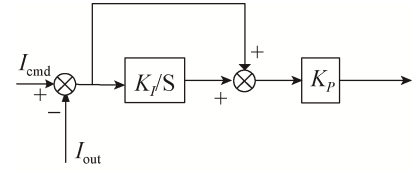


图 2 电流闭环控制传函框图

Fig. 2 Block diagram of current closed loop control

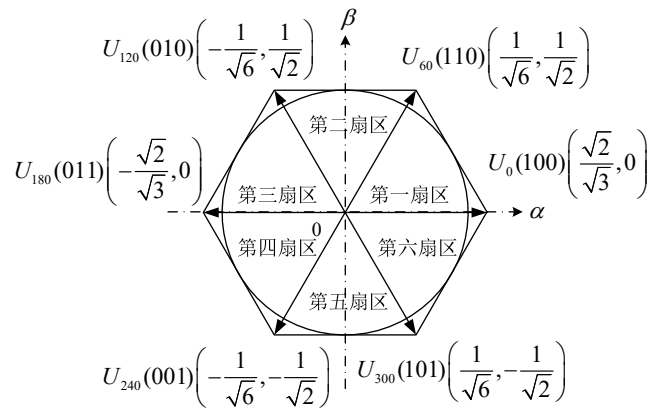


图 3 混合矢量坐标图

Fig. 3 hybrid space vector coordinate diagram

表 1 扇区判断过程

Table 1 Sector judgment process

扇区	反正切值	比值范围
1	$0 < \arctan\left(\frac{U_\alpha}{U_\beta}\right) \leq 60^\circ$	$0 < \frac{U_\alpha}{U_\beta} \leq \sqrt{3}$
2	$60^\circ < \arctan\left(\frac{U_\alpha}{U_\beta}\right) \leq 120^\circ$	$\left \frac{U_\alpha}{U_\beta}\right \geq \sqrt{3}$
3	$120^\circ < \arctan\left(\frac{U_\alpha}{U_\beta}\right) \leq 180^\circ$	$-\sqrt{3} < \frac{U_\alpha}{U_\beta} \leq 0$
4	$180^\circ < \arctan\left(\frac{U_\alpha}{U_\beta}\right) \leq 240^\circ$	$0 < \frac{U_\alpha}{U_\beta} \leq \sqrt{3}$
5	$240^\circ < \arctan\left(\frac{U_\alpha}{U_\beta}\right) \leq 270^\circ$	$\left \frac{U_\alpha}{U_\beta}\right \geq \sqrt{3}$
6	$270^\circ < \arctan\left(\frac{U_\alpha}{U_\beta}\right) \leq 360^\circ$	$-\sqrt{3} < \frac{U_\alpha}{U_\beta} \leq 0$

根据作用矢量分布的扇区计算相应扇区的作用时间 T_1 、 T_2 。计算完成后输出完成标志“svpwm_done”信号。完成一次 SVPWM 变换需要 39 个 FPGA 时钟周期。混合矢量的作用时间推算见表 2。

表 2 混合矢量作用时间表
Table 2 Hybrid space vector action timetable

扇区	作用矢量	U_x 的作用时间 T_1	$U_{x\pm60}$ 作用时间 T_2
1	$U_x = U_0(100), U_{x\pm60} = U_{60}(110)$	$T_1 = \left(\sqrt{\frac{3}{2}}U_\alpha - \frac{1}{\sqrt{2}}U_\beta \right) \frac{T_{\text{PWM}}}{E}$	$T_2 = \left(\sqrt{2}U_\beta \right) \frac{T_{\text{PWM}}}{E}$
2	$U_x = U_{60}(110), U_{x\pm60} = U_{120}(010)$	$T_1 = \left(\sqrt{\frac{3}{2}}U_\alpha + \frac{1}{\sqrt{2}}U_\beta \right) \frac{T_{\text{PWM}}}{E}$	$T_2 = \left(-\sqrt{\frac{3}{2}}U_\alpha + \frac{1}{\sqrt{2}}U_\beta \right) \frac{T_{\text{PWM}}}{E}$
3	$U_x = U_{120}(010), U_{x\pm60} = U_{180}(011)$	$T_1 = \left(\sqrt{2}U_\beta \right) \frac{T_{\text{PWM}}}{E}$	$T_2 = \left(-\sqrt{\frac{3}{2}}U_\alpha - \frac{1}{\sqrt{2}}U_\beta \right) \frac{T_{\text{PWM}}}{E}$
4	$U_x = U_{180}(011), U_{x\pm60} = U_{240}(001)$	$T_1 = \left(-\sqrt{\frac{3}{2}}U_\alpha + \frac{1}{\sqrt{2}}U_\beta \right) \frac{T_{\text{PWM}}}{E}$	$T_2 = \left(-\sqrt{2}U_\beta \right) \frac{T_{\text{PWM}}}{E}$
5	$U_x = U_{240}(001), U_{x\pm60} = U_{300}(101)$	$T_1 = \left(-\sqrt{\frac{3}{2}}U_\alpha - \frac{1}{\sqrt{2}}U_\beta \right) \frac{T_{\text{PWM}}}{E}$	$T_2 = \left(\sqrt{\frac{3}{2}}U_\alpha - \frac{1}{\sqrt{2}}U_\beta \right) \frac{T_{\text{PWM}}}{E}$
6	$U_x = U_{300}(101), U_{x\pm60} = U_0(100)$	$T_1 = \left(-\sqrt{2}U_\beta \right) \frac{T_{\text{PWM}}}{E}$	$T_2 = \left(\sqrt{\frac{3}{2}}U_\alpha + \frac{1}{\sqrt{2}}U_\beta \right) \frac{T_{\text{PWM}}}{E}$

