Research on control law accelerator of digital signal process chip TMS320F28035 for real-time data acquisition and processing

Shuangle Zhao, Xueyi Zhang, Shengli Sun and Xudong Wang

Applied Liberal and College, Tianjin University of Science and Technology, Tianjin economic and Technology Development Zone No. 13, 29 Avenue.
Email: zhsl347new@qq.com

Abstract. TI C2000 series digital signal process (DSP) chip has been widely used in electrical engineering, measurement and control, communications and other professional fields, DSP TMS320F28035 is one of the most representative of a kind. When using the DSP program, need data acquisition and data processing, and if the use of common mode C or assembly language programming, the program sequence, analogue-to-digital (AD) converter cannot be real-time acquisition, often missing a lot of data. The control low accelerator (CLA) processor can run in parallel with the main central processing unit (CPU), and the frequency is consistent with the main CPU, and has the function of floating point operations. Therefore, the CLA coprocessor is used in the program, and the CLA kernel is responsible for data processing. The main CPU is responsible for the AD conversion. The advantage of this method is to reduce the time of data processing and realize the real-time performance of data acquisition.

1. Introduction
In real-time data acquisition, if the data analysis program running time is longer than the data acquisition time, it will cause the ADC not work for a period of time, resulting in the omission of the detection results [1]. The core is characterized by the parallel operation between the master kernel and the CLA kernel, and the real-time performance of the data acquisition is achieved. DSP TMS320F28035 introduces its inner structure as shown in figure 1. The CLA core frequency is the same as the main CPU, which is 60MHz, which guarantees the speed required for high-speed data operation.

Figure 1. Chip 28035 (left) and Internal CLA Architecture (right)
CLA allows direct access to the ePWM, the comparator, and the ADC result register. CLA can be programmed using the C language and assembly language, and the CLA support settings need to be prepared in the pre-processing section before compiling [2].

Figure 2. Normal DSP operation model (a) and operation mode with CLA (b)

The DSP with CLA kernel can synchronize the main processor and the CLA kernel seen as figure 2. This parallel operation reduces the running time and improves the operation efficiency. It is very important for time sensitive programs.

2. Steps for the Use of the CLA Core
CLA kernel programming, you can use the C language or assembly language, here focuses on the use of C language [3].

2.1. CLA Interrupt Settings
The PCLKCR3 registers enable the CLA clock, i.e., SysCtrlRegs.PCLKCR3.bit.CLA1ENCLK = 1;

2.2. Initialize the Peripheral interrupt (PIE) Controller, and then Copy the PIE Vector Table into the DSP Memory by Programming;

2.3. CLA ISR Set
Distribute the user-defined interrupt service program (ISR) to the peripheral interrupt vector table (PIE), copy the CLA program code into memory, and get the initial address of the CLA program memory. Then, the offset address of the CLA task vector is computed based on the initial address, and the 8 CLA task vectors (MVECT) are filled.

2.4. Mapping the Task Interrupt Source and Opening the CLA Interrupt
The interrupt source for 8 tasks in the CLA program, using the Cla1Regs.MPISRCSEL1.bit.PERINT1SEL statement setting, 0 for the ADCINT interrupt trigger, 1 for none, and 2 for the EPWM1INT interrupt trigger. You do not need to set up this interrupt when using software to run tasks. Interrupt enable register MIER.

2.5. Enable CLA Interrupts
This section includes peripheral interrupt extensions PIEIER, IER, EINT, ERTM which will be enabled for 1.

2.6. IACK Software Triggers and CLA Data Space and Program Space Mapping Registers
IACK=1 IACK is the software triggers.
MMEMCFG.bit.RAM0E=1 CLA RAM0 data space.
MMEMCFG.bit.RAM1E=1 CLA RAM1 data space.
MMEMCFG.bit.PROGE=1 CLA program space.
When allocating memory to the CLA program and data, the CPU cannot access the program and data of the corresponding address.

2.7. CMD File Configuration

![Figure 3. DSP 28035 RAM mapping](image)

The CMD file allocates memory space to each module of the DSP [4]. The memory map for the 28035 chip is shown in figure. Figure 3 shows the DSP 28035 the memory allocation, the CLA program must be kept in the range of 0x009000-0x00A000 addresses, CLA data can be stored in the 0x008800-0x008C00 address range is within the range of L1DPSARAM, or the 0x008C00-0x009000 address range that is within the range of L2DPSARAM, when CLA use these programs or data storage space, and CPU will not be able to use this space to write the program, so CPU and CLA core part based on chip memory capacity.

3. CLA Core Debugging Platform

![Figure 4. CCS 5.5 Developing Environment. Hardware (Left) and Software (Right).](image)

Using TI's TMS320F28035 minimum system, the DSP integrates a CLA core, using CCS5.5 as a program debugging platform. In debugging, you can observe the value of the control register in the CLA kernel.
4. Experimental Result
For testing the CLA work, there are two same projects written using CLA core and not-using CLA. Figure 5 shows the two experimental test codes. When AUX1 pin is high-voltage, the DoArcDetect function executes twice, and then the AUX1 is low-voltage. The oscilloscope DL850E saves the AUX1 signal. The DoArcDetect function can use CLA core or without it. The cost time of sub-function is described in table 1.

![Image](image1.png)

**Figure 5.** The test project code.

| Sub Function          | Time(ms) |
|-----------------------|----------|
| Data Acquisition      | 10       |
| Data Processing       | 19       |

![Image](image2.png)

**Table 1.** The function time

(a) ![Image](image3.png) (b) ![Image](image4.png)

**Figure 6.** The experimental result. (a) not-using the CLA core. (b) Using the CLA core.

The experimental result is shown as in figure 6. The function execution time of the non-using CLA core is 29ms, and the other execution time is only 18ms when using the CLA core. So it is losing 9ms data when the software project acquisition data, whereas using the CLA core the data processing section execution time is 8ms, shorten then the data acquisition section 10ms, so this is not losing the data.
5. Conclusion
In conclusion, the CLA core help the main CPU core in parallel program execution. It is import when the program includes the AD data acquisition section and data processing section. When data processing time is larger than the data acquisition time, and software is also sequence run, the data must be lost. So the CLA core support an effective methods to solve the problem.

The appearance of CLA kernel gives more choices to designers and programmers, reduces the workload of main CPU and improves the real-time performance of data analysis. But CLA also has its shortcomings, the use of many constraints, cannot be nested interrupts, cannot use more than 64 data, using the judgment statement of lower efficiency in CLA, in addition to the ADC result register, a comparator and a ePWM, CLA cannot access other peripherals. If CLA takes up too much data or programs, CPU will not be able to access the remaining memory space, resulting in a waste of memory space.

6. Acknowledgments
Project supported by Tianjin University of Science and Technology Youth Innovation Fund (2016LG31)

7. References
[1] Yang Chuanhai, Xiao Bing. Design of Infinite Impulse Response Low Pass Filter Based on TMS320F28035 Control Law Accelerator [J]. Electric Drive, 2012, (05): 26-29.
[2] DSP application development utility subroutine [M]. People’s post and Telecommunications Press. Wang Anmin, Cheng Yu edited, 2005.
[3] TMS320x2803xPiccolo Control Law Accelerator (CLA) Reference Guide. Texas Instruments Incorporated. 2009.
[4] C2000 CLA C Compiler. http://processors.wiki.ti.com/index.php/C2000_CLA_C_Compiler?keyMatch=cla%20c&tisearch=Search-EN [EB/OL]. Texas Instruments Incorporated.