Design and Implementation of Private Owner Code Region in Memory Protection Unit

Bo Qian[1, a)], Weiping Jing[1, b)] and Bin Jiang[2, c)]

[1]Jiangsu Key Laboratory of ASIC Design, Nantong University, Nantong Jiangsu 226019, China.
[2]C*Core Technology C0. Ltd, Suzhou Jiangsu 215000, China.

[a)] Corresponding author: 946161273@qq.com
[b)]13906294039@163.com
[c)]bin.jiang@china-core.com

Abstract. Embedded systems often use a number of multitasking operations and controls, these systems must have a mechanism to ensure the implementation of the current task, so that it has no effect on other tasks, that is, to prevent the system's resources and tasks are accessed illegally. There are usually two ways to achieve this goal: software and hardware protection. Software protection means alone software to protect resources of the system, and no hardware involved. However, when a multitasking operation occurs, it must be coordinated through the system and can easily affect the operation of the current task, resulting in irrational use of resources. Conversely, if the system has specialized hardware to detect and limit access to resources, there is a good guarantee of ownership of resources and the rules governing the running of the task. The chip's internal memory protection unit is very effective to achieve such a function. The article focuses on its private owner code region, the region can not only be accessed by the user, and when the region is monopolized by the user, which can also be used as a user to access other regions.

Key words: accessed illegally; multitasking; guarantee of ownership; Embedded systems; implementation

INTRODUCTION

SOC is the abbreviation of System on Chip, meaning chip-level system; it integrates the key components of the system on a chip, which is equivalent to a tiny system [1]. C0 MCU is a 32-bit low-power microprocessor core independently designed by Suzhou C*Core Technology Co., Ltd. MPU is the chip's internal memory protection unit, which can achieve the chip's internal storage space and security management. It allocates different storage spaces for different applications of the chip and protects the data of each application storage space from being illegally accessed and modified and can indicate the errors when the memory and the protected registers are accessed illegally [2]. Private Owner Code Region: referred to as CP region. The size of the region is configurable and can be set anywhere on the chip. The design of the region is a great innovation. Any user can apply for a CP region. After the region is monopolized by the user (valid), the user can flexibly configure access permissions of the region and other users cannot change. If other users want to use this region, only the user can release the region's ownership before other users can re-occupy and use.

FUNCTION AND FEATURES OF MPU

The main functions of the MPU are as follows: It can reasonably divide the region and set permissions of FLASH and RAM. It also solves the problem of ROM encryption and manages the permissions of special registers. The MPU module uses separate access control and data encryption storage control on the chip storage region to realize permissions management and storage encryption for the storage region by user [3]. Only users can turn on and turn off the use of MPU.

Protection of the EFLASH region: You can configure four regions with user attributes, four for user code region (UC) and four for user data region (UD). Different users can configure the register to set whether to allow the user to read, write, execute and whether to allow other users to read, write, execute, and so on. The range of each user region in FLASH can be flexibly configured, and the range between users is not allowed to be out of bounds. In addition, the user can also configure the FLASH region is encrypted storage.

Protection of the SRAM region: It can be configured according to different user needs for SRAM Data Region; referred to as SD region, a total of 4, one user can occupy multiple data region. The region's read, write, execute permissions control is similar to FLASH permission control. The data of SD region is not encrypted, the range of each region can be flexibly configured to achieve independent protection of SRAM regions for different user.

Protection of the ROM region: The ROM region is encrypted and stored without permission management.