2.2.7 PWM 波形产生模块设计

根据 SVPWM 计算得到的作用时间 T_1 、 T_2 及换向策略表产生相应的脉冲。电机驱动芯片内部集成了三相 MOSFET 全桥驱动电路。其中, $\overline{AH}$ 、 $\overline{AL}$ 、 $\overline{BH}$ 、 $\overline{BL}$ 、 $\overline{CH}$ 、 $\overline{CL}$ 分别是左、中、右三个桥臂的开关管控制信号,上桥臂控制信号低电平有效,下桥臂控制信号高电平有效。内部集成的六个 MOSFET 开关状态受 $\overline{AH}$ 、 $\overline{AL}$ 、 $\overline{BH}$ 、 $\overline{BL}$ 、 $\overline{CH}$ 、 $\overline{CL}$ 控制,高电平时,所有 MOSFET 均关断。死区时间可通过内部逻辑设置。死区时间典型值为 5 μs , 极限值为 3 μs ~7 μs 。换向策略表见表 3、表 4。

表 3 扇区 1、3、5 换向策略表
Table 3 Sector 1,3,5 reversing strategy tables

扇区	段 1 ($T_0/4$)	段 2 ($T_1/2$)	段 3 ($T_2/2$)	段 4 ($T_0/2$)	段 5 ($T_2/2$)	段 6 ($T_1/2$)	段 7 ($T_0/4$)
1	$U_{\text{zero}}(000)$	$U_x = U_0(100)$	$U_{x\pm60} = U_{60}(110)$	$U_{\text{zero}}(111)$	$U_{x\pm60} = U_{60}(110)$	$U_x = U_0(100)$	$U_{\text{zero}}(000)$
3	$U_{\text{zero}}(000)$	$U_x = U_{120}(010)$	$U_{x\pm60} = U_{180}(011)$	$U_{\text{zero}}(111)$	$U_{x\pm60} = U_{180}(011)$	$U_x = U_{120}(010)$	$U_{\text{zero}}(000)$
5	$U_{\text{zero}}(000)$	$U_x = U_{240}(001)$	$U_x = U_{240}(001)$	$U_{\text{zero}}(111)$	$U_{x\pm60} = U_{300}(101)$	$U_x = U_{240}(001)$	$U_{\text{zero}}(000)$

表 4 扇区 2、4、6 换向策略表
Table 4 Sector 2,4,6 reversing strategy tables

扇区	段 1 ($T_0/4$)	段 2 ($T_1/2$)	段 3 ($T_2/2$)	段 4 ($T_0/2$)	段 5 ($T_2/2$)	段 6 ($T_1/2$)	段 7 ($T_0/4$)
2	$U_{\text{zero}}(000)$	$U_{x\pm60} = U_{120}(010)$	$U_x = U_{60}(110)$	$U_{\text{zero}}(111)$	$U_x = U_{60}(110)$	$U_{x\pm60} = U_{120}(010)$	$U_{\text{zero}}(000)$
4	$U_{\text{zero}}(000)$	$U_{x\pm60} = U_{240}(001)$	$U_x = U_{180}(011)$	$U_{\text{zero}}(111)$	$U_x = U_{180}(011)$	$U_{x\pm60} = U_{240}(001)$	$U_{\text{zero}}(000)$
6	$U_{\text{zero}}(000)$	$U_{x\pm60} = U_{360}(100)$	$U_x = U_{300}(101)$	$U_{\text{zero}}(111)$	$U_x = U_{300}(101)$	$U_{x\pm60} = U_{360}(100)$	$U_{\text{zero}}(000)$

