마이크로프로세서

- Decode 구현 실습-

Daejin Park

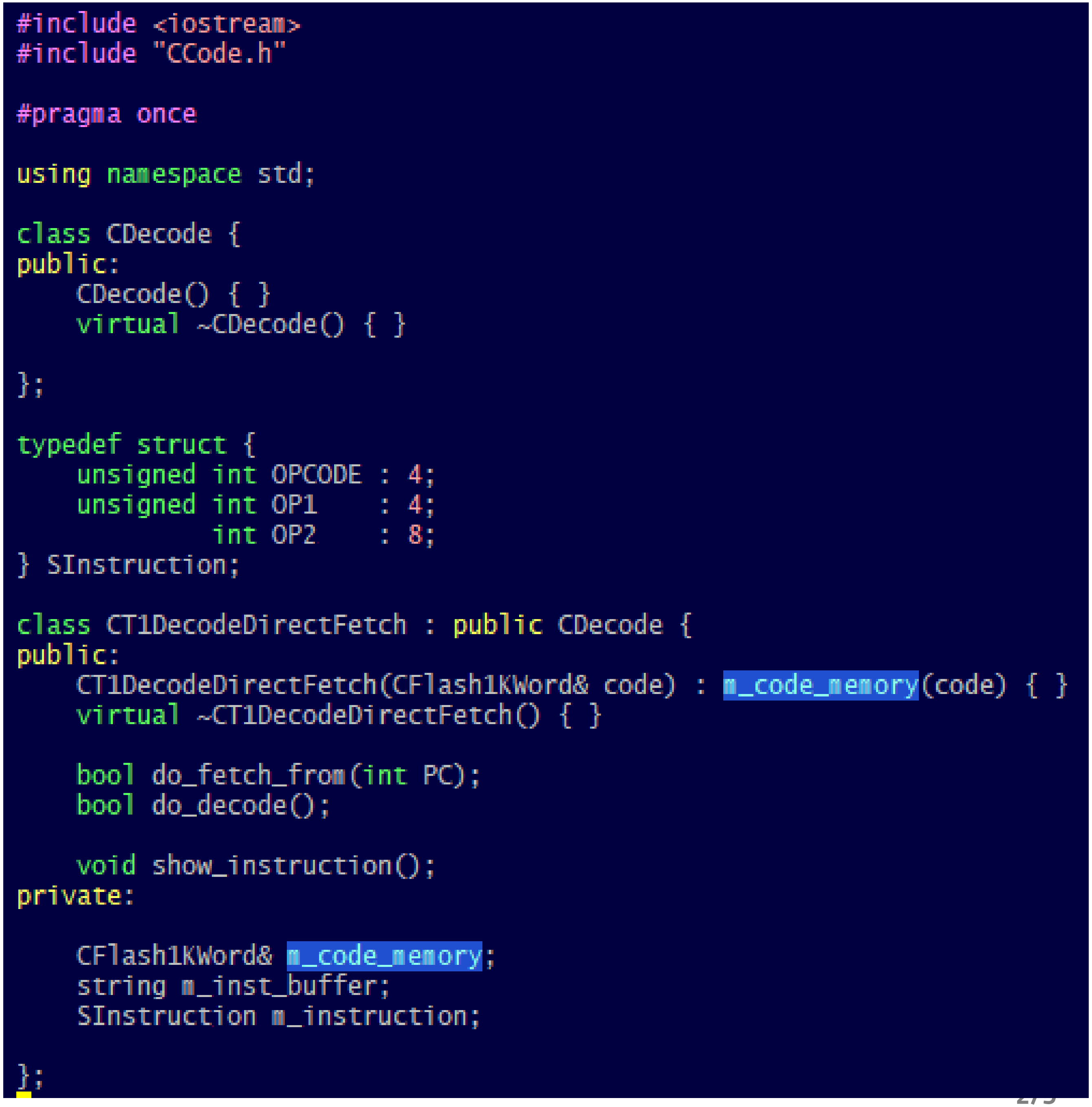
School of Electronics Engineering, KNU, KOREA

2019.03.22

ACES Lab.

