A Design Method for Low-cost and SOPC-based Flexible Lifting Control System

Dechun Zheng*, Li Xu, Yongping Zhang, Guojun Li
School of Electronic and Information Engineering, Ningbo University of Technology, China
**Corresponding Author.*

Abstract

To efficiently reduce the development and production costs of the intelligent lifting control system, we introduce a design method for the intelligent lifting system with client-server architecture. We replace DSP processor core or DSP (Digital Signal processor) core with Nois II soft-core processor so that the design and production costs can be effectively cut. By replacing DSP processor or DSP processor core with Nois II soft-core processor, the design and production costs can be significantly reduced. In our design, loop vector control units work as a server processor, and a central computing unit with four independent multipliers and two adders is employed, with the implementation method based on a state machine. The experimental results prove effective in reducing resource requirements for FPGA (Field Programmable Gate Array), show that the proposed method can be successfully applied to the implementation of a complete intelligent flexible lifting control system on a low-end Altera Cyclone FPGA, and servo motor control achieves better dynamic performance.

Keywords: *Flexible Lifting, Nois II, System on Chip (SoC), Servo Control, FPGA.*

I. Introduction

The primary function of intelligent flexible lifting system is to deliver swift, accurate movement by tracking operating force or the pressing of keys, and achieve servo tracking to effectively reduce the labor intensity and improve work efficiency. At present, there are three ways to realize the intelligent flexible lifting system: DSP processor implementation, FPGA-based ASIC implementation and FPGA-based SoC implementation. In the case of DSP processor, its real-time performance is mediocre because of its weak parallel computing ability. Even in the multi-processor scheme, its cost may exceed its benefits due to the complexity of its algorithm. In the case of FPGA-based ASIC solution, the system can carry out parallel operation at a fast speed, but the user interface and system upgradability can be compromised. As for FPGA-based SoC solution, both the speedy responsiveness and cost-effectiveness can be ensured[1-5].

Servo motor has been widely used on an industrial scale， thanks to its low upkeep, high performance and durability, so the key to the development of intelligent flexible lifting system is to adopt a reliable control technology and method to improve the synchronization and accuracy of intelligent flexible lifting system. As the dynamic model of servo motor is generally coupled, multivariable and highly nonlinear, it is difficult to obtain ideal dynamic performance. In order to solve the abovementioned problems, many control methods are proposed for servo motor, such as dynamic surface control, Hamiltonian control, sliding mode control, back-stepping control, fuzzy logic control and so on[6-10]. Although these methods will improve the control performance of servo motor in theory, they are all designed for servo motor driver system operating on a continuous basis. For the purpose of reducing the computational pressure of DSP, some scholars use FPGA to realize some necessary links in the motor control system including photoelectric pulse encoder interface module, A/D sampling[11] interface module, PWM (Pulse Width Modulation) pulse modulation module[12]. Others realize the system on chip based on PI (Proportional-Integral) algorithm or motor control IP special core, and preliminarily achieve the system on chip based on FPGA[13-15]. However, all these solutions are solely aimed at controlling the servo motor without considering the design and production cost.

This paper presents a design method of intelligent flexible lifting control system based on master-slave architecture. In order to save cost, Nois II soft core and current loop vector control unit are used to construct SoC intelligent flexible lifting control. The use of Nois II processor can not only cut the processor and add the required IP (intellectual property) module according to the system needs, but also save the cost of purchasing processor or processor core. When designing the current loop vector control unit, we design a core operation unit comprising four independent multipliers and two adders by analysing and normalising the algorithm, and then the whole vector control algorithm is completed by time division multiplexing of the core operation unit of the timing state. The result show that the operation time of the current loop vector control is greatly reduced, and the FPGA resources are effectively saved. In the second part of the paper, the architecture of SOPC-based (System On Programmable Chip) flexible lifting control system is given. The third part focus on the design of current loop vector control operation unit. The fourth part reveals the experimental platform nuclear test results. The fifth part is the discussion and further research we will conduct.

