어셈블리프로그램 설계 및 실습

Lab #1 MDK-ARM Setup & Basic Example

Teaching assistants

Jiwoon Lee jwlee@linux.com

Jeongwon Hwang jeong202192@kw.ac.kr

KWANGWOON UNIVERSITY

Introduction

1. Instructor

- 이형근 교수님 hklee@kw.ac.kr

2. Teaching Assistants

- 이지운 jwlee@linux.com
- 황정원 jeong202192@kw.ac.kr

Course Outline

Week	Lecture Description	Lab Description
1	Introduction to assembly programming Instruction Set Architecture	
2	Introduction to ARM processor Introduction to ARM assembly program	Install ARM software development tool Basic examples
3	ARM instruction set	Data transfer from/to memories
4	ARM instruction set - data transfer	Control flow & Data processing
5	ARM instruction set - control flow & data processing Loop examples	Second operand & Multiplication
6	ARM instruction set - format and second operand	Subroutine calls
7	ARM instruction set - Block data transfer & stacks Subroutine calls	Floating point number & addition
8	중간고사	
9	Floating point number & addition	
10	Multiplication of constants & floating-point numbers	
11	ARM instruction set - Pseudo instructions ARM Assembly Programming Performance Issues	
12	Project proposal (volunteers only)	
13	Project Q&A	
14	Project presentation (volunteers only)	
15	기말고사	

Grade Information

- Midterm exam : 20%
- Final exam: 20%
- Assignments : 30%
  - 실습 수업 진행 후 보고서를 통해 채점 수행
- Term project : 20%
  - 설계과제는 제안서, 결과보고서를 통하여 채점을 수행
- Attendance : 10%

Lecture overview

- Online lecture
  - 이론에 대한 강의
- Q&A
  - 인터넷 강의 중 질문할 사항에 대해서 Q&A 시간을 가짐
- Practice
  - 인터넷 강의와 Q&A 시간에 배운 점을 활용하여 실습 진행
  - 결과값과 성능 (cycles, memory 등) 를 확인

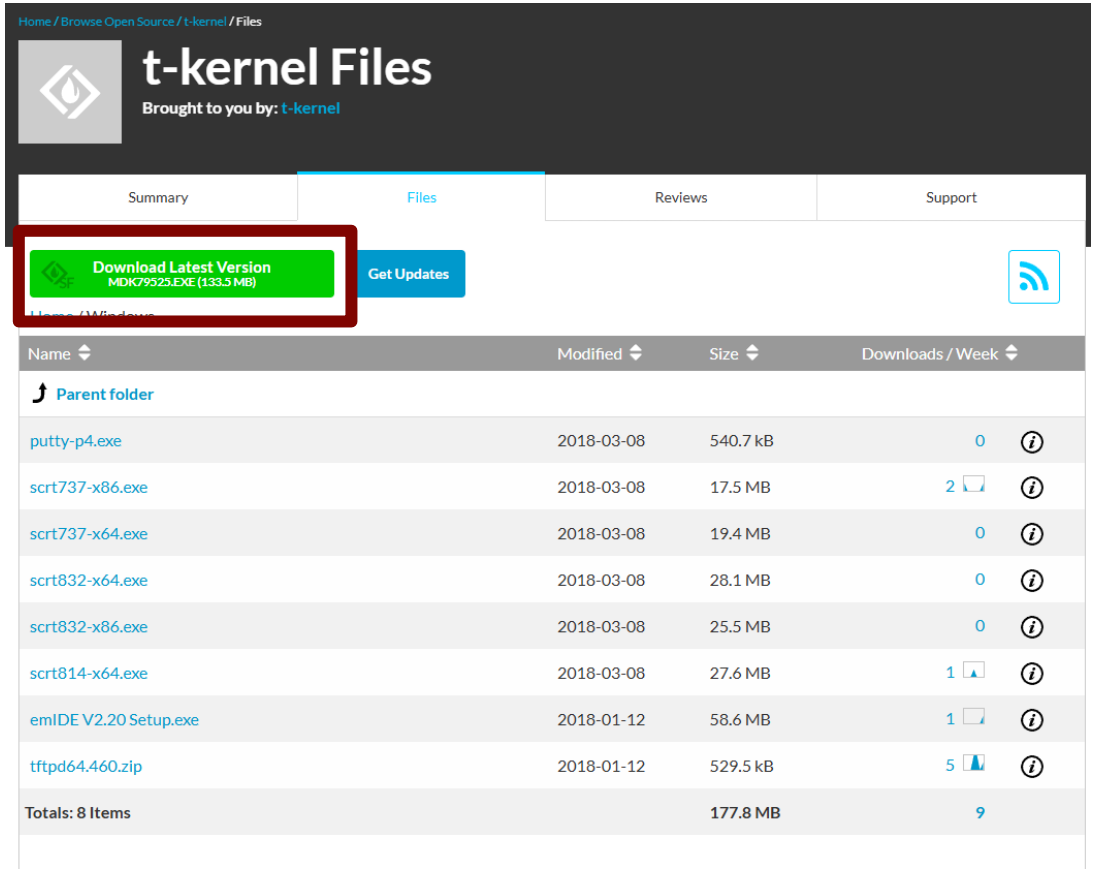
Download (1/2)