DESIGN MODELS AND THEORIES OF CP REGION

Fig.1 is the functional diagram of CP module

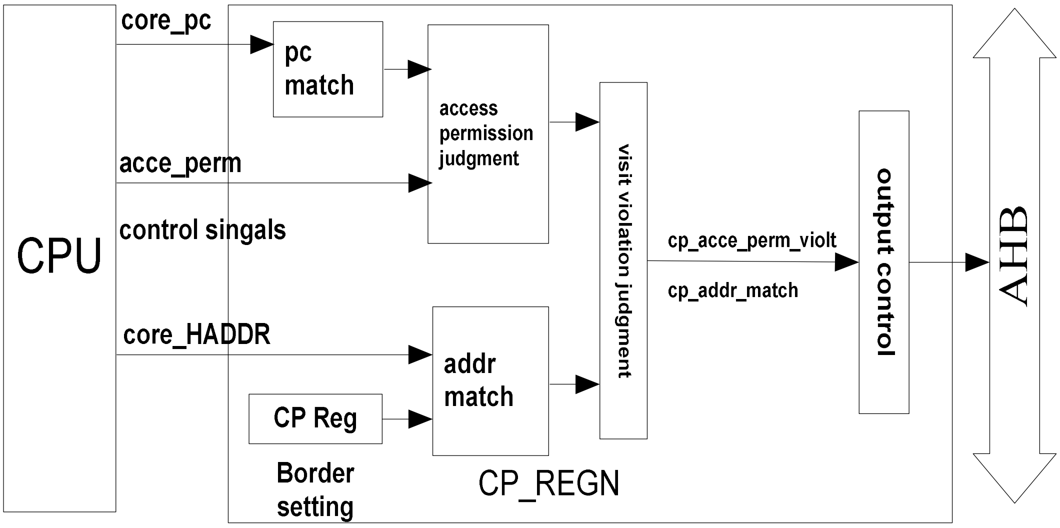

FIGURE 1. Functional diagram of CP module

Design Models

CP module can be divided into two main parts: Control part and Output part.

Control Part

Pc match unit, which implements the code and the boundaries of the regional configuration address to determine the region where the code is located; Address match unit, which compares the address of the visit with the border address of each region to determine the region of access; Access permission judgment unit, that determines whether the actually occurring access meets the set access permissions; Visit violation judgment unit, which compares the addresses and permissions of each region to make single-visit valid. The address boundaries and access permissions of the programmable region are set by the CP control register.

1. Control Register: The user to configure the control register, to achieve the programmable region settings and catch up illegal access. The programmable region settings include: programmable region address of the upper and lower boundary and programmable region access permissions settings.

2. Access Violation Mechanism: When the chip accesses the memory through the AHB bus, the MPU compares the high and low addresses of each region to determine which region the current address to be accessed belongs to. When the corresponding address matches, the current command will determine whether the current operation can be performed according to the access permissions of the corresponding region [4]. If the current region does not allow the current operation, then there will be an access violation.

3. Regional judgment: When the CP region is monopolized by the user, other users cannot monopolize again, and can only be monopolized again after the user released. In addition, the user cannot monopolize the CP region multiple times, that will generate invalid error [5].

Output Part

Based on the result of the control part determination, it is determined whether to output the access information to the peripheral bus. If the control part determines that the access is valid, the access information is output, and if the control part determines that the access is violated, the access information is blocked. In this way to achieve the role of memory protection.

Specific Functions

The overall function of the module to use: Set the CP control register, CPU initiated access, the access information through the control part of the CP module to effectively determine. Finally, MPU sends output or blocked information to the peripheral bus through the output part. The signals are shown in Table 1.

TABLE 1. CP region signal description

Name	Description	Type
core_HADDR	Current execution region	Input
core_pc	Region of code	Input
acce_perm	Current operation	Input
cp_valid	CP is monopolized	Input
cp_uc_pc_match	CP region match	Input
cp_regn_perm	CP region permissions	Input
cprh/l_reg	CP region range	Input
cp_pc_match	Code region match	Output
cp_regn_violt	CP permissions violation	Output

Specific function and register description:

1. According to core_pc and cprh / l comparison to determine which user is executing the code, resulting in cp_pc_match signal.

2. According to the result of comparing core_HADDR with cprh / l, a cp_addr_match signal is generated to determine the address accessed.

3. The acce_perm_violt signal is generated based on the current type of operation (acce_perm) and the processed signal (cp_uc_pc_match), by comparing the type of operation allowed in that region (cp_regn_perm).