II. Architecture of SOPC-based flexible lifting control system

The main function of the intelligent flexible lifting system is to reduce the labour intensity of porters or assemblers and perform quickly and accurately according to the touch force or others applied on the object. Therefore, the controller of the intelligent flexible lifting system is able to accurately measure the applied force as well as quickly and accurately control the motor. To make the controller meet the needs of industrial production, this method can reduce the design and production costs as much as possible on the premise of ensuring reliability. Based on these considerations, a control scheme of intelligent flexible lifting system is proposed by using Nois II processor and current loop vector control unit, as is shown in Figure 1. By replacing DSP processor with Nois II processor, both the costs and space can be saved; In the design of current loop vector control unit, the resource requirements for FPGA core multipliers can be effectively reduced through algorithm analysis and normalization and the use of a state machine, which makes it possible to realize the system on FPGA at low costs. Structurally, the entire system resembles a dual-core process system with a Master-Slave structure, where the Nois II soft processor works as a Master system, and spatial vector control module works as a slave system that can work at a high speed, and the on-chip control system and various peripherals are mounted on the Nois main processor through the high-speed Avalon. Nois II is responsible for real-time control for slower-paced outer loop (speed loop, position loop), making connection to upper machines, and real-time monitoring of the entire system. It is also responsible for the detection and signal processing of applied force, human-computer interface and communication and control parameter initialization with current loop vector control unit which is mainly realised by answering for the realisation of current loop vector control algorithm.

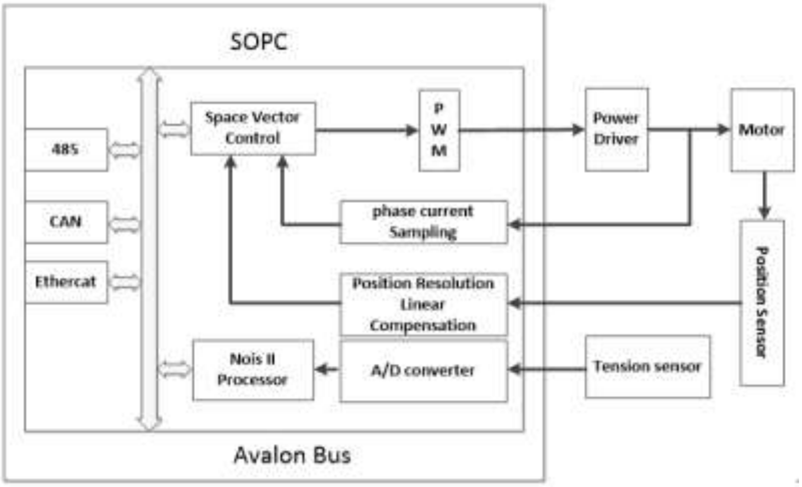


Fig. 1: Architecture diagram of SOPC-based flexible lifting control system

III. Design of arithmetic unit for 3-current loop vector control

The commonly used servo motor control system is shown in Figure 2, in which the control algorithm implemented for outer loop (speed loop and position loop) control algorithm is the same as other processors, while the vector control part is partially or completely implemented on FPGA. Notation n in the graph represents the rotation angle and rotation speed for motor respectively; PI is a current loop regulator. The current vector control unit first collects the phase current, and then decouples the flux linkage through Clarke and Park transform. The excitation component of stator current is controlled to zero or close to zero, so the motor torque can be controlled linearly. In addition, the current vector control algorithm is composed of Clarke transform module, Park transform module, PI regulator module, Park inverse transform module and SVPWM module. Therefore, in the implementation of SoC, the traditional method is to use sequential modularisation or pipelining, which requires more multiplication and accumulation units and FPGA logic gate resources.

3.1 Calculation steps of current loop vector control

In practice, the control of current loop vector consists of seven steps, including Clarke transformation module calculation, Park transformation module calculation, PI controller module calculation, Park inverse transform, SVPWM (Space Vector Pulse Width Modulation) space sector calculation, SVPWM theoretical switch calculation and SVPWM actual pulse width modulation calculation. The operation unit of each step is equivalent to completing the multiplication and accumulation as shown in formula (1).

$$\begin{cases} y_1 = c_1 \times x_1 + c_2 \times x_2 \\ y_2 = c_3 \times x_3 + c_4 \times x_4 \end{cases} \tag{1}$$

The specific calculation steps of current loop vector control are as follows:

1) Clarke transformation

The parameter matrix C and input variables in equation (1) are

$$C = \begin{pmatrix} c_1 & c_2 \\ c_3 & c_4 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \dfrac{1}{\sqrt{3}} & \dfrac{2}{\sqrt{3}} \end{pmatrix}, \quad \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} i_a \\ i_b \end{pmatrix} \tag{2}$$

Its output is $\begin{pmatrix} i_{s\alpha} \\ i_{s\beta} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$.Where i_a and i_b are the current flowing through the U and V phases of the motor respectively.

