Software Engineering
2020; 8(2): 6-12
http://www.sciencepublishinggroup.com/j/se
doi: 10.11648/j.se.20200802.11
ISSN: 2376-8029 (Print); ISSN: 2376-8037 (Online)

The Design of SVPWM Waveform Generator Based on CPLD_DSP

Li Hui*, Li Jing

Faculty of Automation, Huaiyin Institute of Technology, Huaian, China

Email address:
13645234923@163.com (Li Hui)
*Corresponding author

To cite this article:
Li Hui, Li Jing. The Design of SVPWM Waveform Generator Based on CPLD_DSP. *Software Engineering*. Vol. 8, No. 2, 2020, pp. 6-12.
doi: 10.11648/j.se.20200802.11

Received: July 19, 2020; **Accepted**: August 17, 2020; **Published**: August 27, 2020

Abstract: Space vector pulse width modulation (SVPWM) control technology could obtain quasi circular rotating magnetic field by switching the space voltage vector of the inverter. Under the condition of low switching frequency, the performance of AC motor was better than that of sine wave pulse width modulation. When this technology was applied to inverter control system, better output voltage waveform could be obtained than conventional six step waveform. And in order to further improve the control performance of the motor, it was very necessary to design SVPWM waveform generator. The design of SVPWM waveform generator based on Complex Programmable Logic Device and Digital Signal Processing (CPLD_DSP) was introduced in the article. The main idea of SVPWM was to synthesize reference voltage vector by switching space voltage vector of inverter. Time holding data and sector number could be calculated from coordinate switch of Voltage gathered from inverter by DSP. And the data was sent into CPLD. SVPWM waveform was generated by state machine programed in VHDL. Finally program was download in CPLD chip named EPM1270T144C5N. The transformation from time signal to SVPWM switch signal was finished successfully. The three phrase output signal had good coherence. Thus the controlling function of the motor control system was improved in a certain extent.

Keywords: SVPWM Waveform Generator, Inverter, Sector Number, State Machine

1. Introduction

The research based on the SVPWM waveform generator once became a research hotspot of scholars at home and abroad [1-15]. The traditional SVPWM waveform generator was difficult to substantially improve the control accuracy of the system due to the rough control scheme and simple control means [1]. A adaptive SVPWM algorithm was presented by Mokhtar Aly in renewable energy applications [3]. And Palanisamy Ramasamy presented a SVPWM control strategy for a three phase five level dual inverter fed open-end winding induction motor [4]. Ahmed A presented novel SVPWM based on first order equation [13]. The pure digital motor control system composed of CPLD and DSP could achieve higher control accuracy. The system had good hardware coordination performance, rapid system response and good stability. The use of DSP's powerful computing power combined with CPLD's pure hardware parallel logic control function could not only ensure the accuracy and speed of SVPWM waveform, but also ensure the consistency of the output of multiple control signals, while using CPLD to achieve SVPWM waveform could also greatly save the port resources of the DSP and improve the working efficiency of the DSP, thereby realizing the improvement of the working efficiency of the system.

2. The Overall Design of the System

This control system was mainly composed of keyboard acquisition module, liquid crystal display module, current and position detection module, protection module, coordinate transformation module, driving SVPWM waveform generation module, etc. The specific design block diagram was shown in Figure 1.

3. The Design of DSP Part

The coordinate transformation, data collection and display drive were completed in DSP part. The coordinate transformation and data collection part was mainly executed in the interrupt program. The specific interrupt execution flowchart was shown in Figure 2. When the DSP received the falling edge signal from the CPLD, it triggered an interrupt, and executed this part function of the workflow, and converted the voltage signal into a corresponding time signal, and sent a positive pulse to the CPLD to notify the CPLD to read the time signal in real time.