SVPWM 的周期波形图一般采用对称的形式,这样产生的谐波最小。图 4 是混合矢量 SVPWM 发生器的一个扇区的波形图。

从图 4 中可以看出, SVPWM 的每个周期中都包含两个相邻矢量和一些零矢量。这些矢量在 T_{PWM} 周期中的分布策略有很多,可以是对称或非对称的,采用 7 段式换向方式。SVPWM 计算模块在每个周期都重新计算每个矢量的作用时间,按照比例设定相应的比较器值。其中, $T_{\text{PWM}} = 0.02 \text{ ms} = T_0 + T_1 + T_2$,

T_0 由 FPGA 计算得到, E 为电压值, $E=5\text{ V}$ 。当 $T_1=T_2=0$ 时, TPWM 周期内全部输出零矢量 (0,0,0)。 T_1 、 T_2 溢出保护: 当 $T_1+T_2>T_{\text{PWM}}$, FPGA 输出零矢量 (0,0,0), 以保护转台产品。

3 仿真实现

仿真条件: FPGA 工作频率为 60 MHz, 电流采样频率为 10 kHz, “clock_start_tick” 信号周期为 100 μs , 仿真结果如图 5 所示。

在电流环 PI 控制模块中, 当检测到 “park_done” 信号有效后, 启动 PI 调节模块状态机的跳转。当完成闭环控制后输出完成 tick 信号 “data_pid_done”。完成一次电流环 PI 控制需要 8 个 FPGA 时钟周期。仿真结果如图 6 所示。

当检测到 “clock_start_tick” 有效后启动 Clack 变换, 直至 SVPWM 变换结束输出完成标志 “svpwm_done”。中间计算的时间为 1.48 μs , 仿真结果如图 7 所示。

判断所处扇区位置, 根据扇区位置有效标志 “sector_tick” 及计算得到的 T_1 “time_t1_temp”、 T_2 “time_t2_temp”、死区时间 “dead_time” 产生相应的 “pwm_ah”、“pwm_al”、“pwm_bh”、“pwm_bl”、“pwm_ch”、“pwm_cl” 信号。仿真结果如图 8 所示。

