마이크로프로세서

- SW 해석 시작 : Decode -

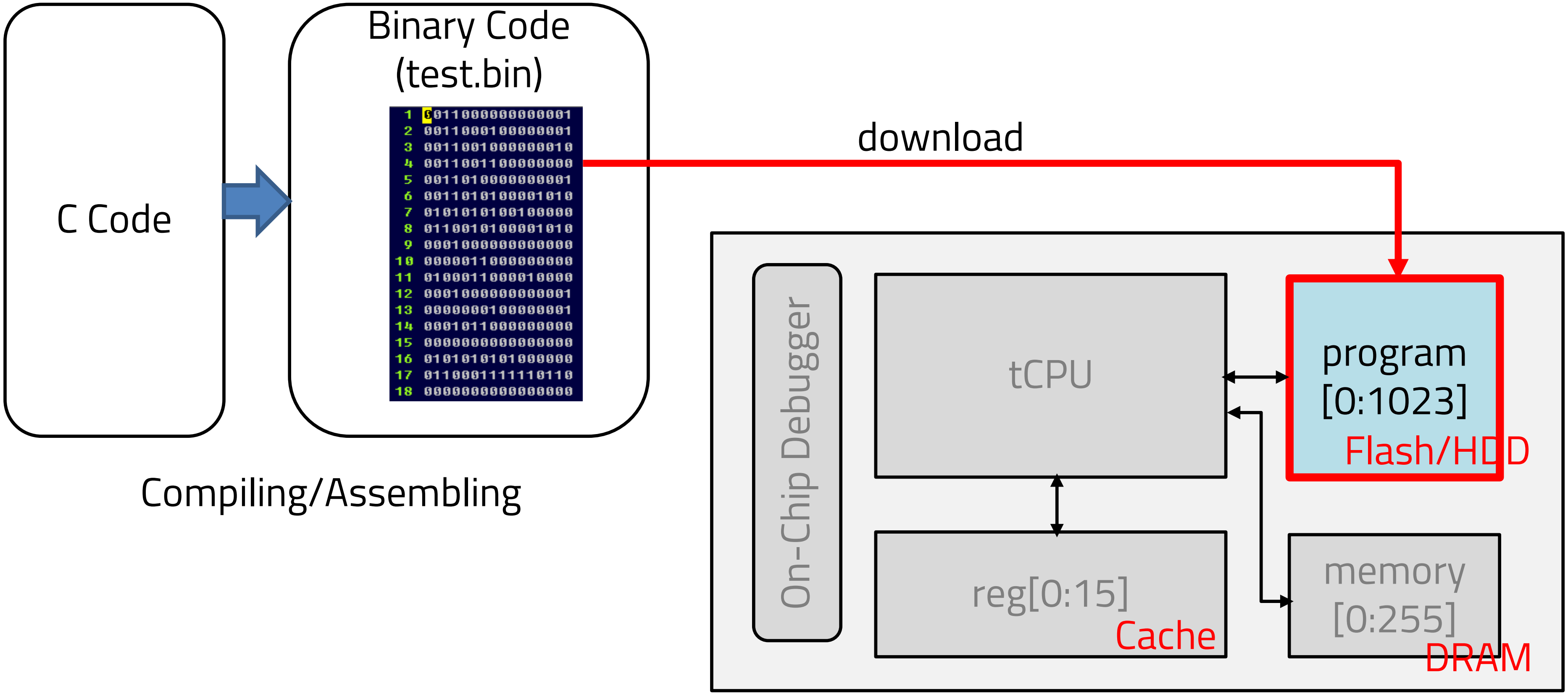
Daejin Park

School of Electronics Engineering, KNU, KOREA

2019.03.20

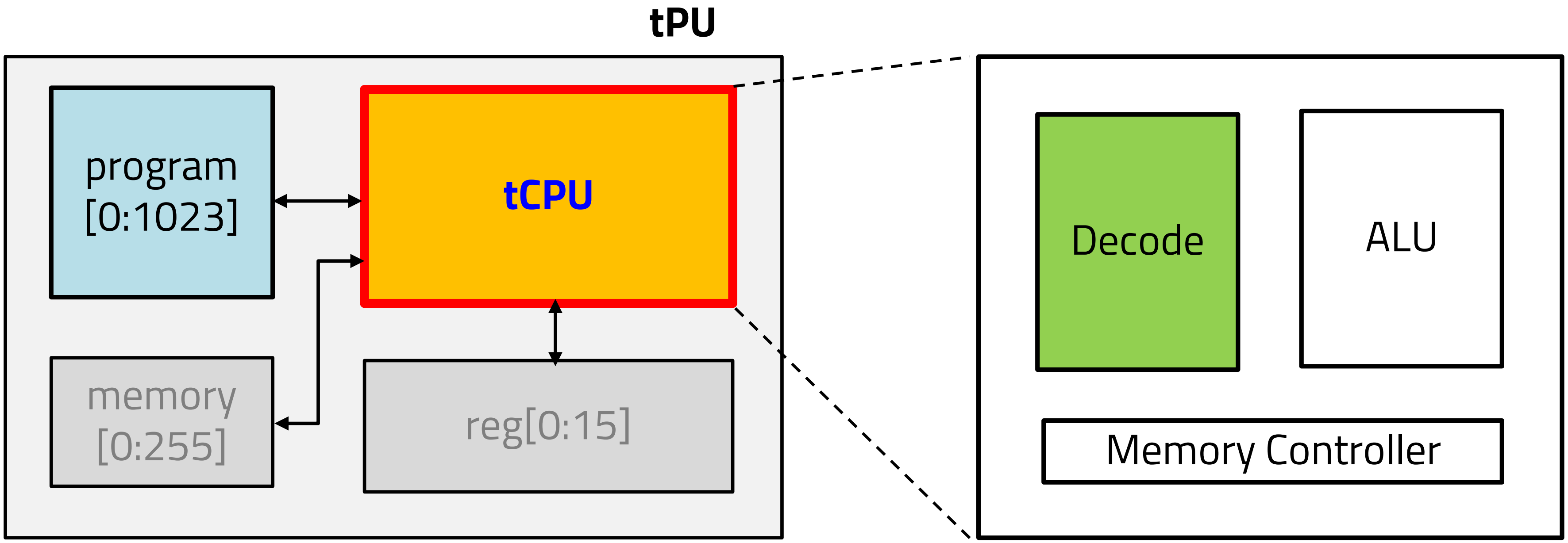
ACES Lab.

tPU Code Memory



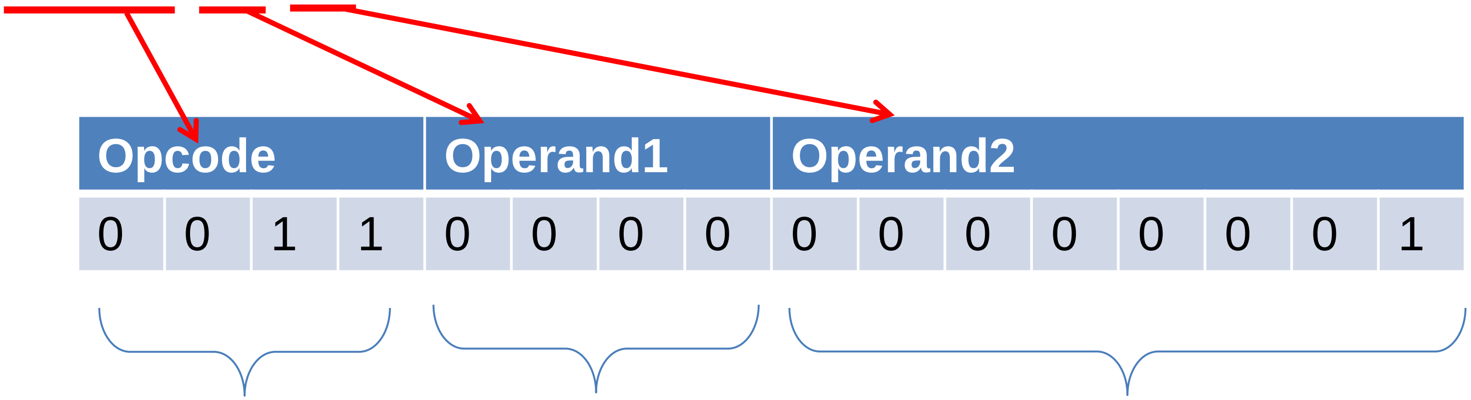
```
0011000000000001
0011000100000001
0011001000000010
0011001100000000
0011010000000001
0011010100001010
0101010100100000
0110010100001010
0001000000000000
0000011000000000
0100011000010000
0001000000000001
0000000100000001
0001011000000000
0000000000000000
0101010101000000
0110001111110110
0000000000000000
```

