Space Vehicle Control Software Architecture Design Based on Heterogeneous Multi-core Processor

Chao Zhang[a,b], Dongdong Wang[a,b], Aiju Shao[a,b]
[a]Beijing Institute of Space Long March Vehicle, 1Nandahongmen Fengtai District, Beijing China
[b]China Academy of Launch Vehicle Technology, 1Nandahongmen Fengtai District, Beijing China
* Corresponding author: zhangch20090909@163.com

ABSTRACT

The control software running on the traditional single-core processor can not meet the requirements of multi-task parallel computing and high real-time performance. An aircraft control software architecture design based on heterogeneous multi-core processor is proposed in this paper. The control software functions are divided into multiple tasks, which are decoupled and distributed on different CPU (Central Processing Unit) to achieve parallel computing, and meet requirements of high performance in complex tasks. In this paper, Asymmetric multi-processing (AMP) is taken as an example, and inter-core communication mode based on shared memory is adopted to complete the design of aircraft control software architecture.

Keywords: Space Vehicle Control, Heterogeneous Multi-core, Software Architecture Design

1. INTRODUCTION

With the rapid development of aerospace technology, the requirements for automation and coordination capabilities of aerospace vehicles are getting higher and higher, and then the requirements for control system capabilities are getting higher and higher. Faced with such complex tasks, the traditional single-core flight control system has been increasingly difficult to meet. This paper proposes an aircraft control software design architecture with heterogeneous multi-core parallel computing to solve this problem. In this architecture, tasks are decomposed and deployed on different computing cores for multi-core parallel computing to achieve higher computing power and real-time performance.

At present, multi-core processors are mainly divided into symmetric multi-core processors (SMP)[1] and asymmetric multi-core processors (AMP)[2]. Most multi-core processors adopt symmetric multi-core processors, and all CPUS are equal. Asymmetrical Multi-core Processor (AMP) is mainly used for embedded chips, one CPU is the master control and the others are the slave control. The master CPU controls system operation and resource management, and the Slaves independently handle real-time tasks and communicates with each other through shared memory[3,4]. The chip used in this paper is ZYNQ UltraScal[5], which is a new asymmetric multi-core programmable chip designed by Xilinx Company and has quad-core Arm Cortex-A53.

The aircraft control software architecture of heterogeneous multi-core parallel computing[6] divides the flight control software into multiple computing tasks, which running on multiple CPUS respectively.

2. SCHEME DESIGN IDEA

The four core processing chip selected in this paper adopts AMP mode, each core adopts bare core operation. Hardware resources can be reasonably allocated through configuration to ensure non-conflict of resources in multi-core parallel operation.

The general idea of the aircraft control software architecture of heterogeneous multi-core parallel computing is that main CUP(CPU0) is responsible for the operation of main control execution task, CPU1 is responsible for the operation of multi-mode navigation task, CPU2 is used to the operation of image pattern recognition task, and CPU3 is responsible for the operation path planning and guidance task. The master CPU0 and the slave CPU (1, 2, and 3) use shared memory communication to complete data interaction.

The bottom driving layer mainly completes sensor data acquisition, all kinds of IO signal control input and output tasks, etc; The data interaction layer mainly completes the operation of message queue data storage, forwarding and structure

Third International Symposium on Computer Engineering and Intelligent Communications (ISCEIC 2022), edited by Xianye Ben, Proc. of SPIE Vol. 12462, 124620Z · © The Authors. Published under a Creative Commons Attribution CC-BY 3.0 License · doi: 10.1117/12.2660838

transformation. The data computing layer mainly includes data fusion computing, data correction computing and inertial extrapolation navigation computing. The data application layer is used to pre-launch inspection, timing control, in-flight acquisition, actuation control and other functions. The overall architecture is shown in Figure 1.