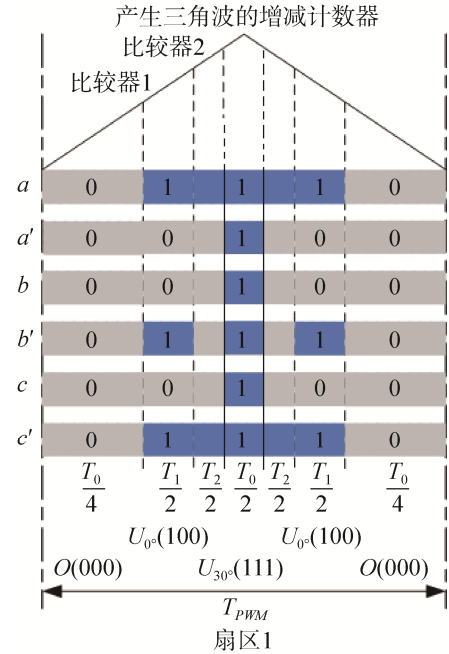


图 4 扇区 1 波形图
Fig. 4 Sector 1 waveform

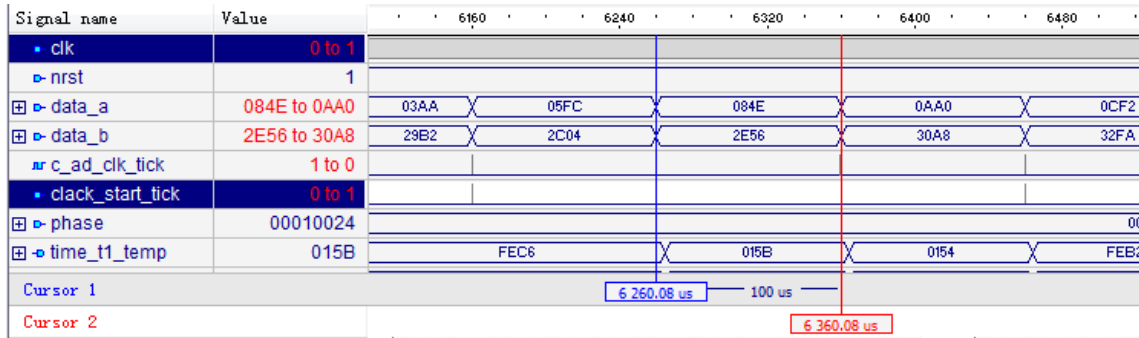


图 5 电流采样周期波形图
Fig. 5 Current sampling period waveform

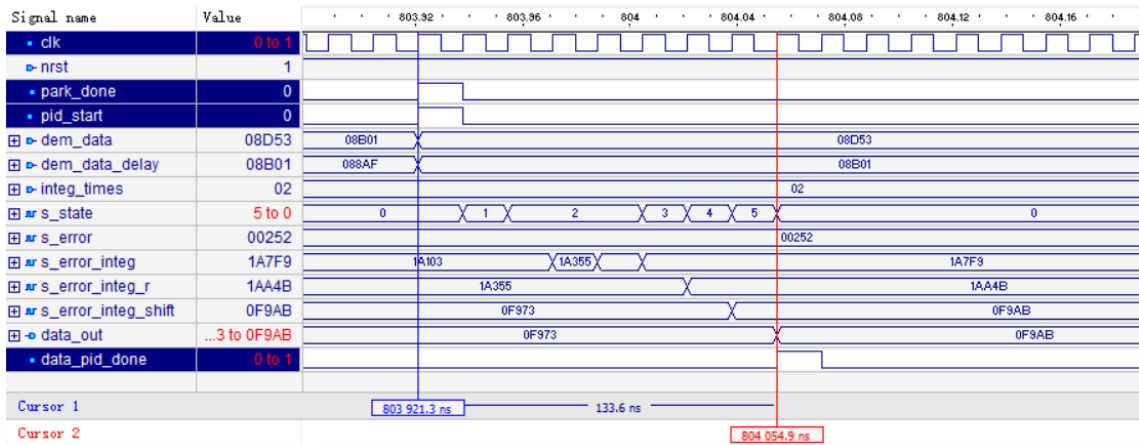


图 6 电流环控制模块仿真波形图
Fig. 6 Current loop control module simulation waveform

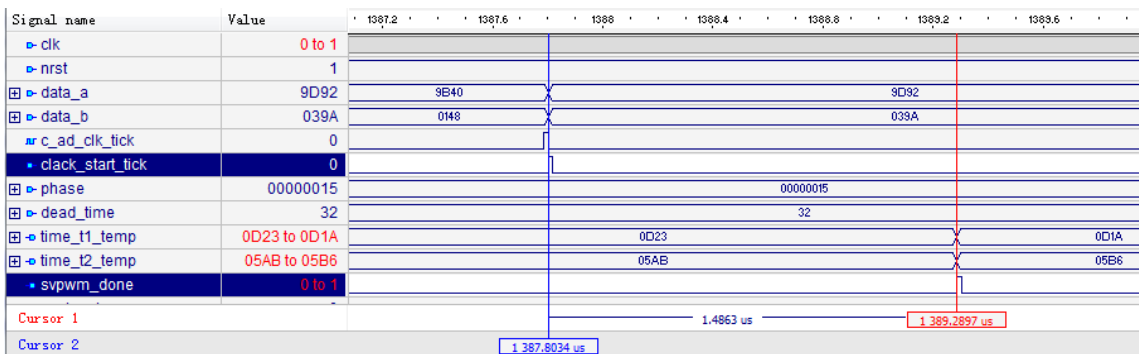


图 7 FOC 矢量计算时间波形图
Fig. 7 FOC vector calculation time waveform

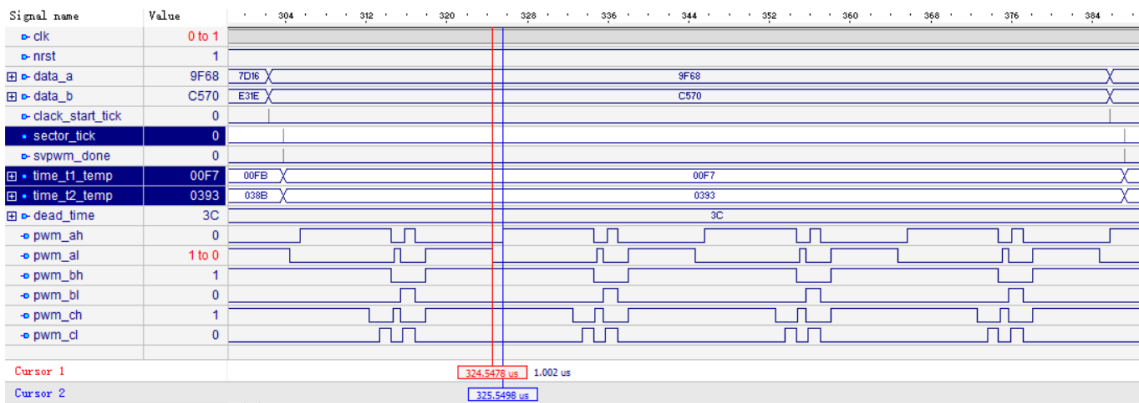


图 8 PWM 波形图
Fig. 8 PWM waveform

仿真分析可知,当 FPGA 工作频率为 60 MHz、电流环采样频率为 10 kHz、 $T_{\text{PWM}} = 0.02 \text{ ms}$ 、死区时间设为 $1 \mu\text{s}$ 时,从启动电流采样到完成计算产生 PWM 需要 $1.48 \mu\text{s}$,即需要 89 个 FPGA 系统时钟。与文献[5]中比较,相同条件下缩短了约 $0.5 \mu\text{s}$,相当于 30 个时钟周期的性能提升,减小了 FOC 矢量数字控制的延时,提高了系统的响应时间和带宽,使得系统可以采用更高开关频率的功率器件来提高系统响应性能。为了保持波形的完整性,在设计时考虑只有完成当前周期波形完整输出才会响应新的作用时间 T_1 、 T_2 。上电后可通过 SiP6117S 实时控制电流采样与检测,可实现对电机的实时控制。

4 结束语

伺服控制器采用全同步化、流水线设计,具有可靠性高、稳定性强等优点。通过对控制矢量算法的硬件加速,可实现对电机系统的实时控制。可在 89 个 FPGA 系统时钟周期内完成 FOC 矢量控制算法,从而实现快速响应的伺服控制系统,可以满足航天应用中实时性较高的场景。实验表明,该矢量伺服控制器电流响应时间可以达到 μs 级,其系统误差满足工程要求,具有良好的动态性能。

参考文献

- [1] 王苏洲,舒志兵,李照. 基于 STM32+CPLD 全数字同步伺服驱动器的设计与实现[J].电机与控制应用, 2017, 44(4): 74–79, 100.