4. The uc_regn_violt signal will be generated and output to the MPU status register module when the access address matches in this region and an access error occurs.

The same user's user code region (EFLASH region), the user code monopolize region (SRAM region) and CP region have the same privileges. That is, when the CP region is monopolized by the user, the CP is equivalent to the concept of owning a user, and the permissions of other regions to which the user himself can access also apply to the CP. The CP register is written by AHB and needs to be guaranteed to finally configure the CP control register. When valid = 0, the configuration information of CPRH, CPRL and CP control register (except valid bit) can be directly written into the register, and at the same time, the number of the user who writes the CP control register is cached. According to whether the region is valid, to determine whether the valid set 1.

FUNCTION SIMULATION AND RESULT ANALYSIS

The results for the CP region can be divided into the following parts: Verification of Region and Verification of Violation of Access Permissions.

The user can reasonably set the monopolize region of the shared code (the corresponding register is MPUCpRH and MPUCpRL), generally in the EFLASH region, the SRAM region and the ROM region. To facilitate the test, we will set it in the SRAM region. For the permission of the CP region, the cached user number (uid) and the valid configuration flag are assigned by the CP control register. For the above register, when the corresponding valid bit is invalid, it can be directly configured by the CPU; When the region is applied for monopolize use, that is valid, it cannot be modified unless it is validated by user first. Different users set their own permissions (MPUCpCR), different users access themselves and other user regions. Whether the corresponding user monopolize region is protected (that is, the information cannot be read and written by an illegal user), the corresponding error can be confirmed by querying the status in the MPU status register.

Verification of Region

The verification of the region verifies whether the CP region configuration meets the requirements and verifies whether the uid of the corresponding user can be correctly cached when the user occupies the CP region. It is verified whether other users operate in the CP region. Simulation results as shown in Fig 2: First, set the range of the CP region, 1. The user (u0, u1, u2, u3), respectively, to their monopolize and release;2.u0 monopolize CP region, other users cannot be released, only after the release of this user can be monopolized and used again by other users;3.u0 monopolize CP region, the user and other users to configure the CP region again, it will generate an error (invalid_e); 4. If the CP region is set in other protected regions have been configured, it is impossible to configure the success of only When the CP region configuration is reasonable and effective, can the user monopolize and use.

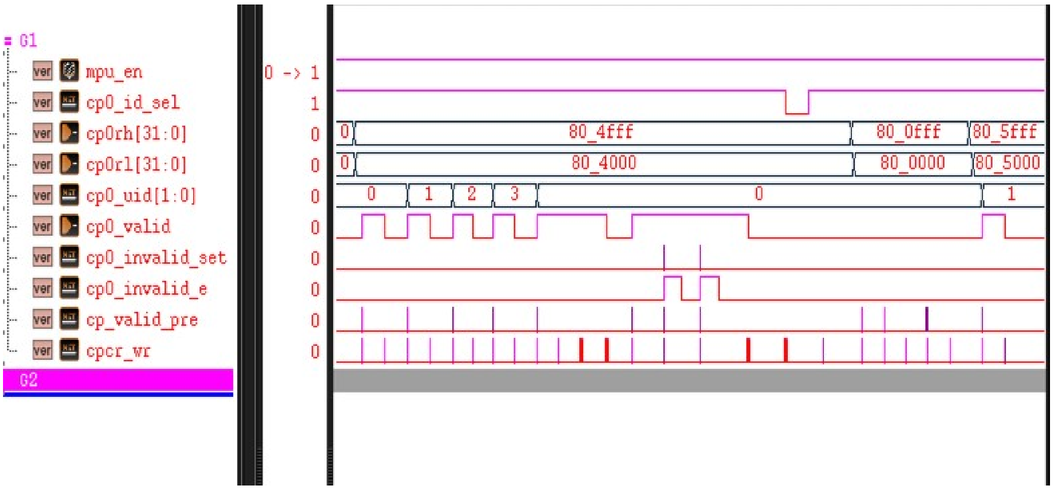

FIGURE 2. Verification of uid and invalid

Verification of Access Permissions

Verification of Violation of Access Permissions includes:1. The user verifies the access rights of the CP region; 2. After the user monopolizes the CP region, whether the rights of other regions are consistent with the user.