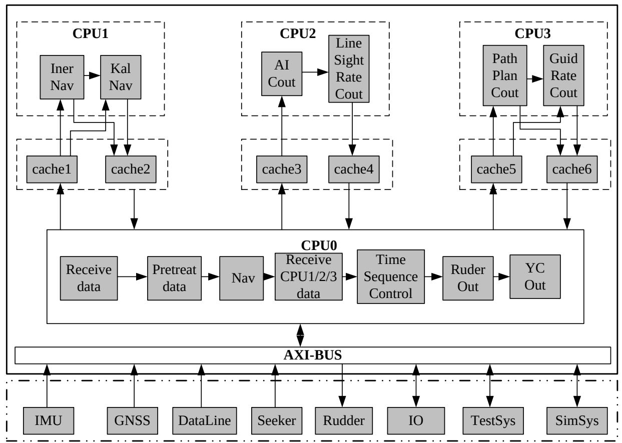

Figure 1. Framework of multi-core heterogeneous flight control software

3. AIRCRAFT SOFTWARE ARCHITECTURE DESIGN

The aircraft control software architecture based on heterogeneous multi-core parallel computing takes the task as the major job. Through task scheduling, the load can be balanced and the software run efficiently. The communication mechanism of shared memory is adopted between tasks. The following details the design idea of aircraft control software architecture based on heterogeneous multi-core parallel computing.

3.1. Master Execution Task

According to the overall composition and system tasks of the aerospace control system, the overall architecture of the aircraft control system software is divided into three layers, namely hardware drive layer, data link interaction layer and data application layer, according to data acquisition, data interaction and data application. The specific hierarchical access design is shown in Figure 2.

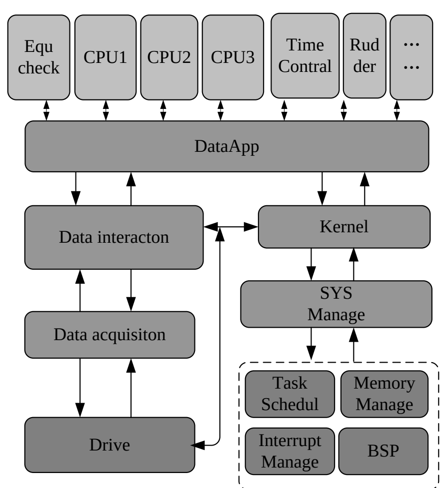

Figure 2. CPU0 software functional architecture

The hardware driver layer is the bridge between software and hardware. According to different hardware interface such as 1553B, SPI, UART, IO, etc, set the corresponding rate, corresponding communication protocol, data acquisition mode such as interruption mode, query mode and so on.

The data link interaction layer ,classifies the data of each port of the hardware driver, unpacks the data, and converts it into the corresponding message queue data structure. The data structure of message queue includes device number, data update flag etc. .

The main function of the data application layer is data filtering processing, stage control calculation,etc. The application layer waits for the data update of the driver layer, calculates the algorithm according to the data flow rules, and sends the calculated results to the message queue. The flow chart of the application layer is shown in Figure 3.

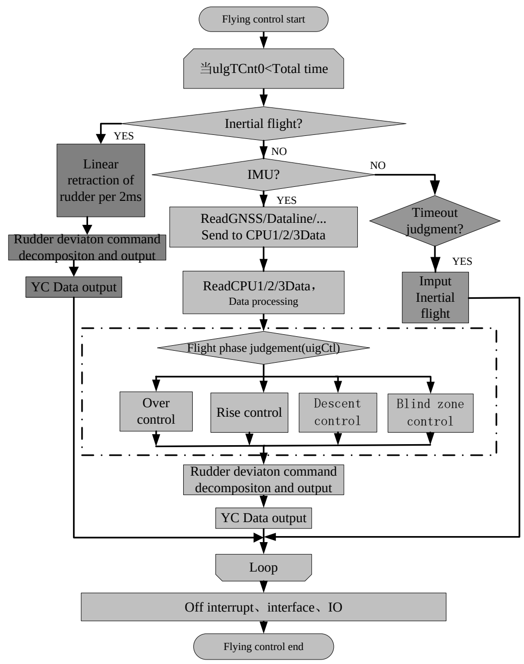

Figure 3. Sequence diagram of CPU0 software application layer

After data application layer starts running, it first determines whether it is in inertial flight state. If so, it enters inertial flight and conducts linear retracting until the motion reaches zero. If it is in non-inertial state, it judges IMU, GNSS, data link, delay, PPS, seeker data, and forwards the data to the corresponding CPU. Determine whether the IMU data is valid., Progressive corresponding calculation and read the output data of the other three CPUs.

