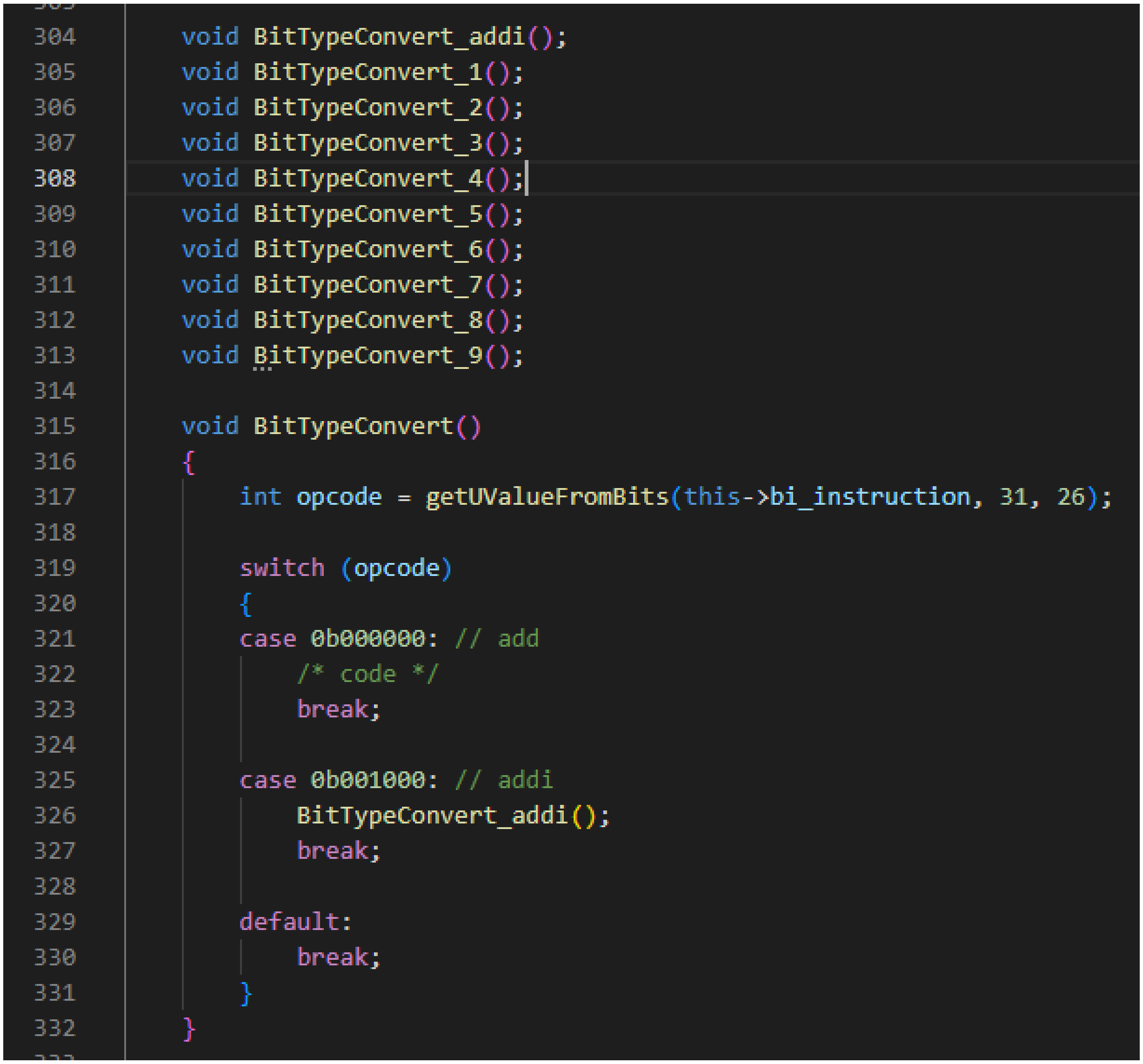