CDecode.h

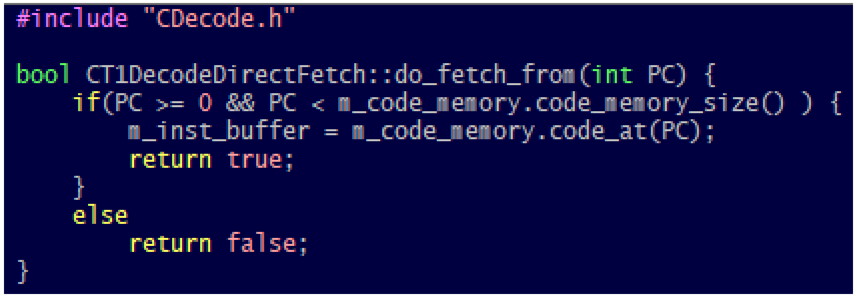
CT1DecodeDirectFetch (CDecode) 작성해두기



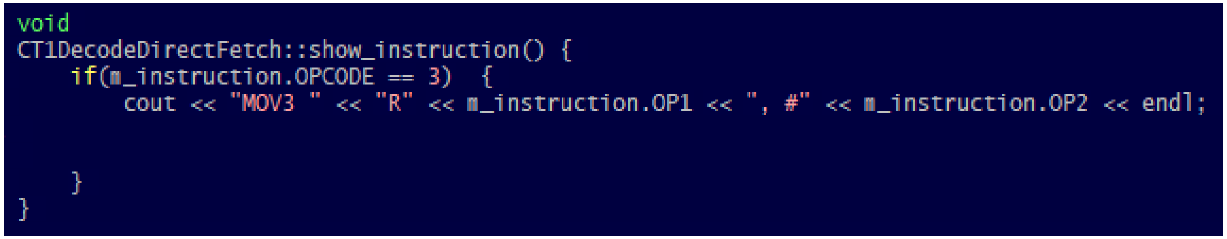
```cpp
#include <iostream>
#include "CCode.h"

#pragma once

using namespace std;

class CDecode {
public:
    CDecode() { }
    virtual ~CDecode() { }

};

typedef struct {
    unsigned int OPCODE : 4;
    unsigned int OP1    : 4;
             int OP2    : 8;
} SInstruction;

class CT1DecodeDirectFetch : public CDecode {
public:
    CT1DecodeDirectFetch(CFlash1KWord& code) : m_code_memory(code) { }
    virtual ~CT1DecodeDirectFetch() { }

    bool do_fetch_from(int PC);
    bool do_decode();

    void show_instruction();
private:

    CFlash1KWord& m_code_memory;
    string m_inst_buffer;
    SInstruction m_instruction;

};
```

CDecode.cpp

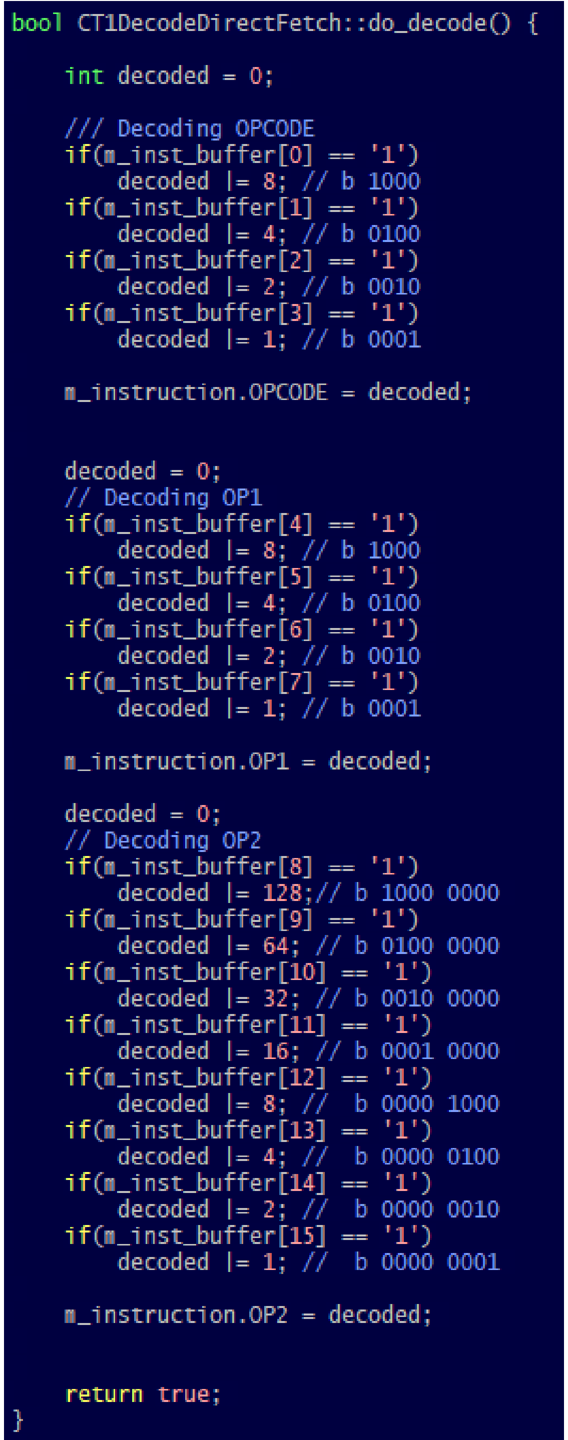
Member function 3개 구현



```cpp
#include "CDecode.h"

bool CT1DecodeDirectFetch::do_fetch_from(int PC) {
    if(PC >= 0 && PC < m_code_memory.code_memory_size() ) {
        m_inst_buffer = m_code_memory.code_at(PC);
        return true;
    }
    else
        return false;
}
```



```cpp
void
CT1DecodeDirectFetch::show_instruction() {
    if(m_instruction.OPCODE == 3)  {
        cout << "MOV3 " << "R" << m_instruction.OP1 << ", #" << m_instruction.OP2 << endl;


    }
}
```