- Download MDK 5.29 with below link (MDK529.EXE, RECOMMENDED)
  - http://armkeil.blob.core.windows.net/eval/MDK529.EXE

- You can also download the latest version (Not recommended)
  - Keil Downloads
  - But they don't provide compiler version 5 since 5.37
  - And also, there's a problem in license since 5.26
  - So, you need to install compiler separated.

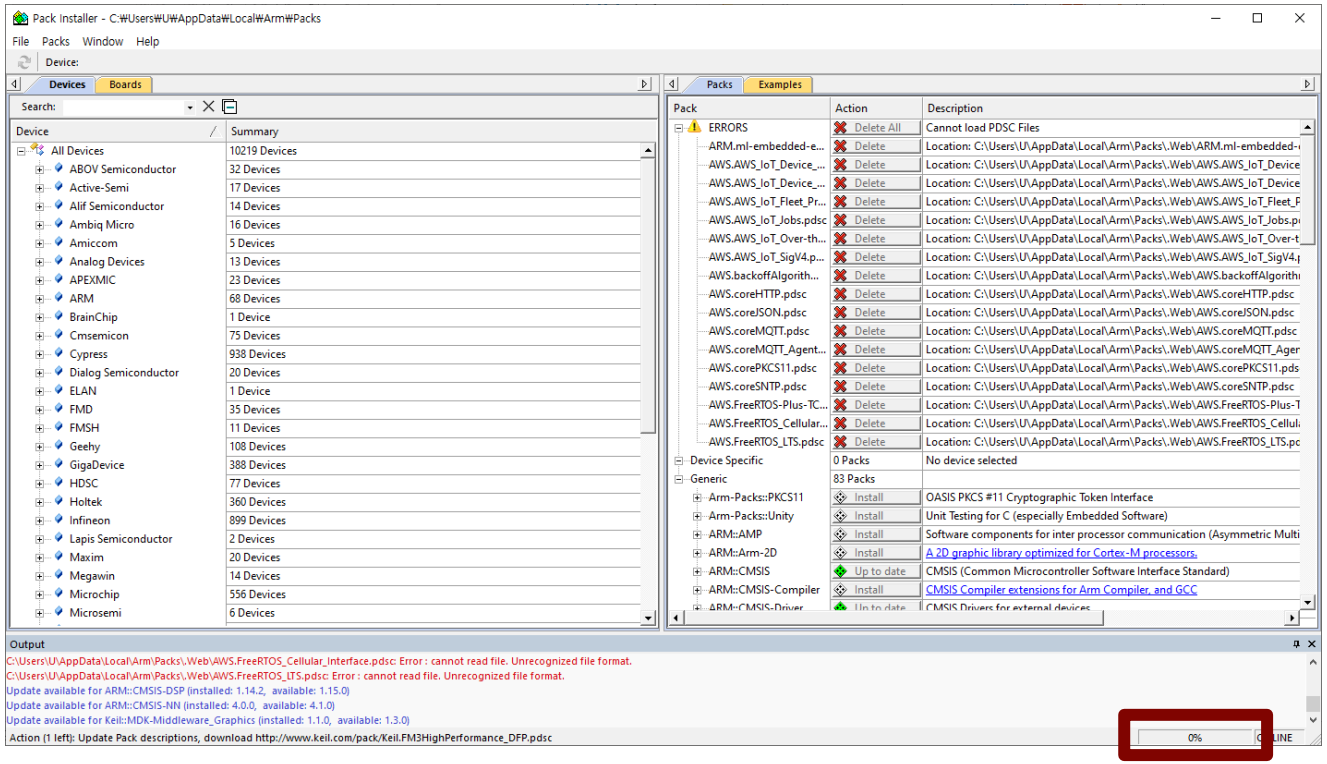
Download (2/2)