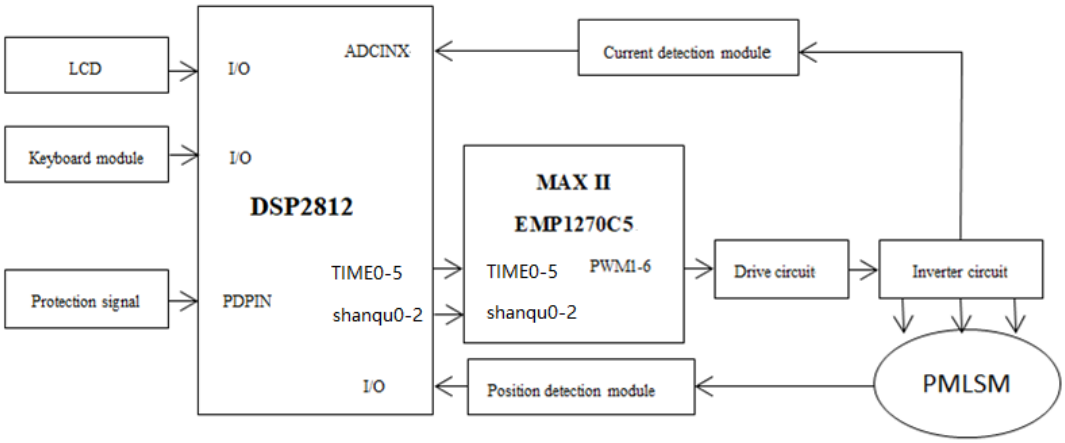

Figure 1. *System overall design block diagram.*

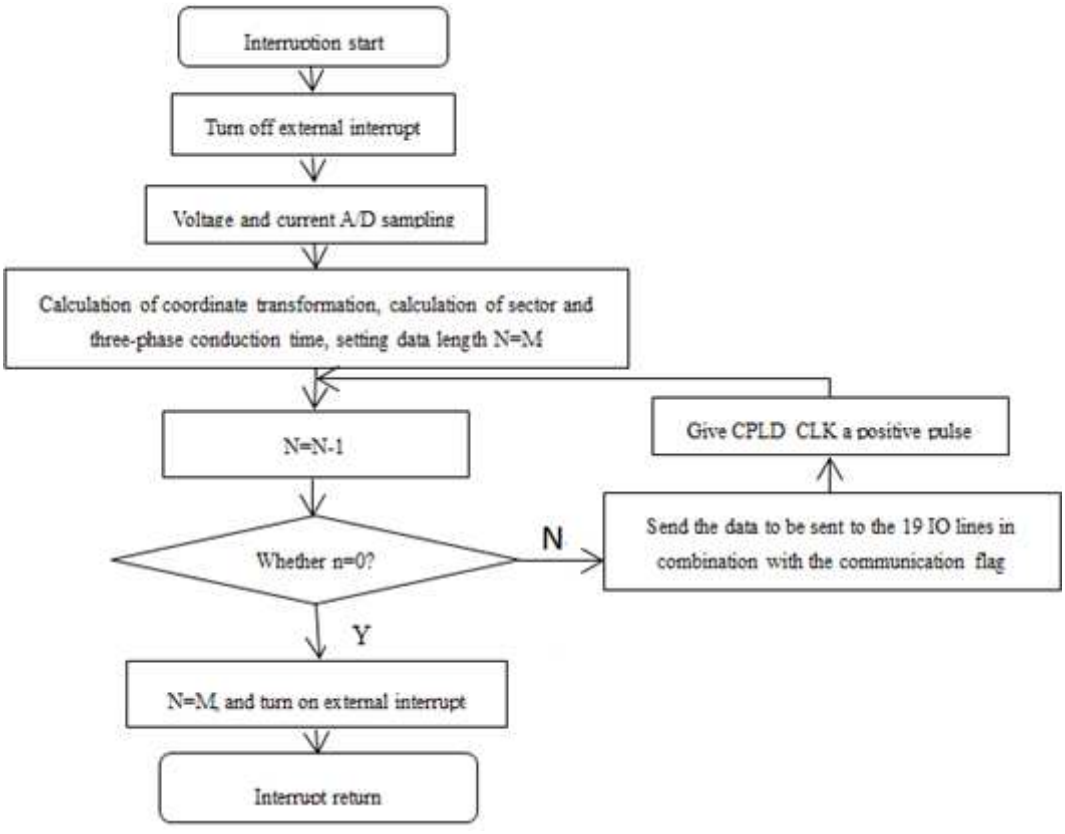

Figure 2. *DSP interrupt flow.*

4. The Software Design in CPLD

4.1. The Design of Top File

This design used the 1MHZ signal as the system's reference clock. The time was expressed by 8 bits data. The maximum time data could be 256 microseconds. The signal frequency of the system output SVPWM waveform was 4KHZ (that is, 250 microseconds). The system adopted a top-down, hierarchical and modular design idea. The top part of the system mainly included DSP data communication module, data latch module, SVPWM wave generation module and dead zone control

module. The second module used the state machine to generate SVPWM waveform based on the time data latched by the previous module. The last module was the dead zone control module, which mainly generated the dead zone control signal to prevent the power device from passing through, and set a 6ms dead zone delay for the bridge arm.

The design block diagram of the top-level module of the system was shown in Figure 3.