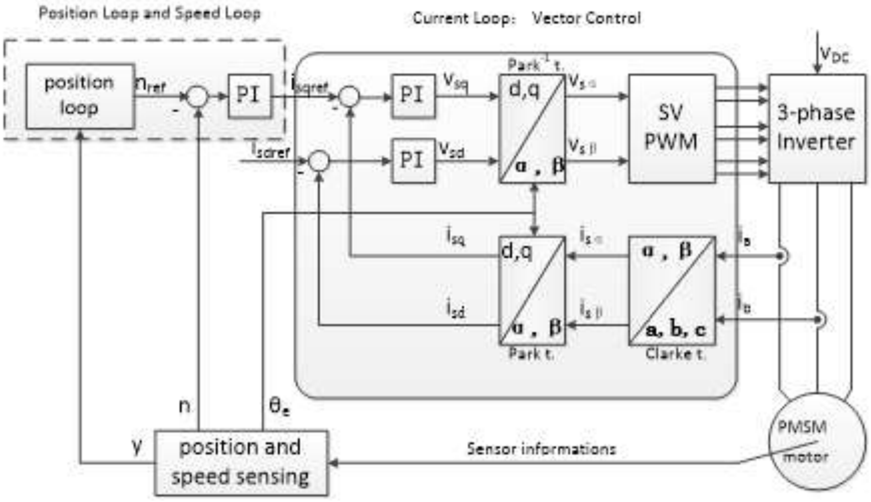


Fig. 2: Block Diagram of Servo Motion Control System

2) Park Transform

The parameter matrix C and input variables in equation (1) are

$$C=\begin{pmatrix} c_1 & c_2 \\ c_3 & c_4 \end{pmatrix}=\begin{pmatrix} \cos\theta_e & \sin\theta_e \\ -\sin\theta_e & \cos\theta_e \end{pmatrix},\ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}=\begin{pmatrix} x_3 \\ x_4 \end{pmatrix}=\begin{pmatrix} i_{s\alpha} \\ i_{s\beta} \end{pmatrix} \tag{3}$$

Its output is $\begin{pmatrix} i_{sd} \\ i_{sq} \end{pmatrix}=\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$.Where θ_e is rotation angle of the motor.

3) Calculating the difference of PI regulator(using incremental implementation)

The parameter matrix C and input variables in equation (1) are

$$C=\begin{pmatrix} c_1 & c_2 \\ c_3 & c_4 \end{pmatrix}=\begin{pmatrix} K_{d\_new} & K_{d\_old} \\ K_{q\_new} & K_{q\_old} \end{pmatrix},\ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}=\begin{pmatrix} e_d(k) \\ e_d(k-1) \end{pmatrix},\ \begin{pmatrix} x_3 \\ x_4 \end{pmatrix}=\begin{pmatrix} e_q(k) \\ e_q(k-1) \end{pmatrix} \tag{4}$$

Its output is $\begin{pmatrix} V_{sd}(k) \\ V_{sq}(k) \end{pmatrix}=\begin{pmatrix} V_{sd}(k-1)+y_1 \\ V_{sq}(k-1)+y_2 \end{pmatrix}$.Where K_{d_new}, K_{d_old}, K_{q_new}, K_{q_old} is decided by the parameters of the PI regulator. V_{sd} (k), V_{sq} (k), e_d(k), e_q(k) is beat value of current control beat. V_{sd}(k-1), V_{sq} (k-1), e_d(k-1), e_q(k-1) is beat value of last control beat.

4) Calculating the inverse Park transformation

$$C=\begin{pmatrix} c_1 & c_2 \\ c_3 & c_4 \end{pmatrix}=\begin{pmatrix} \cos\theta_e & -\sin\theta_e \\ \sin\theta_e & \cos\theta_e \end{pmatrix},\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}=\begin{pmatrix} x_3 \\ x_4 \end{pmatrix}=\begin{pmatrix} V_{sd} \\ V_{sq} \end{pmatrix} \tag{5}$$

Its output is $\begin{pmatrix} i_{s\alpha} \\ i_{s\beta} \end{pmatrix}=\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$.

5) Calculating the space vector pulse width modulation (SVPWM) of the space sector

The parameter matrix C and input variables in equation (1) are

$$C=\begin{pmatrix} c_1 & c_2 \\ c_3 & c_4 \end{pmatrix}=\begin{pmatrix} \sqrt{3} & -1 \\ -\sqrt{3} & -1 \end{pmatrix},\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}=\begin{pmatrix} x_3 \\ x_4 \end{pmatrix}=\begin{pmatrix} V_{S\alpha} \\ V_{S\beta} \end{pmatrix}$$

$$\mathrm{Sec}t\_No=A+2B+4C \quad A=\begin{cases} 1 & V_a>0 \\ 0 & V_a<0 \end{cases} \quad B=\begin{cases} 1 & V_b>0 \\ 0 & V_b<0 \end{cases} \quad C=\begin{cases} 1 & V_c>0 \\ 0 & V_c<0 \end{cases} \tag{6}$$

Its output is $\begin{pmatrix} V_a \\ V_b \\ V_c \end{pmatrix}=\begin{pmatrix} V_{s\beta} \\ y_1 \\ y_2 \end{pmatrix}$.