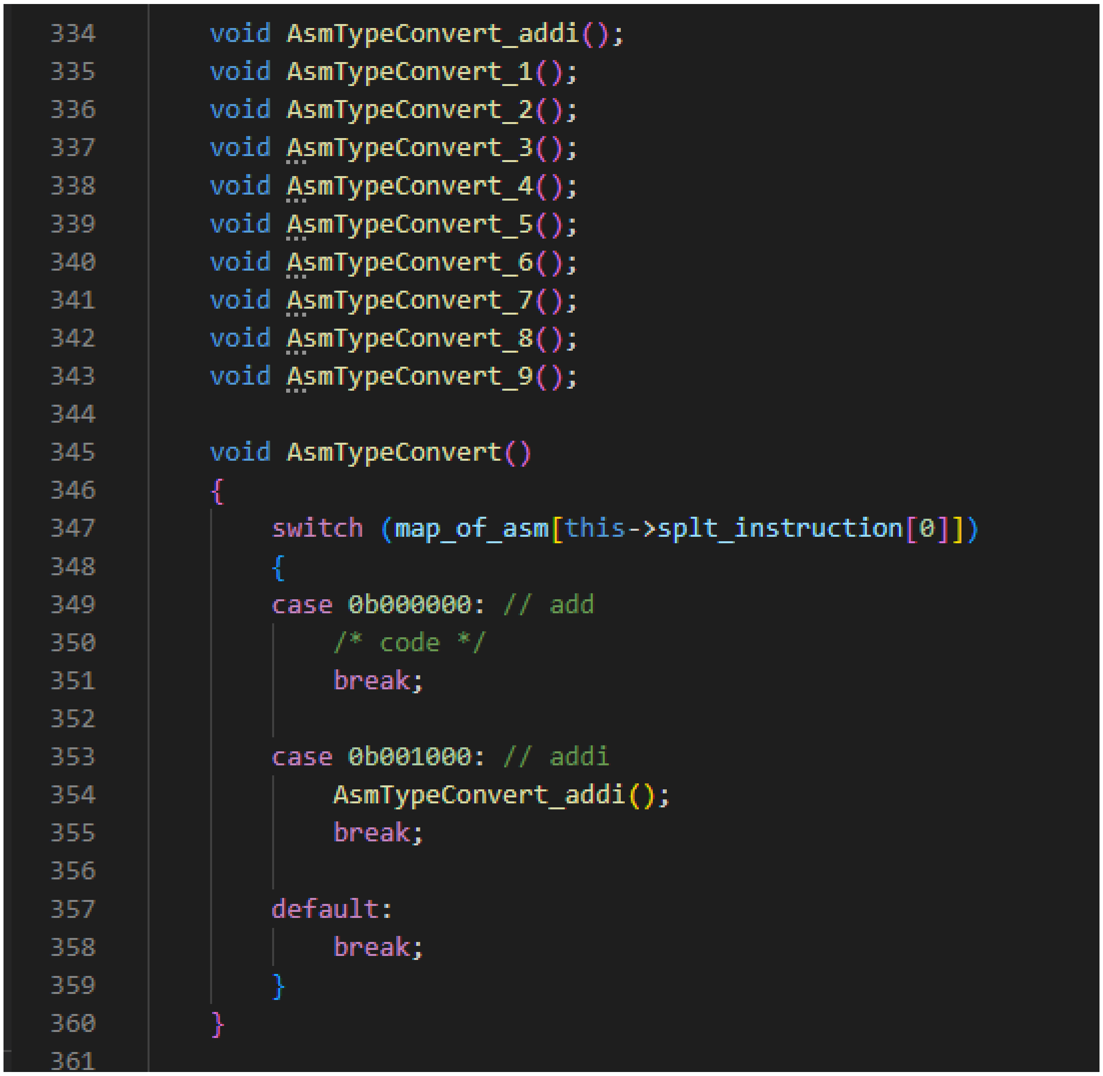
需要做的事情：