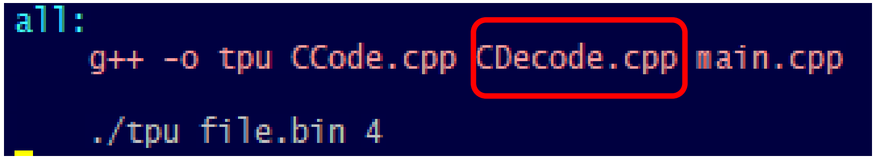
```cpp
bool CT1DecodeDirectFetch::do_decode() {

    int decoded = 0;

    /// Decoding OPCODE
    if(m_inst_buffer[0] == '1')
        decoded |= 8; // b 1000
    if(m_inst_buffer[1] == '1')
        decoded |= 4; // b 0100
    if(m_inst_buffer[2] == '1')
        decoded |= 2; // b 0010
    if(m_inst_buffer[3] == '1')
        decoded |= 1; // b 0001

    m_instruction.OPCODE = decoded;


    decoded = 0;
    // Decoding OP1
    if(m_inst_buffer[4] == '1')
        decoded |= 8; // b 1000
    if(m_inst_buffer[5] == '1')
        decoded |= 4; // b 0100
    if(m_inst_buffer[6] == '1')
        decoded |= 2; // b 0010
    if(m_inst_buffer[7] == '1')
        decoded |= 1; // b 0001

    m_instruction.OP1 = decoded;

    decoded = 0;
    // Decoding OP2
    if(m_inst_buffer[8] == '1')
        decoded |= 128;// b 1000 0000
    if(m_inst_buffer[9] == '1')
        decoded |= 64; // b 0100 0000
    if(m_inst_buffer[10] == '1')
        decoded |= 32; // b 0010 0000
    if(m_inst_buffer[11] == '1')
        decoded |= 16; // b 0001 0000
    if(m_inst_buffer[12] == '1')
        decoded |= 8; //  b 0000 1000
    if(m_inst_buffer[13] == '1')
        decoded |= 4; //  b 0000 0100
    if(m_inst_buffer[14] == '1')
        decoded |= 2; //  b 0000 0010
    if(m_inst_buffer[15] == '1')
        decoded |= 1; //  b 0000 0001

    m_instruction.OP2 = decoded;


    return true;
}
```