3.2. Multi-mode Navigation Computing Task

The calculation tasks of multi-mode navigation task mainly include Kalman navigation calculation and pure inertial navigation calculation. Pure inertial navigation uses IMU data,, to carry out attitude calculation, speed calculation and position calculation. Kalman navigation calculation uses pure inertial navigation data (which is position, speed and attitude) and GNSS data, namely wet-guide speed and position. Kalman filter is used to predict position, speed and attitude. The functional sequence of CPU1 is shown in Figure 4.

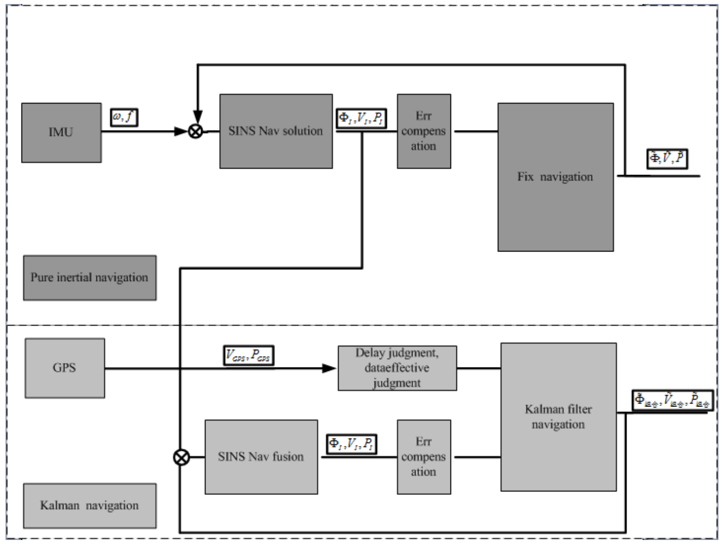

Figure 4. Sequence diagram of CPU1 software functions

Pure inertial navigation navigation calculation content is: attitude, speed, position calculation. The attitude calculation and speed are based on the attitude data of the previous step and the angular velocity and acceleration of the current step, and finally calculate the attitude data at k moment.

Kalman navigation calculation uses Kalman filter to deduce the position and velocity. The kalman navigation filter algorithm use the result of pure inertial navigation and the compensation value of position and speed information which outputs by GNSS receiver.

3.3. Seeker and AI Pattern Recognition Calculation

Seeker calculation and AI pattern recognition computing tasks mainly include image pattern recognition, line of sight Angle and line of sight Angle rate extraction calculation.

Image pattern recognition calculation to capture and identify the target, the main content :using Darknet network ,using boundingBox,fusion feature, tracking fusion algorithm calculation the center position of the target in the image coordinates.

Line of sight Angle and line of sight rate calculation provide line of sight Angle and line of sight velocity for terminal guidance control. The main content is based on the target center position in the seeker coordinate system, the supporting role of the aircraft platform, the preset Angle of the aircraft platform, The line of sight Angle and line of sight rate are calculated by using motion equation, geometric relation, line of sight rate decoupling algorithm and Kalman filter.

3.4. Path Planning and Guidance Calculation

Path planning calculation and guidance calculation tasks mainly include path planning calculation and guidance rate calculation. Path planning uses IMU data, GPS data various binding data, current control parameter data, etc., using the aircraft movement model, and the improved Newton iterative algorithm to calculate the control adjustment parameters.