- Download MDK Version 4 Legacy Pack (MDK79525.EXE)
  - [t-kernel - Browse /Windows at SourceForge.net]



Setup (1/2)

- Execute “MDK529.EXE”
  - Click “Next” button until starting setup
- When installation is completed, a window like the one in the picture below will appear.



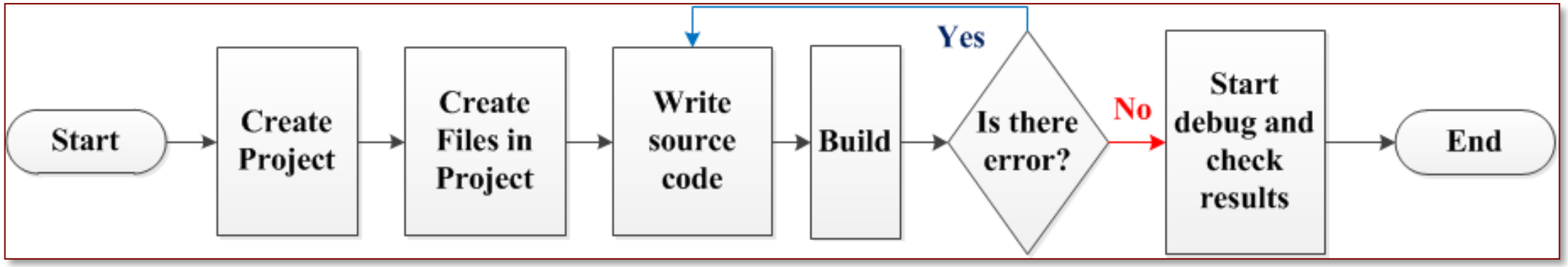
- After the update has completed, close the window.

Setup (2/2)

- **Execute “MDK79525.EXE”**
  - Click “Next” button until starting setup

Assembly Programming with Keil uVision

- Workflow

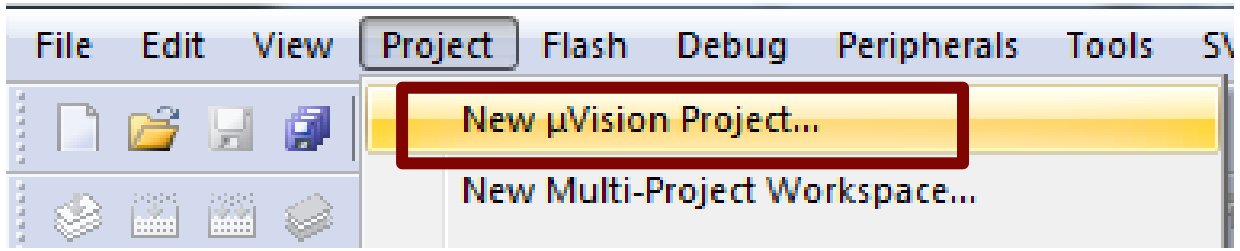


- Create Project
- Create Files
- Write source code
- Build
- Debug

Project Creation (1/5)