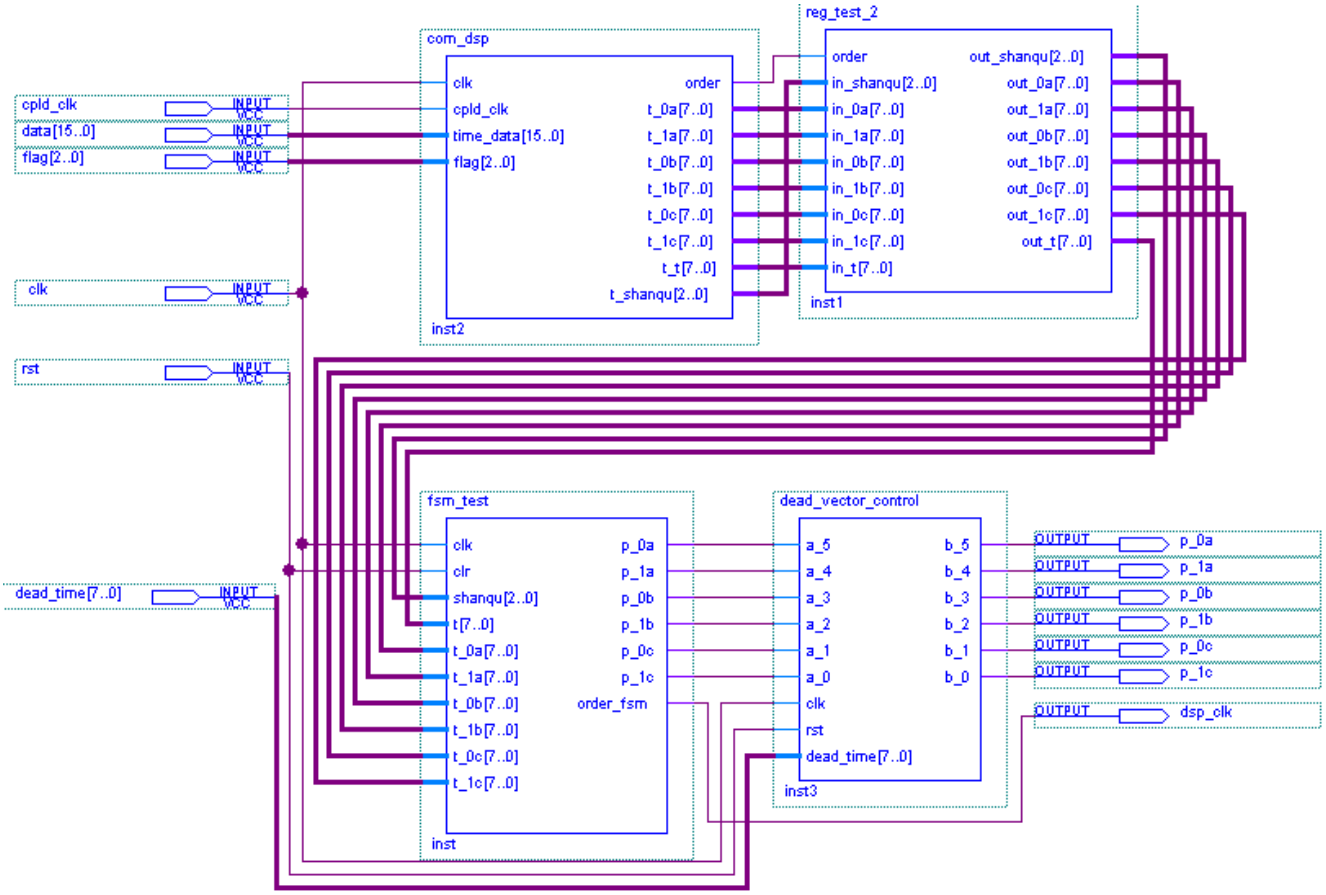

Figure 3. *CPLD control system module.*

4.2. The Design and Simulation of Sub-module of the Bottom Layer

4.2.1. DSP Data Communication Sub-module

This module mainly completed the interface communication with the DSP, and the interface was through the cpld_clk signal. The cpld_clk signal came from the DSP part. Each time the DSP completed the parameter collection and coordinate transformation, it gave cpld_clk a rising edge. CPLD used the rising edge to read the time in a serial manner. The signal convert the serial data into parallel three-phase holding time, interrupt period, sector number and other data under the action of the state machine. Once the conversion was completed, a rising edge latch data was sent to the data latch module. The state transition diagram of the specific implementation was shown in Figure 4.

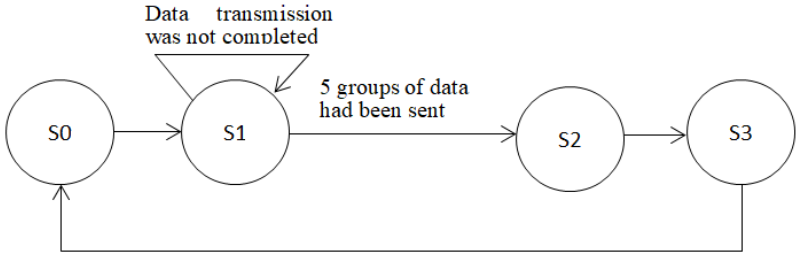

Figure 4. *DSP data communication module state transition diagram.*

Among them, state machine was started by the trigger signal named cpld_clk. The state transition would be triggered once every rising edge. Among them, S0 represented the initial state, S1 represented the sequential reception of 5 sets of data, S2 represented the simultaneous output of 5 sets of data, and S3 represented the provision of a rising edge latch signal to the latch module. The module was simulated by the software plate of Quartus II. The simulation diagram was

shown in figure 5.