1、补全函数

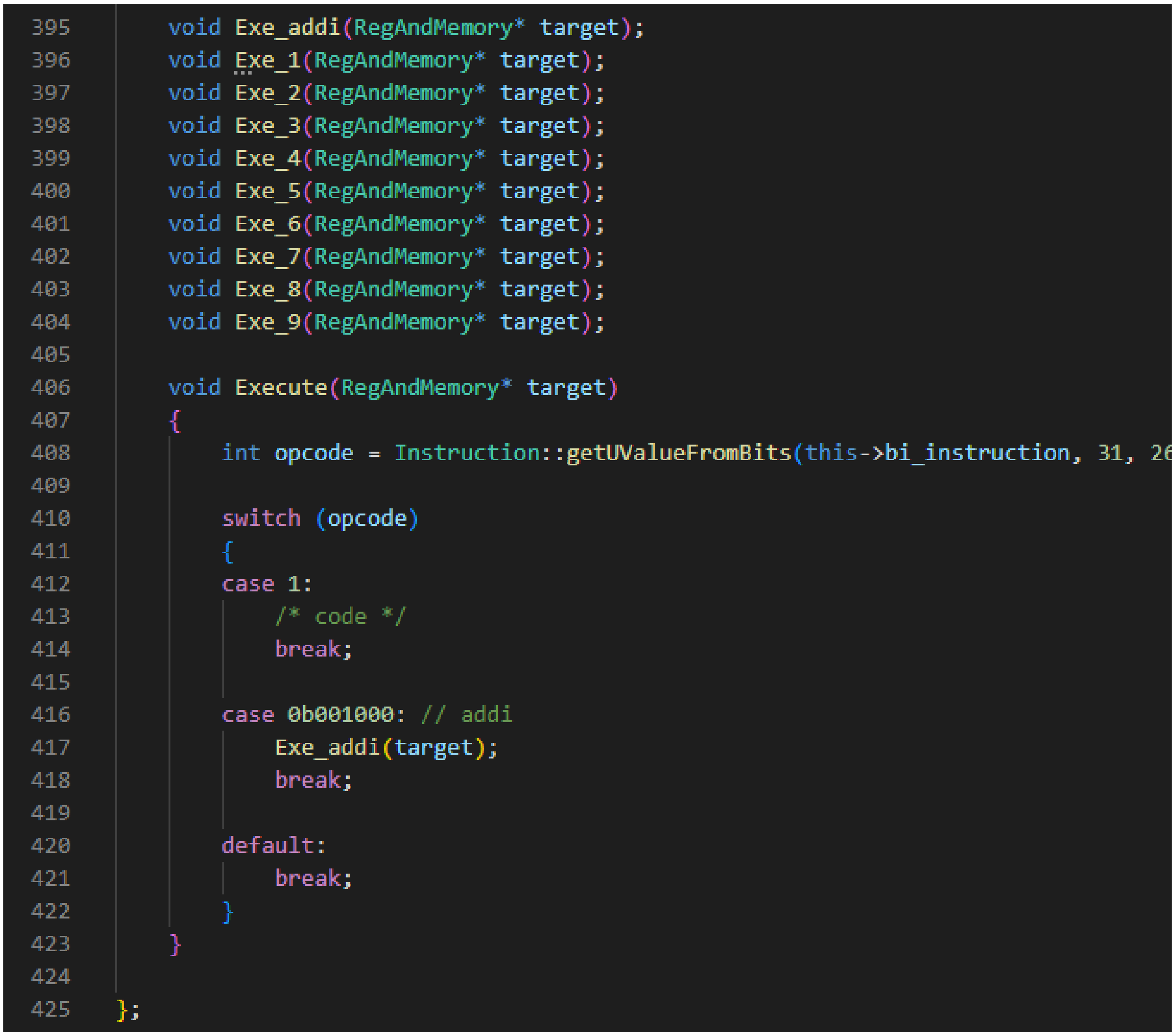
包括函数以及母函数中的接口



```
    void BitTypeConvert_addi();
    void BitTypeConvert_1();
    void BitTypeConvert_2();
    void BitTypeConvert_3();
    void BitTypeConvert_4();
    void BitTypeConvert_5();
    void BitTypeConvert_6();
    void BitTypeConvert_7();
    void BitTypeConvert_8();
    void BitTypeConvert_9();

    void BitTypeConvert()
    {
        int opcode = getUValueFromBits(this->bi_instruction, 31, 26);

        switch (opcode)
        {
        case 0b000000: // add
            /* code */
            break;

        case 0b001000: // addi
            BitTypeConvert_addi();
            break;

        default:
            break;
        }
    }
```