Guidance rate overload instruction is using pitching direction overload, yaw direction of overload, aircraft time, emission rate of x to the component, system for launch of the speed rate of weight, speed, to expect y Angle, theory value, the current velocity Angle velocity Angle, the pitch attitude Angle, step on the overload instruction value, Calculate the overload instruction of yaw and pitch direction.

4. EXPERIMENT VERIFICATION

4.1 Testing Methods

The IMU sensor sends habitual group data in a cycle of 2ms, GNSS sends navigation data in a cycle of 50ms, and data link data in a cycle of 80ms. A control cycle is calculated, and the control cycle is 2ms. The work output is four steering gear control data. The reading timing accuracy is 0.01ms, and the calculation time deviation should be less than 0.1ms.

4.2 Testing Results Statistics

The entire flight trajectory was from takeoff to landing, and 153, 086 frames of telemetry data were received. In the case of a single core processor, the shortest calculation time was 1.52ms, and the longest calculation time was 2.93ms, the average calculation time was 2.06ms; In the case of dual core processors, the shortest calculation time was 0.96ms, and the longest calculation time was 2.54ms, the average calculation time was 1.53ms;In the case of quad core processors the shortest calculation time was 0.62ms, and the longest calculation time was 0.97ms., the average calculation time was 0.83ms.

The actual flight control output results are analyzed, and the results of hardware-in-the-loop simulation based on the algorithm design are consistent. In the whole flight process, the control data change curve of the actuator is shown in Figure 5.

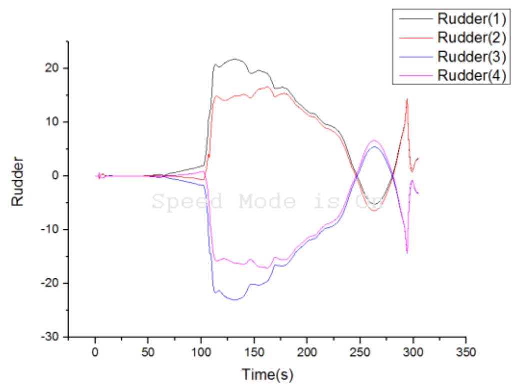

Figure 5. Variation curve of actuation control

As shown in the figure, after take-off, the control program outputs four dynamic curves. The analysis shows that the flight data is correct and the control effect meets the overall design requirements. It shows that the design of flight control software based on multi-core processor is correct.

5. CONCLUSION

This article is based on heterogeneous multi-core parallel processor, proposes a software architecture, which can be used to Space Vehicle Control. The stability and reliability of the software framework are verified by ground comprehensive experiment, Multiple tests and flight test.. Using this architecture design of Vehicle Control software has the characteristics of usability, high scalability and high reliability, for other aircraft flight control software design has some reference value.

REFERENCES

[1] RenHe Cai, “Research and implementation of embedded multi-core operating system kernel based on ARM Cortex-A9 MPCore,” University of electronic science and technology of china.

[2] Xingjie Zheng, “Research and implementation of real-time strategy for embedded multi-core system based on ZYNQ platform,” University of electronic science and technology of china.

[3] Gong Xingliang, Sun H,Zhou Qiang, Wang Hao, Liu Wei, “Research on multi-core AMP architecture for MPSoc,” Microcontrollers & embedded systems. Papers 22(2),32-35(2022).

[4] Ren Zhi ming., “Design and analysis of real-time task scheduling architecture based on DSP C6678 multi-core platform,” Shipboard electronic countermeasure. Papers 39(5),90-93(2016).

[5] Xing Yan fang, Zhu Jin fu,Zhou Xiao mei, “Design of multi-core processing based on ZYNQ,” Computer technology and development. Papers 28(3),60-66(2018).

[6] Lin Li ming, Yu Dong, Hu Yi, Qin Cheng-gang, Wang Zhi-cheng, Fend Qiang, “Development of heterogeneous multi-core microprocessor based on embedded NC system,” Modular machinetool & automatic manufacturing technique. Papers (11),78-82(2012).