Logical judgment and Sect_No is implemented in additional logic units of SoC.

6) Calculating the theoretical switching time of each phase of SVPWM

First, calculating the intermediate variables (x, y, z) from equation (1), and the corresponding parameter matrix C and input variables are

$$C=\begin{pmatrix} c_1 & c_2 \\ c_3 & c_4 \end{pmatrix}=\begin{pmatrix} \frac{\sqrt{3}}{2}V_{DC\_init} & \frac{3}{2}V_{DC\_init} \\ \frac{\sqrt{3}}{2}V_{DC\_init} & -\frac{3}{2}V_{DC\_init} \end{pmatrix},\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}=\begin{pmatrix} x_3 \\ x_4 \end{pmatrix}=\begin{pmatrix} V_{S\beta} \\ V_{S\alpha} \end{pmatrix}$$

$$V_{DC\_init}=\frac{PWMPRD}{V_{dc}}$$

$$t_1=\begin{cases} Z & \mathrm{Sec}t\_No==1 \\ Y & \mathrm{Sec}t\_No==2 \\ -Z & \mathrm{Sec}t\_No==3 \\ -X & \mathrm{Sec}t\_No==4 \\ X & \mathrm{Sec}t\_No==5 \\ -Y & \mathrm{Sec}t\_No==6 \end{cases} \quad t_2=\begin{cases} Y & \mathrm{Sec}t\_No==1 \\ -X & \mathrm{Sec}t\_No==2 \\ X & \mathrm{Sec}t\_No==3 \\ Z & \mathrm{Sec}t\_No==4 \\ -Y & \mathrm{Sec}t\_No==5 \\ -Z & \mathrm{Sec}t\_No==6 \end{cases} \tag{7}$$

Its output is $\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}=\begin{pmatrix} y_1+y_2 \\ y_1 \\ y_2 \end{pmatrix}$.Where PWMPRD is period of PWM, V_{dc} is bus voltage.

Then, the intermediate variables (X,Y,Z) determine the switching time t_1 and t_2 of the sector through logical table lookup.

7) Calculating the actual pulse width modulation time of each phase of SVPWM

First, the switching time is saturated by equation (1), and the corresponding, parameter matrix C and input variables are

$$C=\begin{pmatrix} c_1 & c_2 \\ c_3 & c_4 \end{pmatrix}=\begin{pmatrix} \frac{PWMPRD}{t_1+t_2} & 0 \\ \frac{PWMPRD}{t_1+t_2} & 0 \end{pmatrix},\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}=\begin{pmatrix} t_1 \\ 0 \end{pmatrix},\begin{pmatrix} x_3 \\ x_4 \end{pmatrix}=\begin{pmatrix} t_2 \\ 0 \end{pmatrix}$$

$$t_{1s}=\begin{cases} t_1 & t_1+t_2\le PWMPRD \\ t_{1SAT} & t_1+t_2>PWMPRD \end{cases} \quad t_{2s}=\begin{cases} t_2 & t_1+t_2\le PWMPRD \\ t_{2SAT} & t_1+t_2>PWMPRD \end{cases}$$

$$t_{aon}=\frac{PWMPRD-t_{1s}-t_{2s}}{2} \quad t_{bon}=t_{aon}+t_{1s} \quad t_{con}=t_{bon}+t_{2s} \tag{8}$$

Its output is $\begin{pmatrix} t_{1SAT} \\ t_{2SAT} \end{pmatrix}=\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$.

Among them, $\frac{PWMPRD}{t_1+t_2}$ is operated by independent dividers, and then the intermediate variable t is calculated by independent units t_{aon} 、 t_{bon} 、 t_{con} and logical look-up table to determine T_a 、 T_b 、 T_c. The specific work is shown in Table 1.

To sum up, the state machine method can be used to design the core operation unit to reduce the use of multiplication accumulation unit and gate resource consumption on FPGA.

Table 1 Modulation time of each phase in different sectors

Sect_on	1	2	3	4	5	6
CMPA	t_{bon}	t_{aon}	t_{aon}	t_{con}	t_{con}	t_{bon}
CMPB	t_{aon}	t_{con}	t_{bon}	t_{bon}	t_{aon}	t_{con}
CMPC	t_{con}	t_{bon}	t_{con}	t_{aon}	t_{bon}	t_{aon}

3.2 Design of core computing unit based on state machine

The vector control core unit of current loop vector control, with its design based on the state machine method, is shown in Figure 3. Under the control of the state machine, the core operation unit completes the vector control calculation of the above seven steps in one beat. The time sequence state value S[2. . 0] of the state machine is shown in Figure 4. The state 0 is used for the conversion of motor rotor angle to electric angle and the compensation of magnetic declination. Under the control of clock ticks, the state value S[2. . 0] corresponding to the seven steps of vector control and the latch signal used to store the intermediate variables are generated. The state quantity is used to control the selector to select the input corresponding to the current state to the fast basic operation unit. After the vector calculation of the current step is completed, after the latch signal drops, the corresponding register is selected as the intermediate variable for the next beat along the latch of the current calculation result.