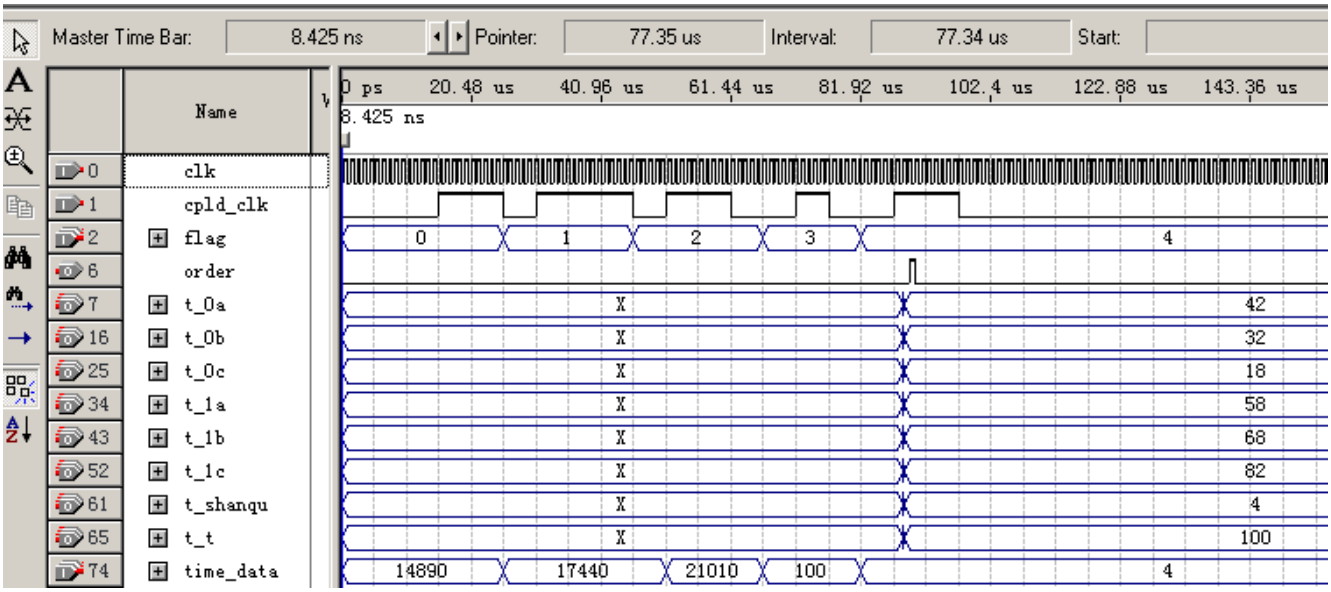

Figure 5. *Simulation diagram of DSP interface module.*

4.2.2. Data Latch Module

The data latch module realized the synchronous latch of all input data, and used the "ORDER" signal as the latch control signal. 1) The "ORDER" signal was the only sensitive signal in the latch process, and the process would be started only when its value changed; 2) When the rising edge of the ORDER signal came, the latch latched the external data to the next one module (waveform generation module) to provide time data; 3) In other states, the time data in the latch remained unchanged, and did not accept external data input.

4.2.3. SVPWM Wave Generation Module

This module used the VHDL language design state machine to generate 6 SVPWM waveforms to drive the IPM, so as to obtain inverter AC power and realize motor operation. This module mainly included three sub-modules. They were carrier signal generator module and comparator module and functional decoding module of the sectors.

(1) The main function of the first sub-module (the carrier signal generator module) was to provide a comparison reference for the comparator module by automatically counting up and down within one SVPWM waveform cycle. The specific implementation code is as follows:

```
process(clk,clr) is
--clr was the clear signal of the data,
--clk is the count clock
begin
if clr='1' then
      t_count<=0;
      ---t_count was the internal count value
elsif rising_edge(clk) then
   if flag=0 then
   --- count up
      if t_count<integer(t/2) then
         t_count<=t_count+1;
         else flag<=1;
         ----add to half of period t and
         -- start to count down
         end if;
   elsif flag=1 then
      ---count down
      if t_count>0 then
         t_count<=t_count-1;
         else flag<=0;
         --After counting down to 0,
         -- it started to count up automatically
      end if;
   end if;
if t_count=0 then
   order_fsm<='1';
   -- At the end of a cycle,
   --a positive pulse is generated
   -- (an interrupt signal to DSP)
   else order_fsm<='0';
end if;
end if;
end process;
```