WANG Suzhou, SHU Zhibing, LI Zhao. Design and implementation of digital synchronous servo driver based on STM32 and CPLD[J]. Electric Machines & Control Application, 2017, 44(4): 74–79, 100.
- [2] 陈平,曾岳南,杨志平. 基于 FPGA 的永磁同步电机控制器设计[J]. 电子设计工程, 2009, 17(12): 82–84.
CHEN Ping, ZENG Yuenan, YANG Zhiping. Design of PMSM controller based on FPGA[J]. Electronic Design Engineering, 2009, 17(12): 82–84.

- [3] 陈进, 王冠凌, 娄珂. 基于 SVPWM 的永磁同步电机控制[J]. 自动化与仪器仪表, 2009(3): 51–52, 107.
CHEN Jin, WANG Guanling, LOU Ke. The control of PMSM based on SVPWM[J]. Automation & Instrumentation, 2009(3): 51–52, 107.
- [4] 颜景斌, 刘嘉辉, 谢金宝, 等. 矢量控制速度伺服系统的 FPGA 实现[J]. 电机与控制学报, 2007, 11(1): 17–20.
YAN Jingbin, LIU Jiahui, XIE Jinbao, et al. FPGA implementation of the vector-controlled speed servo system[J]. Electric Machines and Control, 2007, 11(1): 17–20.
- [5] 伍庆, 基于 FPGA 的交流伺服系统电流环设计[D]. 武汉: 华中科技大学, 2013.

[作者简介]

- 周 禹 1990 年生, 硕士, 工程师, 主要研究方向为 FPGA 软件设计、数字集成电路设计及验证方法。
- 刘鸿瑾 1980 年生, 博士, 研究员, 主要研究方向为大规模集成电路设计、宇航高性能处理器设计。
- 张绍林 1988 年生, 硕士, 高级工程师, 主要研究方向高可靠处理器系统架构设计。
- 施 博 1987 年生, 博士, 高级工程师, 主要研究方向为信号处理、嵌入式微处理器系统设计。
- 王鑫鑫 1992 年生, 硕士, 工程师, 主要研究方向为数字集成电路设计及验证方法。
- 李 宾 1985 年生, 博士, 高级工程师, 主要研究方向为宇航级微系统设计。

(本文编辑: 傅 杰)