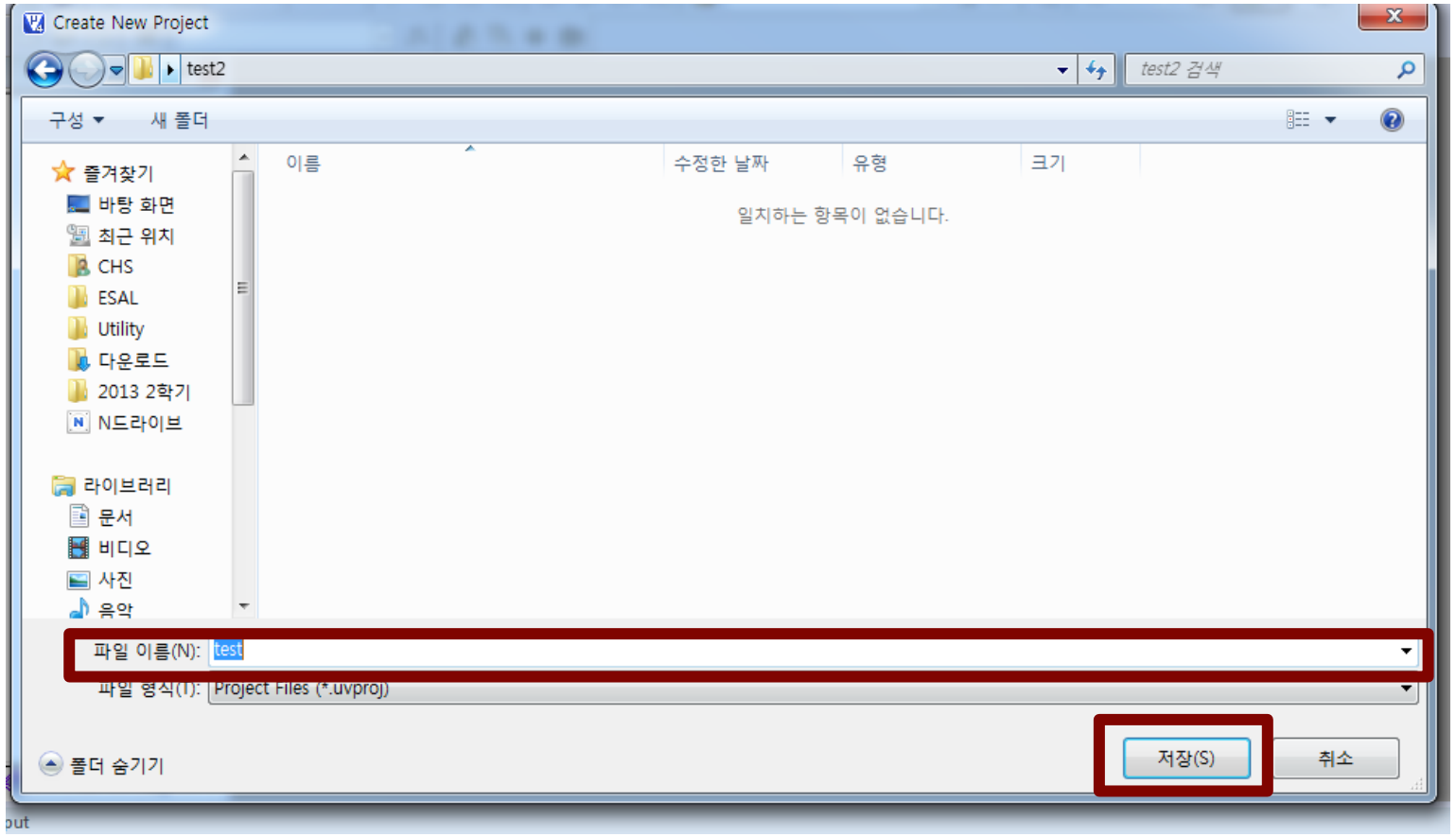
- Execute Keil uVision5

- Click ‘Project’ tab and click ‘New µVision Project’