(2) The second sub-module was the comparator module, which was implemented with three independent state machines, which controlled the three-phase output separately, and compared the input time data with the current count value to determine the state of the state machine. The three independent output signals were different in the different state. Assuming that the switching sequence of the working state of each switch of the upper bridge arm was "1-0-1" in one cycle, each state machine of the system could be include 3 states. There were 3 working states in the cycle, and the holding time of the working state was determined by the input time. When the input sensitive signal changed, the state machine process had been started. The state machine system mainly included

two parts: a combination process and a timing process. In the timing process, the state machine mainly completed the state switching under the action of the clock CLK, and the combination process mainly determined the system next state according to the current state of the state machine and the external input signal. In the next state, decoding output was realized at the same time.

Taking the tmp_0a phase bridge arm as an example (the control methods of the other two phase bridge arms were similar), when the state machine process started, the reference counter (carrier signal generator) started counting immediately, and the comparator was started to compare the current value of the counter and the internal pre-stored time in comparator in real time. The working cycle of the reference counter could be divided into three stages: When the current count value of the reference counter was less than 50% of t_1a, the system state machine would be in the s0_a state, and the bridge arm output on the tmp_0a phase would be high level 1; When the current count value of the reference counter was greater than or equal to 50% of t_1a and less than the sum of 50% of t_1a and t_0a, the system state machine would jump to the s1_a state, and the bridge arm output on the tmp_0a phase would be low Level 0; when the current count value of the reference counter was greater than or equal to the sum of half of t_1a and t_0a, the system state machine would be in the s2_a state, and the bridge arm output on the tmp_0a phase would be high level 1. If the state machine of the system was in an illegal state, both tmp_0a and tmp_1a were output in a high-impedance state.

(3) The third module implemented the functional decoding of sectors. According to the input sector number, the output was determined.

The Quartus II software was used to simulate and verify the logic function of this part. The simulation time was 0-200 μs and the system clock frequency was 1 MHz. Taking sector 1 as an example, the module was simulated.

The corresponding simulation waveform in the sector 1 was shown in Figure 6.

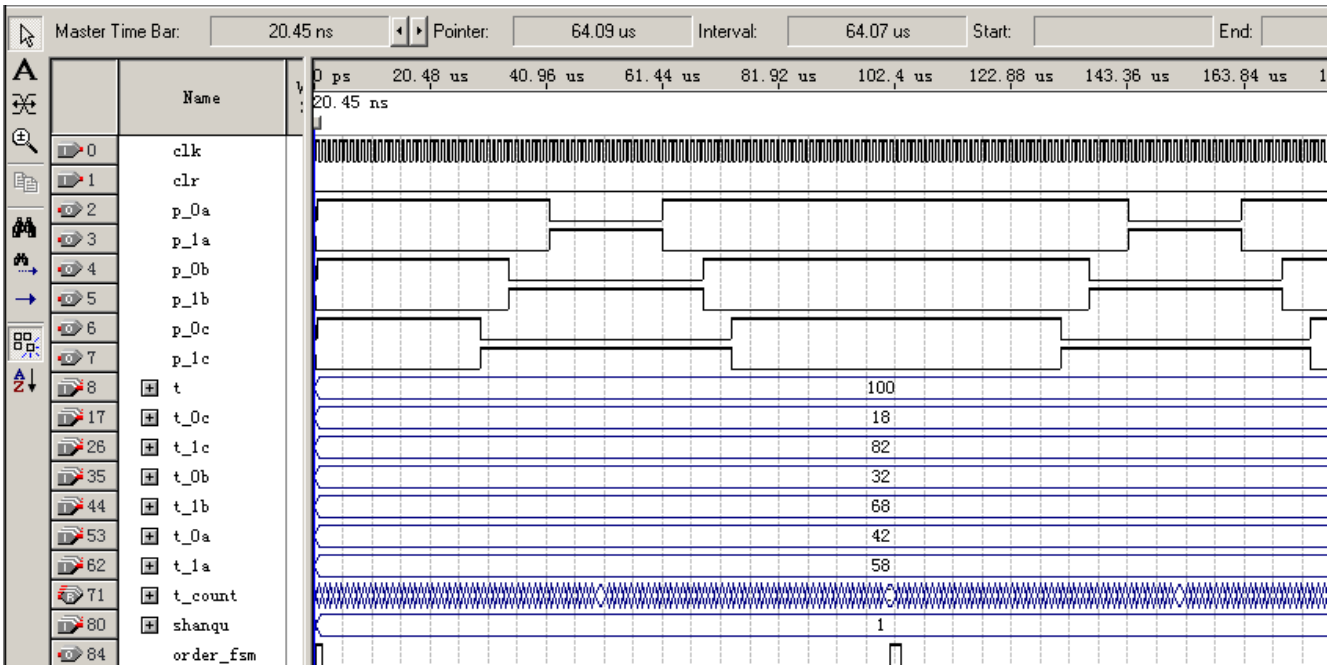

Figure 6. *SVPWM simulation waveform of sector 1.*

It could be seen from the above simulation waveforms that the retention time of the high and low levels of the control signal were depended on the input signals t_1a, t_1b, t_1c and t_0a, t_0b, t_0c, and the sector number shanqu, which completely realized the conversion between the time signal and the trigger signal [2].