The simulation results are shown in the Fig 3: CP region permissions set to 0, that means this user and other users cannot read, write and execute (cp0_regn_perm = 0). After the user has exclusive possession of the CP region, a write operation is performed on the CP region, and a pw_e (the user writes error) is found, that is, a permission violation is returned to the MPU status register and the write operation is blocked, thereby achieving the role of protection.

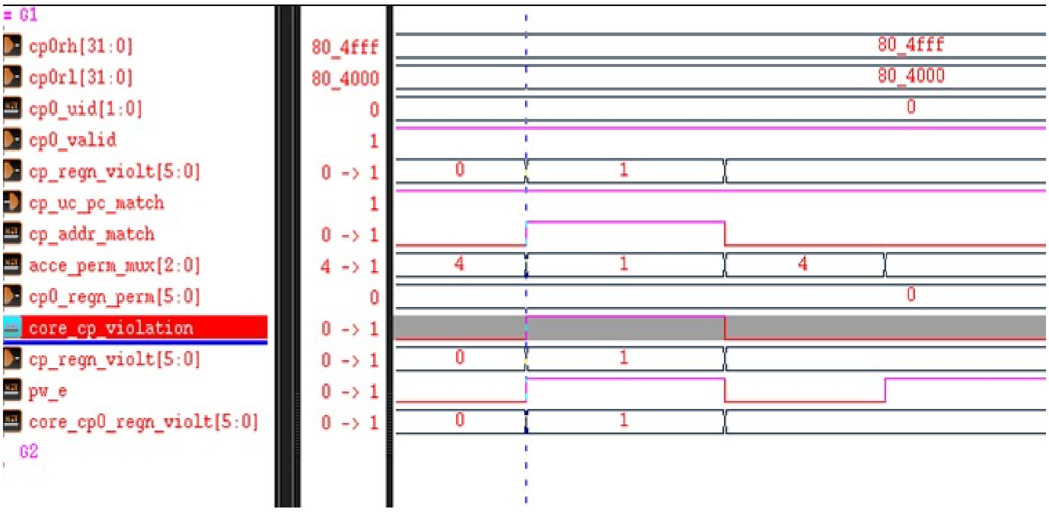

FIGURE 3. Verification of access rights in the CP region

After the user u1 exclusively occupies the CP region, the CP is equivalent to the user u1, and the permissions of the u1 is adaptive to the CP. For the SD1 region, when the user u2 monopolize it, and the permissions of this region is that other users cannot perform read and write operations. Therefore, when the CP accesses SD1 and reads and writes it, a permission violation error occurs and this error will be fed back to the status register. As a result, this operation will be prevented and the role of the protected region will be realized. It can be shown in Fig 4.