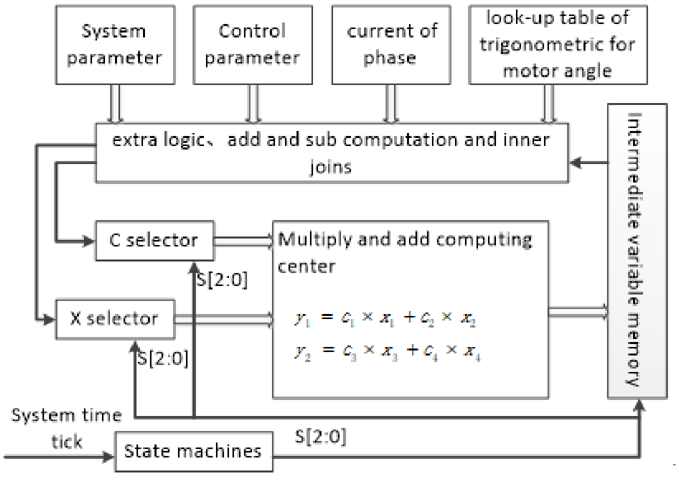

System parameter
Control parameter
current of phase
look-up table of trigonometric for motor angle
extra logic、 add and sub computation and inner joins
Intermediate variable memory
C selector
Multiply and add computing center
$y_1 = c_1 \times x_1 + c_2 \times x_2$
$y_2 = c_3 \times x_3 + c_4 \times x_4$
S[2:0]
X selector
S[2:0]
System time tick
State machines
S[2:0]

Fig. 3: block diagram of vector control core unit

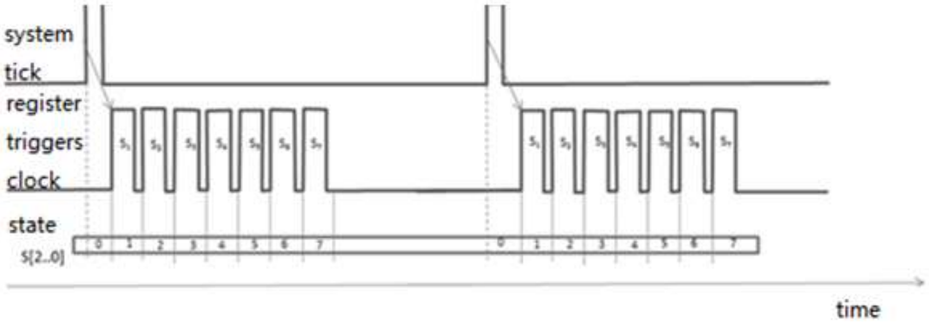


Fig. 4: Timing state of vector control core unit

IIII. Experimental Results

The intelligent flexible lifting system designed by this method is shown in Figure 5. In operation, the maximum lifting capacity of the device is 100kg; The maximum lifting speed (no-load) is 55m/min; The maximum lifting speed (full load) is 28m/min; The maximum lifting speed (suspension working mode) is 25m/min, and the measurement accuracy reaches ±1%.

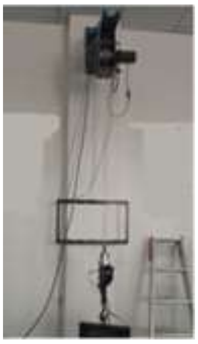
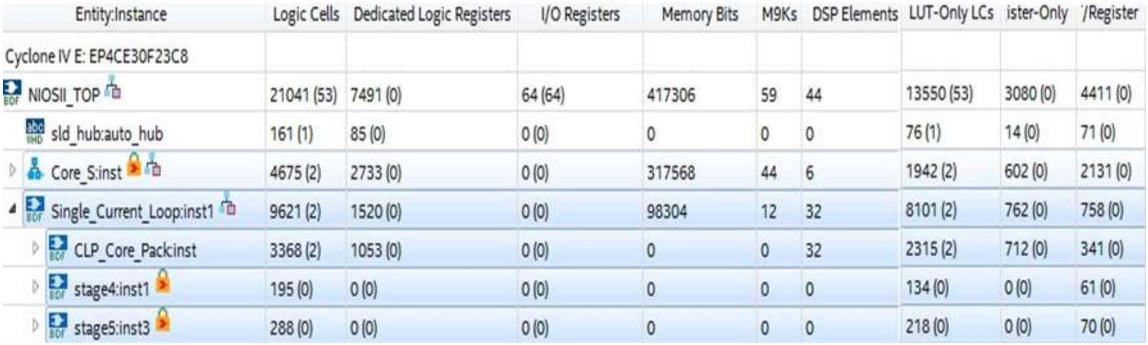