4.2.4. Dead Zone Control Module

(1) Definition of dead zone

The previous discussion assumed that the IGBT of the inverter was an ideal switching device, that was, there was no switching delay of the device, and the actual power device was not an ideal device, there was a certain switching delay, and the power device's conduction time was often less than the turn-off time of the device, so the upper and lower switching devices of the same bridge arm in the power device were prone to pass-through or short-circuit when the state changed. In order to prevent a similar situation, it was often necessary to turn off the power tube in the on state for a period of time until the power tube was reliably turned off before turning on another power tube. This period of time is called the dead zone. In the dead zone range, the upper and lower power tubes did not work. The dead zone could be simulated with the software's own delay. The system setting was 4 microseconds [3].

(2) Design method

The main idea of the dead zone control was to delay the rising edge of the input signal without processing the falling edge. The specific implementation used a state machine [4-5] to achieve, the program defined three states s0, s1, s2, in In the initial state s0, output '0' (low level), if the input signal was

detected as '1' (high level), the program transformed to state s1, in state s1, the output was still '0' (low power Level), at the same time the counter was started to count, when the count value was equal to the set value of the dead time, the state shifted to s2, in the s2 state, the output was '1' (high level), and the input signal was detected at the same time. If the input signal was '0', the program returned to the initial state s0. In this way, the rising edge of the input signal could be delayed, and the specific delay time was equal to the dead time.

The simulation waveform was shown in Figure 7.

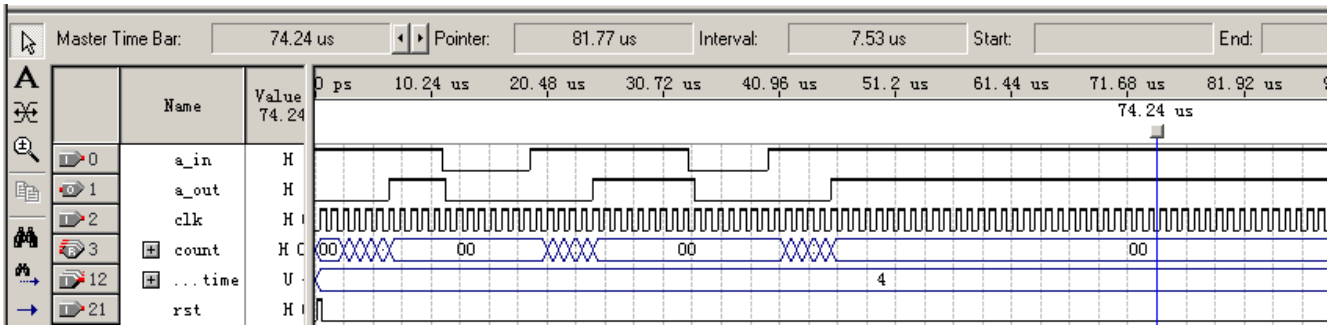

Figure 7. *Simulation waveform of 1-bit dead zone controller.*

The above was the realization of the design of a 1-bit dead-band controller, and a total of 6-channel signals were required to make a dead-band delay, so a top-level file [6-8] was designed to use a 1-bit dead-band controller As the underlying module, the component is instantiated 6 times.

4.2.5. Simulation Study of the Overall Circuit of the System

The above modules were circuit-connected through component instantiation to become the overall top-level file. Taking sector 1 as an example, the final circuit simulation was performed on the system. The simulation waveform was as follows:

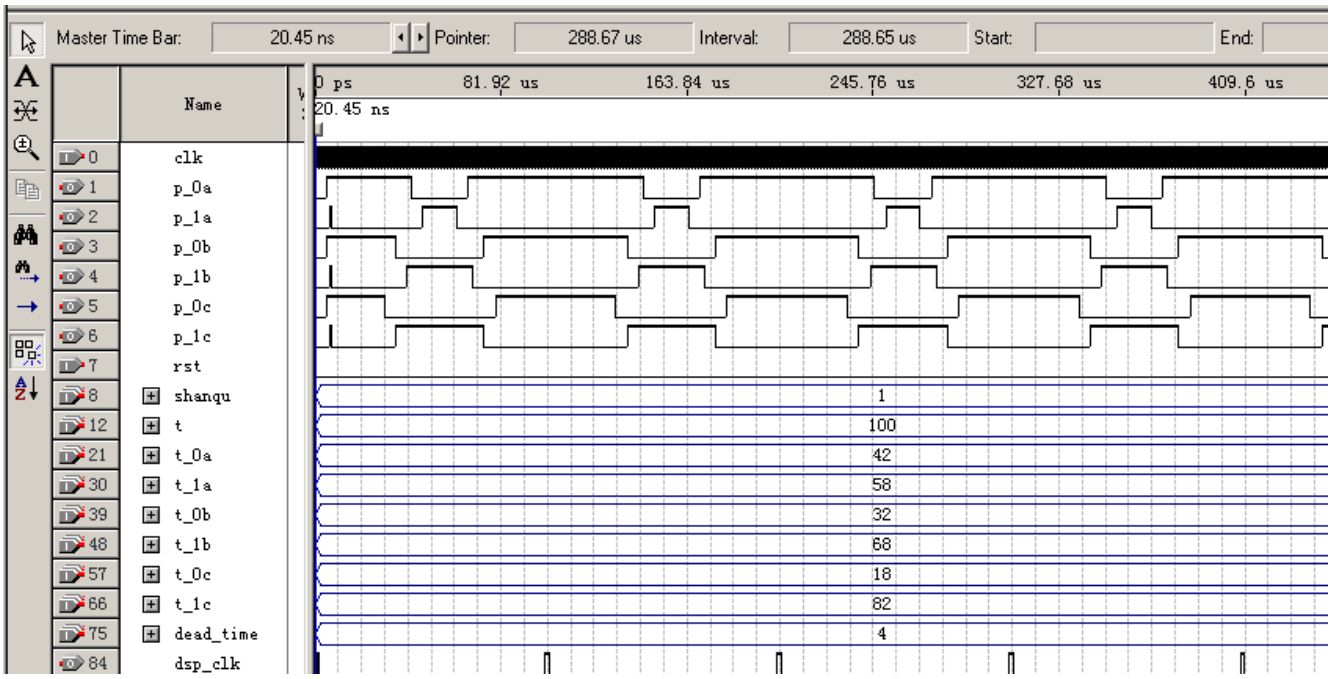

Figure 8. *Simulation waveform of sector 1.*

The final simulation waveform of sector 1 was shown in Figure 8.

It could be seen from the simulation waveform: the 6-channel SVPWM waveform with the sector 1 realized the dead zone delay on the basis of the original waveform respectively.

5. Experimental Results and Analysis

Download the program to the MAX II series CPLD chip EPM1270T144C5N, the system ran well, and the complementary signal waveforms p_1a and p_0a of one pair of bridge arms were measured with an oscilloscope, as shown in Figure 9. It could be seen from the figure that the period of the SVPWM wave was 100 us, which corresponds to the overflow frequency of the carrier signal generator of 10KHZ, and it was used as the external interrupt signal of the DSP.

Figure 10 was the amplified waveforms of the measured p_0a and p_1b, and p_0a and p_1a were the signals of the

complementary power tubes on the same bridge arm. From the figure, we could see that there were two power tubes that were alternately turned on. During the dead time (during this time, both power tubes were turned off to prevent the two power tubes from passing through), it was about 4us, which was consistent with the design requirements. The two experimental waveforms and the circuit simulation were consistent.