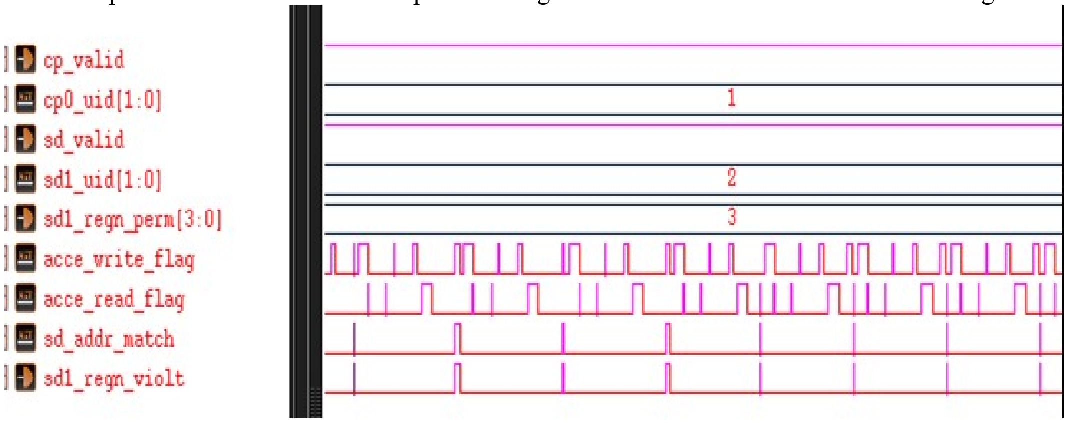

FIGURE 4. CP Read and write another user's SD region

Similarly, U3's own data region UD3, when the CP accesses the UD, an error of permission violation is also generated. Looking at Fig 5 you can find violation.

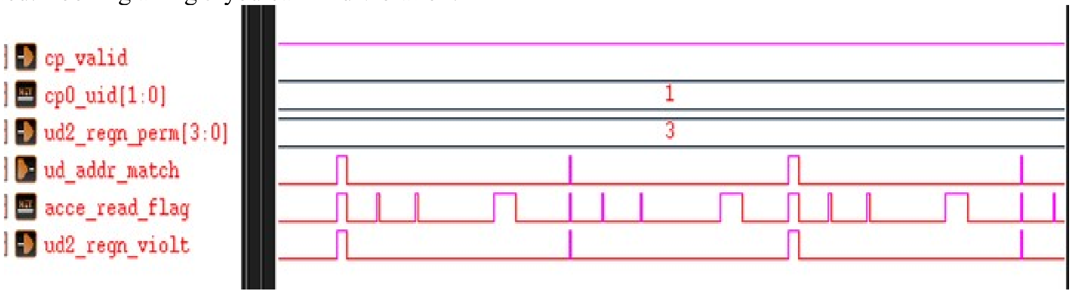

FIGURE 5. CP Read other user's UD regions

CONCLUSION

This paper mainly designs a shared code area (CP) based on the chip's internal MPU. And the designed module has been verified to get the expected result. To guarantee the ownership of system resources from hardware provides a good mechanism for resources not to be accessed illegally and effectively protects the internal memory of the chip. According to the current design ideas, some algorithms can be added in the future to make it more effective to protect the data.

ACKNOWLEDGMENTS

Authors:

Bo Qian (1993-), male, master, main research direction: digital circuit design and testing.

Weiping Jing (1954-), male, master, researcher, doctoral tutor, main research direction: digital-analog hybrid circuit design.

Bin Jiang (1973-), male, Ph.D., senior engineer, main research direction: high-performance embedded CPU design and its application.

Fund Project:

Intelligent hardware and software integrated security chip and its IP core - Ministry of Industry and Industry transformation and upgrading of the tender (0714-EMTC02-5378/7).

REFERENCES

1. Menon J, De Carli L, Thiruvengadam V, et al. Memory processing units[C]. Hot Chips 26 Symposium (HCS), 2014 IEEE. IEEE, 2014: 1-1.
2. Guan L, Lin J, Luo B, et al. protecting private keys against memory disclosure attacks using hardware transactional memory[C]. Security and Privacy (SP), 2015 IEEE Symposium on. IEEE, 2015: 3-19.
3. Malliaros S, Ntantogian C, Xenakis C. Protecting sensitive information in the volatile memory from disclosure attacks[C]. Availability, Reliability and Security (ARES), 2016 11th International Conference on. IEEE, 2016: 687-693.
4. Ko Y, Lee K. Multi-level cache vulnerability estimation: The first step to protect memory[C]. Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on. IEEE, 2016: 001165-001170.
5. Stecklina O, Langendoerfer P, Menzel H. Design of a tailor-made Memory Protection Unit for Low Power Microcontrollers[C]. IEEE International Symposium on Industrial Embedded Systems. IEEE, 2013:225-231.