Fig. 5: intelligent flexible lifting prototype system

The slave control unit takes the servo motor as the control object; The phase resistance of the servo motor is 3.4 Ω; The phase inductance is 10.8Mh; The maximum power is 850W, and the feedback encoder of the motor adopts 2500 line incremental encoder. The whole servo control system (Nois II processor plus slave control unit) consumes 21041 logic gates and 417306-bit memory cells, among which the current vector control unit consumes 9621 logic gates, as shown in Figure 6.



Entity:Instance	Logic Cells	Dedicated Logic Registers	I/O Registers	Memory Bits	M9Ks	DSP Elements	LUT-Only LCs	ister-Only	/Register
Cyclone IV E: EP4CE30F23C8									
NIOSII_TOP	21041 (53)	7491 (0)	64 (64)	417306	59	44	13550 (53)	3080 (0)	4411 (0)
sld_hub:auto_hub	161 (1)	85 (0)	0 (0)	0	0	0	76 (1)	14 (0)	71 (0)
Core_S:inst	4675 (2)	2733 (0)	0 (0)	317568	44	6	1942 (2)	602 (0)	2131 (0)
Single_Current_Loop:inst1	9621 (2)	1520 (0)	0 (0)	98304	12	32	8101 (2)	762 (0)	758 (0)
CLP_Core_Pack:inst	3368 (2)	1053 (0)	0 (0)	0	0	32	2315 (2)	712 (0)	341 (0)
stage4:inst1	195 (0)	0 (0)	0 (0)	0	0	0	134 (0)	0 (0)	61 (0)
stage5:inst3	288 (0)	0 (0)	0 (0)	0	0	0	218 (0)	0 (0)	70 (0)

Fig. 6: SoC servo control consumption of resources

When the current loop vector control unit is in operation, the procedure of running an entire current loop vector control is completed by performing state 1 to state 7. It is showed in Figure 7 that working time is represented by upper part in signal 1, which has a scaling of 500ns. Therefore, an entire current loop vector control is completed within 1.6us, which is significantly less than 5us. The red signal is the transition signal of the state machine, where state 0 starts with the red falling edge, and state 7 starts with the last red rising edge.

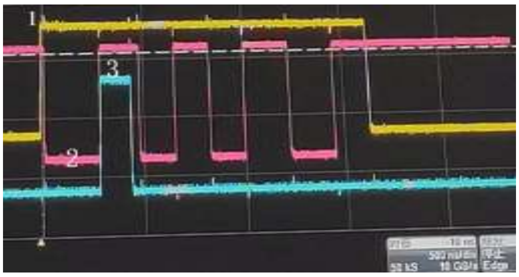


Fig. 7: Actual sequence diagram of current loop vector control

Data in Figure 8 to 12 is gathered from real-time reciprocation with servo motor. Figure 8 and 9 show the tracking performance of current under control of directional magnetic field on d-axis and q-axis. It can be inferred from figures that Id_cal is close to 0 after real-time current measurement and position decoupling. Additionally, Iq_cal follows the referenced current value Iq_ref, which indicates that the system has good linearity after magnetic decoupling, and that vector control has outstanding dynamic performance. The position and speed tracking curve showed in Figure 10 and 11 respectively demonstrate that servo motion control has good dynamic performance and repeated positioning accuracy. Figure 12 shows the real-time collected U and V value for phase current when accelerating/decelerating, which proves that SVPWM in vector controller has flexible commutation ability and digital filtering current noise in current sampling has a good ability to suppress.