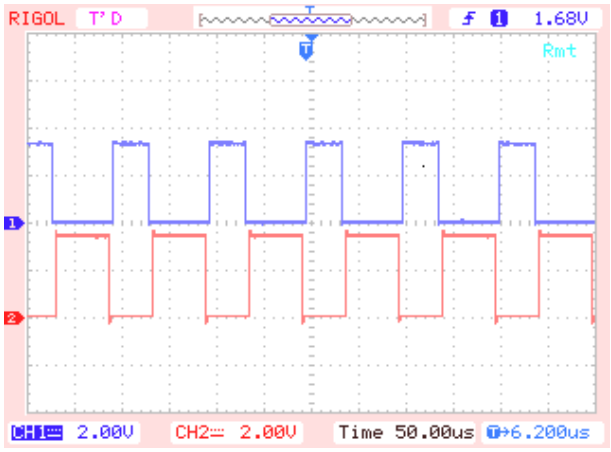

Figure 9. *The waveforms of p_0a and p_1a.*

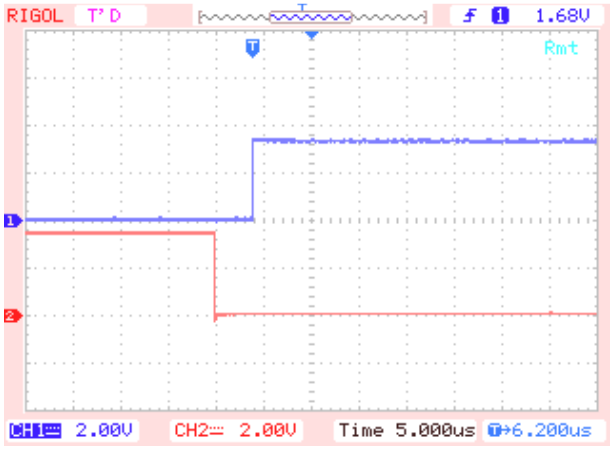

Figure 10. *The amplified waveforms of p_0a and p_1a.*

6. Conclusion

The system design combined the strong computing advantages of DSP and the pure hardware output characteristics of CPLD. Through testing the chip, the total utilization rate of system resources was about 60%. The whole system ran stably, and the output SVPWM waveform had high precision and good consistency, which provided the possibility to fully improve the control performance of the motor.

References

[1] ZHANG Jianwen, YU Haitao, and WANG Yuchen. SVPWM Hysteresis Current Rectification Control Based on Permanent Magnet Linear Motor [J]. MICROMOTORS, 2020, 53, (06): 42–47.

[2] Gao Zhan, Ge Qiongxuan, Li Yaohua, et al. A Carrier-Based SVPWM Begins with the Zero Voltage Vector for Three-Level Neutral Point Clamped Converter [J]. Transactions Of China Electrotechnical Society, 2020, 35, (10): 2194-2205.

[3] Mokhtar Alya, Husam A. Ramadan. Design and implementation of adaptive SVPWM algorithm for multilevel inverters in renewable energy applications [J]. International Solar Energy Society, 2019, (183): 745-754.

[4] Palanisamy Ramasamy, Vijayakumar Krishnasamy. SVPWM control strategy for a three phase five level dual inverter fed open-end winding induction motor [J]. ISA Transactions, 2020, (102): 105-116.

[5] H. Merabet Boulouiha, A. Allali, M. Laouer, et al. Direct torque control of multilevel SVPWM inverter in variable speed SCIG-based wind energy conversion system [J]. Renewable Energy, 2015 (80): 140-152.

[6] K. Thangarajan, A. Soundarrajan. Performance comparison of permanent magnet synchronous motor (PMSM) drive with delay compensated predictive controllers [J]. Microprocessors and Microsystems, 2020 (75): 103081-103091.

[7] A. Sangari, R. Umamaheswari, M. G Umamaheswari, et al. A novel SOSMC based SVPWM control of Z-source inverter for AC microgrid applications [J]. Microprocessors and Microsystems, 2020 (75): 103045-103058.

[8] R. G. Shriwastava, M. B. Daigavane, P. M. Daigavane. Simulation Analysis of Three Level Diode Clamped Multilevel Inverter Fed PMSM Drive Using Carrier Based Space Vector Pulse Width Modulation (CB-SVPWM) [C]. 7th International Conference on Communication, Computing and Virtualization 2016, (79): 616-623.

[9] Nopparat Moungkhum, Wanchai Sbsingha. Voltage Control by DQ Frame Technique of SVPWM ACDC Converter [C]. 10th Eco-Energy and Materials Science and Engineering (EMSES2012), 2013 (34): 341-350.

[10] Lachtar Salah, Bahi Tahar. SVPWM performance of PMSM variable speed and impact of diagnosis sensors faults [C]. International Conference on Technologies and Materials for Renewable Energy, Environment and Sustainability, TMREES15, 2015 (74): 679-689.

[11] Chen Yiguang, Bao Zhenmao, Lin Chenghan. Fault-Tolerant SVPWM Control Strategy for Five-Phase PMSM under Single-Phase Open-Ciruit Fault [C]. 10th International Conference on Applied Energy (ICAE 2018), 2018: 22-25.

[12] N. Rajeswaran, M. Lakshmi Swarupa, T. Sanjeeva Rao. Hybrid Artificial Intelligence based Fault Diagnosis of SVPWM Voltage Source Inverters for Induction Motor [C]. PMME 2016, 2018 (5): 565-571.

[13] Ahmed A. Mansour. Novel SVPWM based on first order equation [J]. Journal of Electrical Systems and Information Technology. 2015 (2): 197-206.

[14] Khaled N. Faris, Ahmed A. Mansour. Hala S. Khalil. Comparative study of two linearization methods for time intervals generation of SVPWM technique [J] Journal of Electrical Systems and Information Technology. 2016 (3): 377-386.

[15] YINGJIE HE, YUNFENG LIU, CHAO LEI, et al. Equivalent Space Vector Output of Diode Clamped Multilevel Inverters Through Modulation Wave Decomposition Under Carrier-Based PWM Strategy [J]. IEEE ACCESS, 2020 (8): 104918-104932.