tPU Decode 유닛



tPU ISA (Instruction Set Architecture)

- MOV3 R0, #1



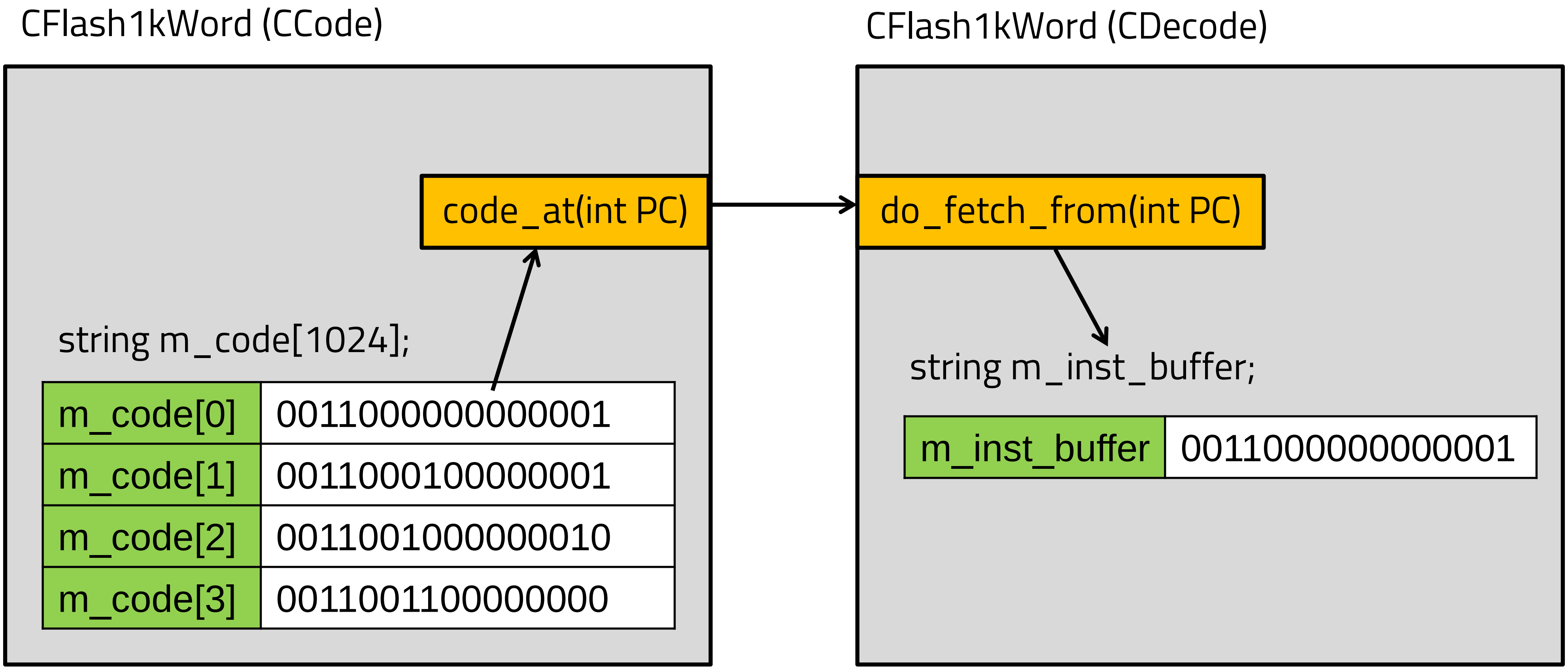
Opcode				Operand1				Operand2							
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1