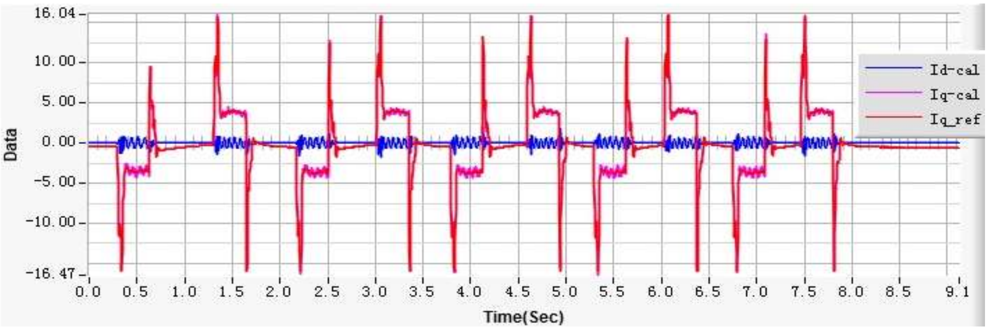


Fig. 8: i_d, i_q trace figure

ISSN: 0010-8189

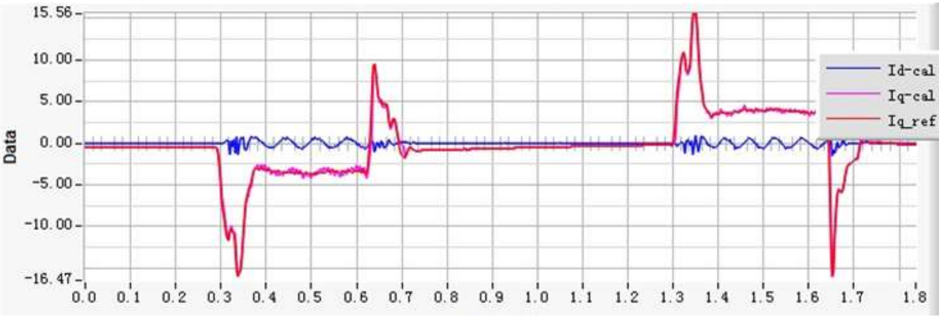


Fig. 9: i_d，i_q figure of current amplification

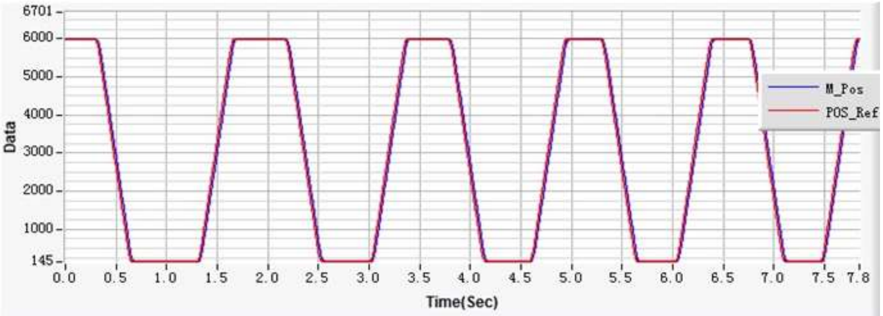


Fig. 10: Figure of position trace

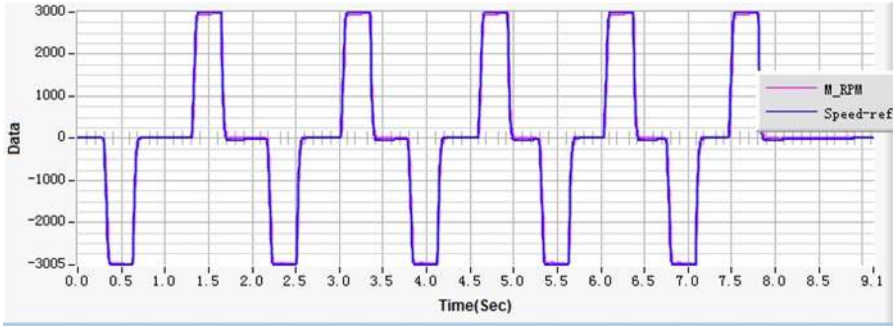


Fig. 11: Speed trace figure

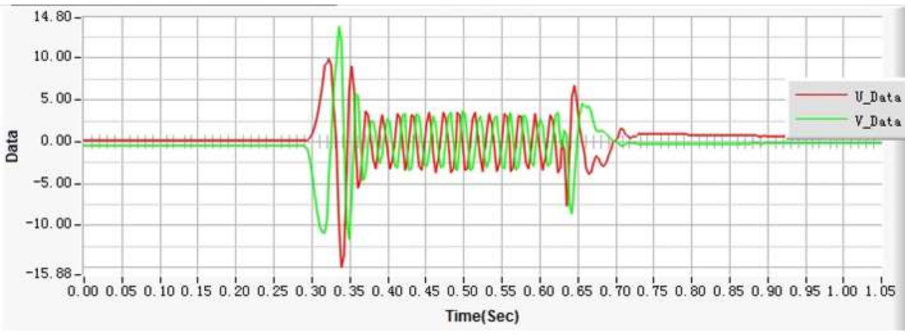


Fig. 12: Current of U、V phase

V. Conclusion

We successfully implement the intelligent flexible lifting system on an Altera Cyclone FPGA. The entire system meets the design requirements regarding stability, accuracy, and speed. Thanks to the integration of flexible lifting system with microprocessor, current sampling, digital filtering, vector control and encoder, the high-performing motion control and good device compatibility and extensibility are ensured. The combination of FPGA parallel calculation with time-division multiplexing enabled by accurate matching algorithm can both save FPGA resources and accomplish vector control algorithm within 1.6us. It can be concluded from the experimental results that the state-machine-based high-speed current vector control unit has the ability to effectively improve the dynamic performance of the system, suppress the current noise and reduce the interference between the control beats of the system. In the further research, the system and state machine will be deeply optimized. Especially, the state machine is optimized to solve the problem that some intermediate quantities are completed by independent units in this design. In the next step, one or two states will be added to further improve the performance of the current loop vector control unit.

Acknowledgements

This work was supported by Basic public welfare research in Zhejiang (LGG19F030007), supported by the teachers scientific research funds of Zhejiang University City College(J-15021),supported by Natural Science Foundation of Zhejiang Province (Y19F020034), supported by the teachers scientific research funds of Zhejiang University City College(JYB18003).

References

[1] J. Fei, R. Deng, Z. Zhang, M. Zhou, Research on embedded CNC device based on ARM and FPGA. In: International Workshop on Automobile, Power & Energy Engineering. Vol. 16, pp: 818–824, 2011.

[2] L. Diao, J. Tang, P. C. Loh, et al. An efficient DSP-FPGA-based implementation of hybrid PWM for electric rail traction induction motor control, IEEE Trans. Power Electron, No.99, pp. 1-10, 2017.

[3] F. A. Aghdam, et al. Implementation of high performance microstepping driver using FPGA with the aim of realizing accurate control on a linear motion system. Chinese Journal of Engineering, pp: 1–8, 2013.