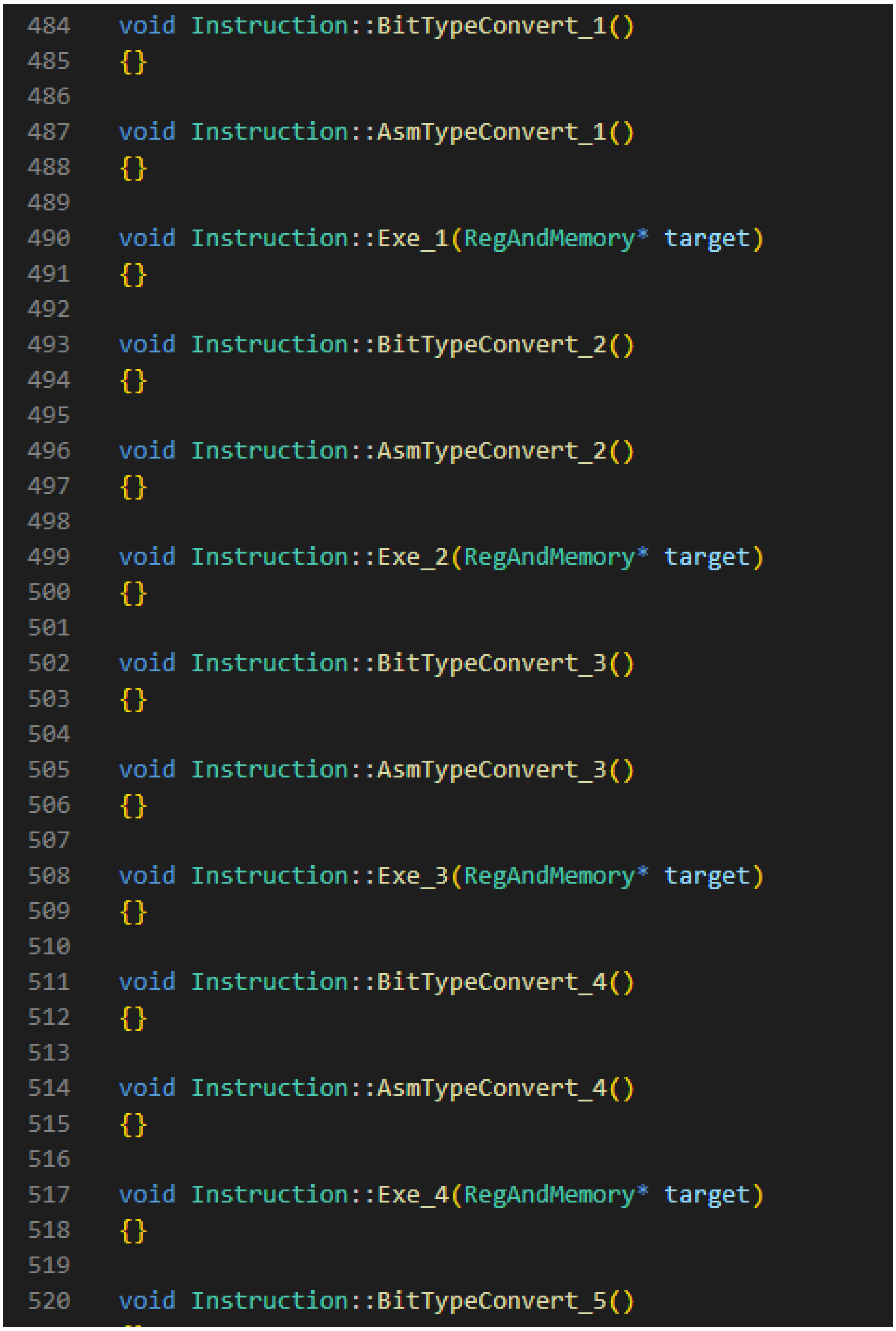


```
    void AsmTypeConvert_addi();
    void AsmTypeConvert_1();
    void AsmTypeConvert_2();
    void AsmTypeConvert_3();
    void AsmTypeConvert_4();
    void AsmTypeConvert_5();
    void AsmTypeConvert_6();
    void AsmTypeConvert_7();
    void AsmTypeConvert_8();
    void AsmTypeConvert_9();

    void AsmTypeConvert()
    {
        switch (map_of_asm[this->splt_instruction[0]])
        {
        case 0b000000: // add
            /* code */
            break;

        case 0b001000: // addi
            AsmTypeConvert_addi();
            break;

        default:
            break;
        }
    }
```