명령어 종류 16가지
(최대)

16개 레지스터
지정가능

0~255 값 표현 가능
(주소 또는 데이터)

Fetch Instruction from Code Memory



CFlash1kWord (CCode)

code_at(int PC)

string m_code[1024];

m_code[0]	0011000000000001
m_code[1]	0011000100000001
m_code[2]	0011001000000010
m_code[3]	0011001100000000

CFlash1kWord (CDecode)

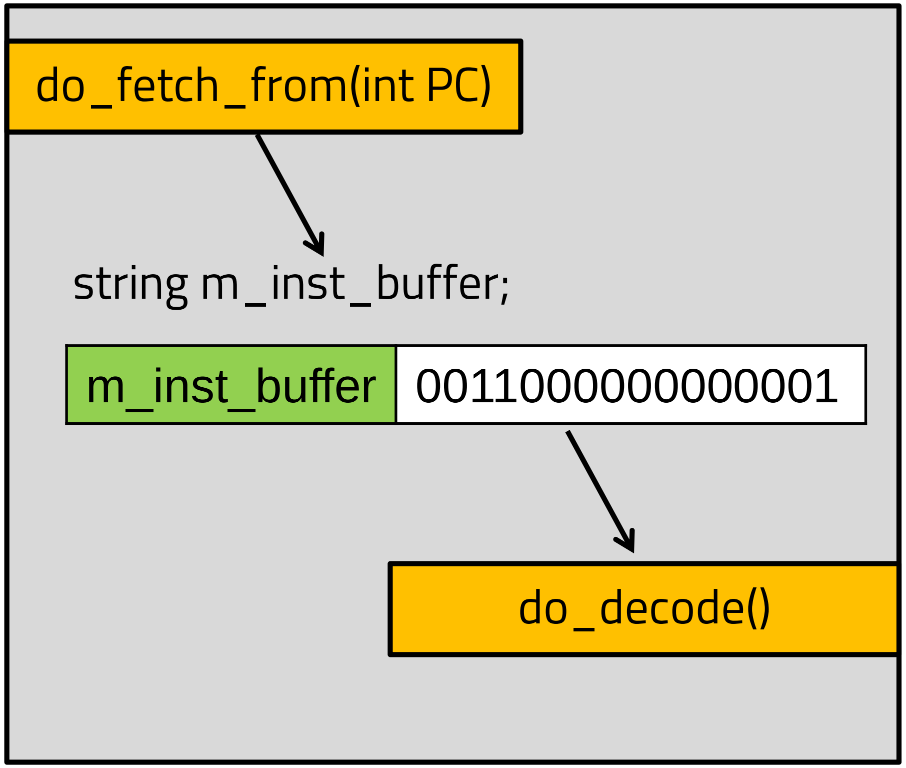
do_fetch_from(int PC)

string m_inst_buffer;

m_inst_buffer	0011000000000001

Decoding Instruction

CFlash1kWord (CDecode)



do_fetch_from(int PC)