C++로 프로세서 모델 구현

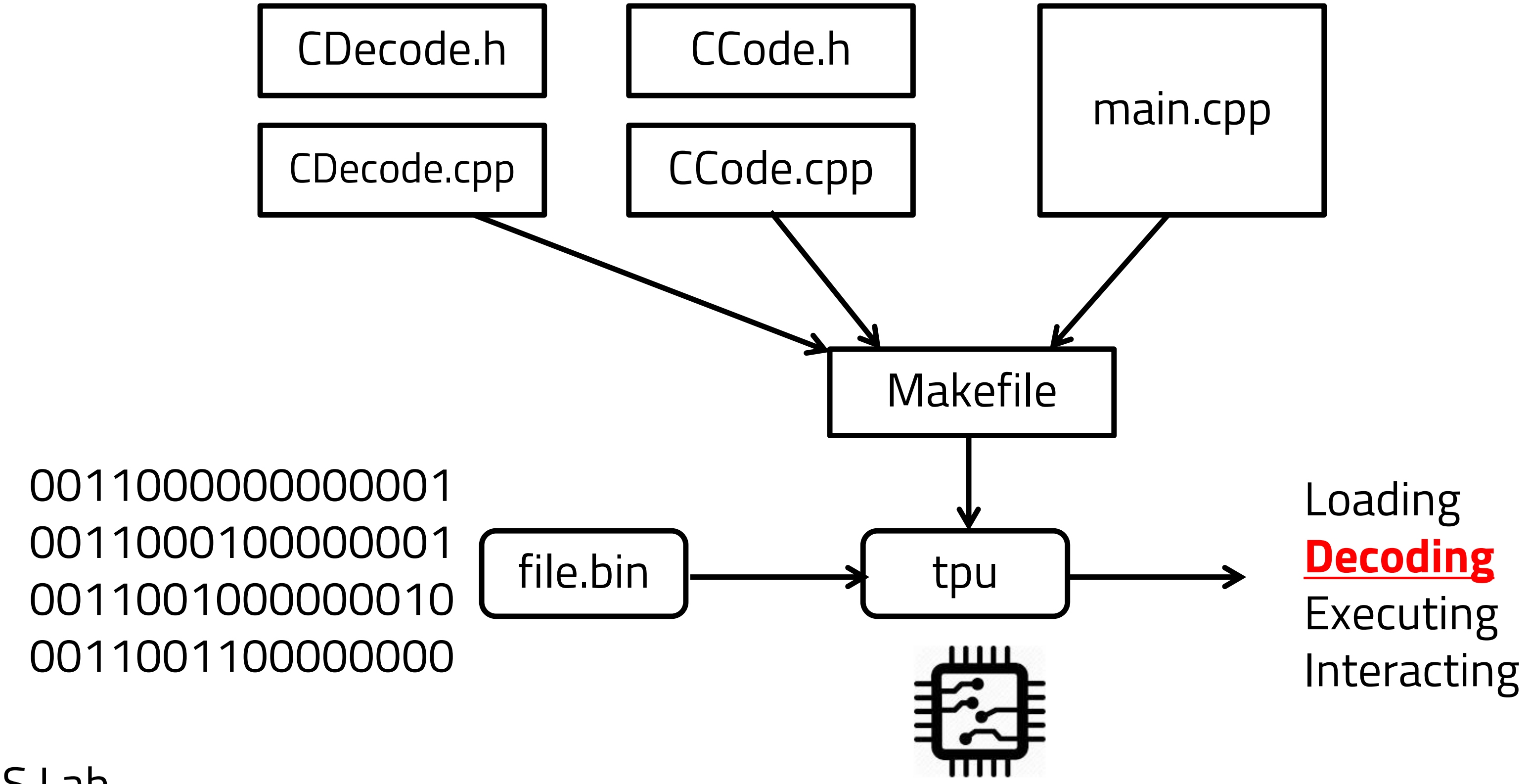
- 터미널에서 다음을 실행

  make all

Makefile 추가



```
all:
	g++ -o tpu CCode.cpp CDecode.cpp main.cpp
	./tpu file.bin 4
```



CDecode.h

CDecode.cpp

CCode.h

CCode.cpp

main.cpp

Makefile

0011000000000001
0011000100000001
0011001000000010
0011001100000000

file.bin

tpu

Loading
Decoding
Executing
Interacting

Q & A

Thank you for your attention

Architecture and Compiler
for Embedded Systems Lab.

School of Electronics Engineering, KNU

ACES Lab (boltanut@knu.ac.kr)