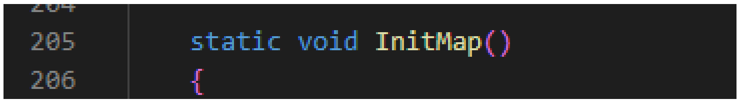
```
    void Exe_addi(RegAndMemory* target);
    void Exe_1(RegAndMemory* target);
    void Exe_2(RegAndMemory* target);
    void Exe_3(RegAndMemory* target);
    void Exe_4(RegAndMemory* target);
    void Exe_5(RegAndMemory* target);
    void Exe_6(RegAndMemory* target);
    void Exe_7(RegAndMemory* target);
    void Exe_8(RegAndMemory* target);
    void Exe_9(RegAndMemory* target);

    void Execute(RegAndMemory* target)
    {
        int opcode = Instruction::getUValueFromBits(this->bi_instruction, 31, 26

        switch (opcode)
        {
        case 1:
            /* code */
            break;

        case 0b001000: // addi
            Exe_addi(target);
            break;

        default:
            break;
        }
    }

};
```

具体补全补全在下面的实现

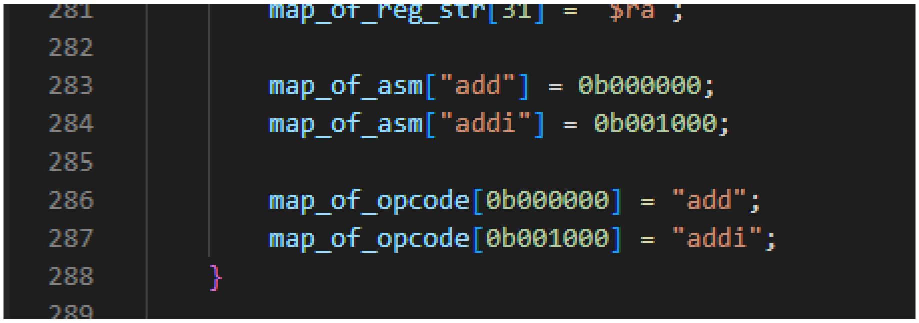


```
void Instruction::BitTypeConvert_1()
{}

void Instruction::AsmTypeConvert_1()
{}

void Instruction::Exe_1(RegAndMemory* target)
{}

void Instruction::BitTypeConvert_2()
{}

void Instruction::AsmTypeConvert_2()
{}

void Instruction::Exe_2(RegAndMemory* target)
{}

void Instruction::BitTypeConvert_3()
{}

void Instruction::AsmTypeConvert_3()
{}

void Instruction::Exe_3(RegAndMemory* target)
{}

void Instruction::BitTypeConvert_4()
{}

void Instruction::AsmTypeConvert_4()
{}

void Instruction::Exe_4(RegAndMemory* target)
{}

void Instruction::BitTypeConvert_5()
```

2、InitMap函数中加上对应指令的操作码



```
    static void InitMap()
    {
```



```
        map_of_reg_str[31] = "$ra";

        map_of_asm["add"] = 0b000000;
        map_of_asm["addi"] = 0b001000;

        map_of_opcode[0b000000] = "add";
        map_of_opcode[0b001000] = "addi";
    }
```

寄存器已经加好了，不用加了！

3、编写和调试请参考现有代码，getValueFromBits该函数还有bug，addi的对应操作没有考虑负数的情况，写的时候请务必注意！！！