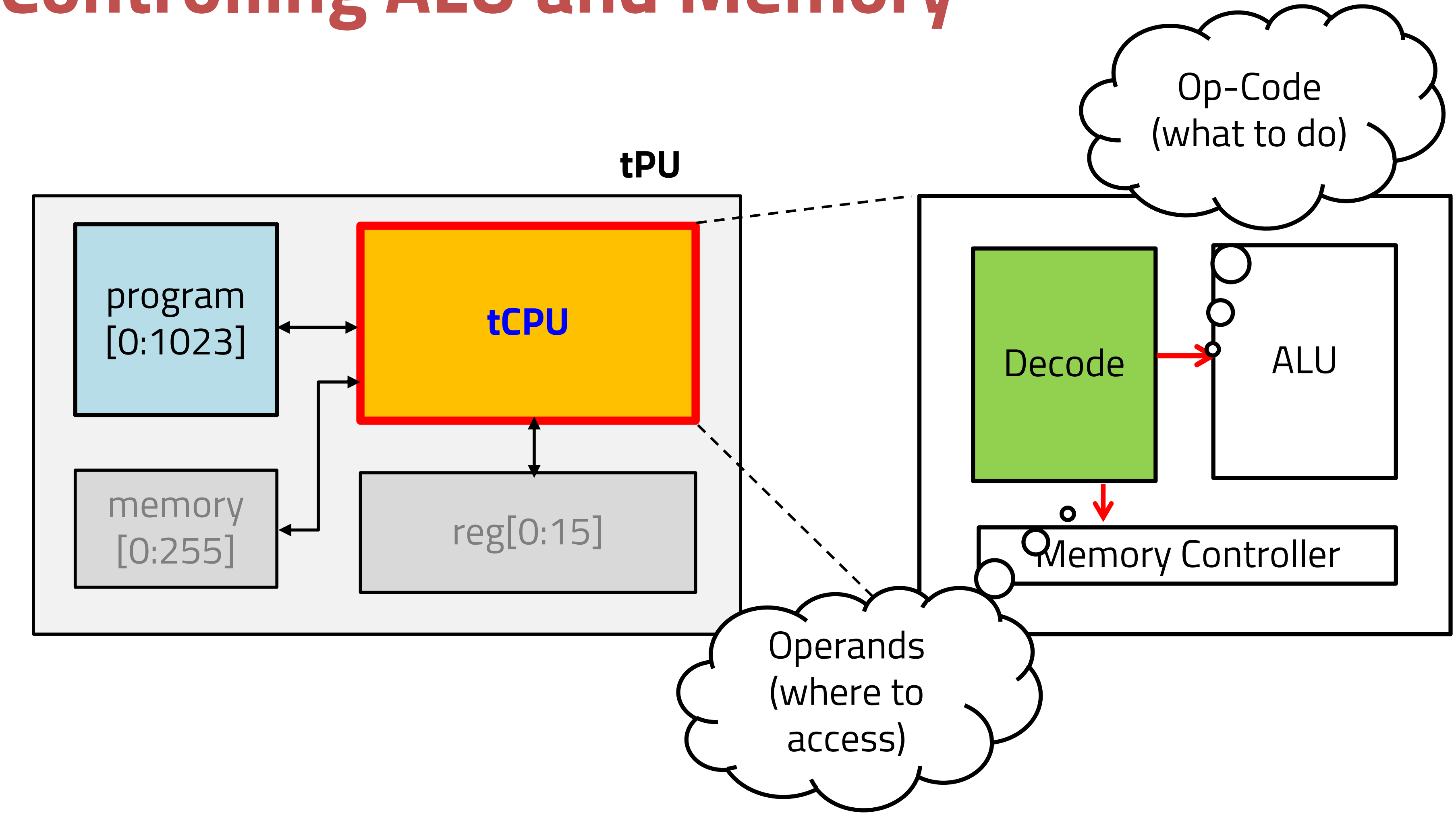
string m_inst_buffer;

m_inst_buffer	0011000000000001

do_decode()