[4] P. Desai Dev, Designing and Improving the Efficiency of Hardware Interpolator, International Conference on Computing, Communication and Automation (ICCCA 2015), pp. 959-962, 2015.

[5]. R. Nithya, K.R. Sarath Chandran, V.P. Chandramani, Run-Time Reconfiguration of Processing Elements Through Soft-Core Processor, 2014 International Conference on Communications and Signal Processing (ICCSP 2014), pp. 813-817, 2014.

[6] D. Jose, P. N. Kumar, J. A. Shirley& S. Ghayathrrie. Implementation of genetic algorithm framework for fault tolerant system on chip. Information Japan, 17(8): 3921-3945, 2014.

[7] Tran, L.T.T, Nordholm, S.E, Schepker, H., Dam, H. Hand Doclo, S. Two-microphone hearing aids using prediction error method for adaptive feedback control. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 26(5), 909–923, 2018.

[8] H.P. Sekaran, M.H. Liyanage, N. Krouglicof, Optimization of Controller Gains for FPGA-based Multi Variable Motion Controller using Response Surface Methodology, IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE 2015), pp. 1307-1312, 2015.

[9] M. A. Saifee, Design and implementation of 2-axis circular interpolation controller in field programmable gate array (FPGA) for computer numerical control (CNC) machines and robotics. International Journal of Computer Applications, 106 (13): p. 1–7, 2014.

[10] Firas Abdullah Thweny, Ahmed Ali Alhammad. Implementation of FPGA Based PSO PID Controller for Feedback IVAX SCARA Robot Manipulator. IJCSET. Vol 3, Issue 10, pp: 367-372, 2013.

[11] J. Acero, D. Navarro, L-A. Barragán, I. Garde, J.-I. Artigas, and J.-M. Burdío, FPGA-based power measuring for induction heating appliances using sigma-delta A/D conversion, IEEE Trans. Ind. Electron, vol. 54, no. 4, pp. 1843–1852, 2007.

[12] Jain, R.V, Aware, M.V and Junghare, A.S. Implementation of a PID control PWM module on Altera DE0 kit using FPGA. 2016 IEEE First International Conference on Control, Measurement and Instrumentation, 341–345, 2016.

[13] Muhammad Majid Gulzar, Ali Faisal Murtaza, et al. Design and Implementation of FPGA-based Concurrent Controller. Proceedings of the Pakistan Academy of Sciences, 55(1), pp: 59-69, 2018.

[14] Dash, S. K, Panda, G, Ray, P. K, et al, Realization of active power filter based on indirect current control algorithm using xilinx system generator for harmonic elimination, Int. J. Electr. Power Energy Syst, 74, pp: 420-428, 2016.

[15] R. K. Pongiannan, et al. Dynamically reconfigurable PWM controller for three phase voltage source inverters. IEEE Transactions on Power Electronics. 26(6): 1790–1799, 2011.