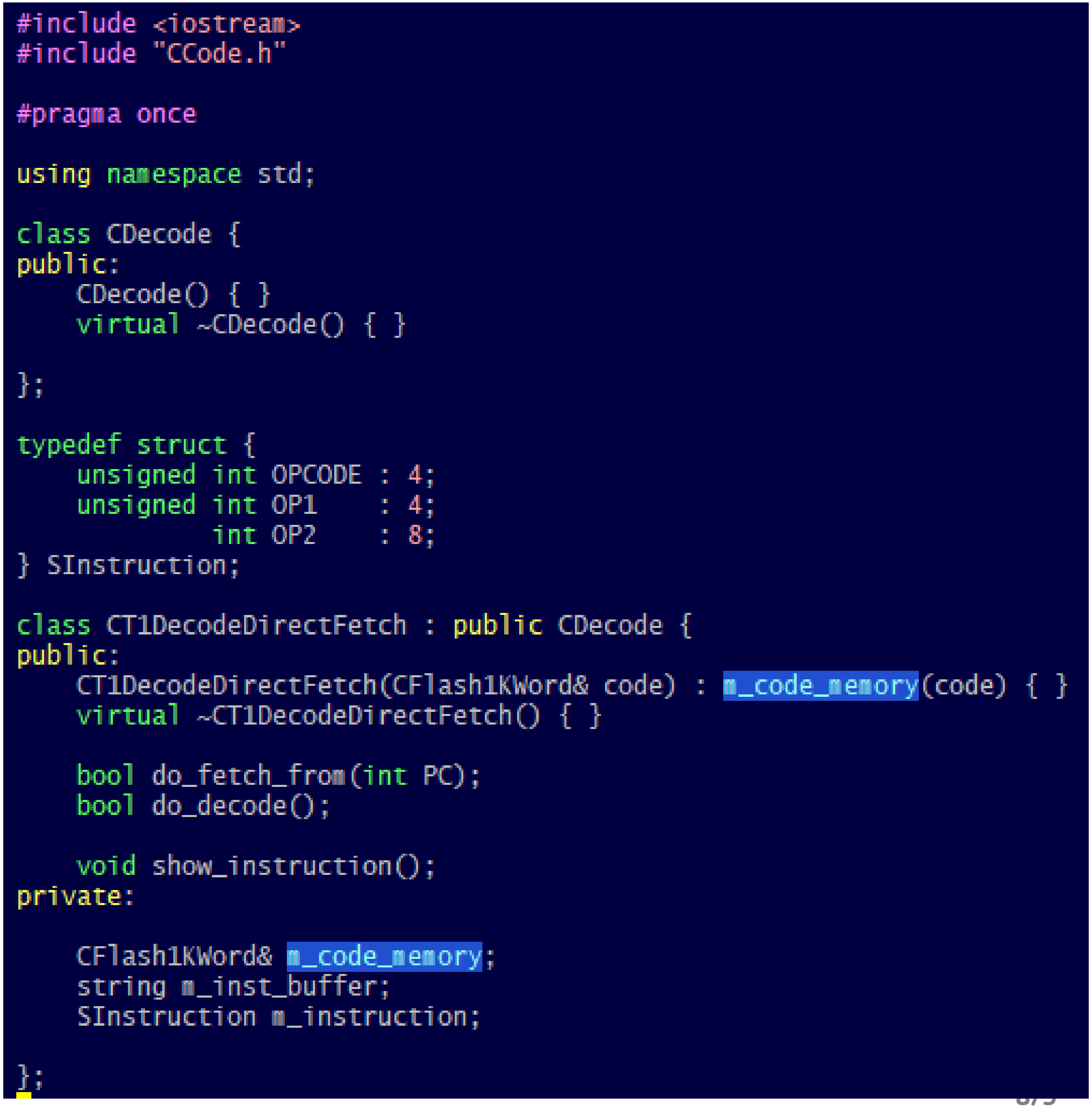
```
bool CT1DecodeDirectFetch::do_decode()
{

    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;


   .....
```

Controlling ALU and Memory



숙제

CT1DecodeDirectFetch (CDecode) 작성해두기



```
#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;

};
```

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)