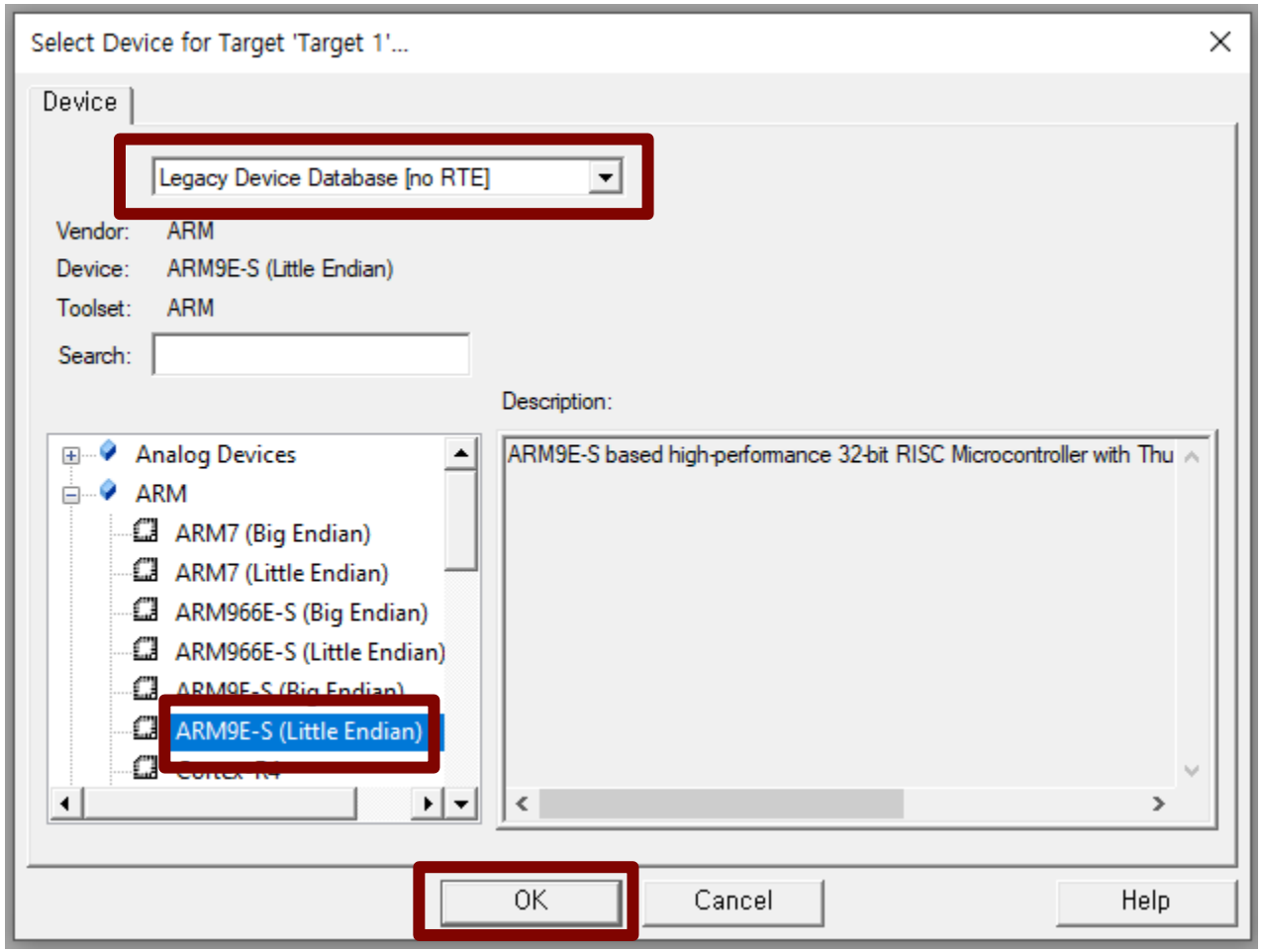
Project Creation (2/5)

- Write project name



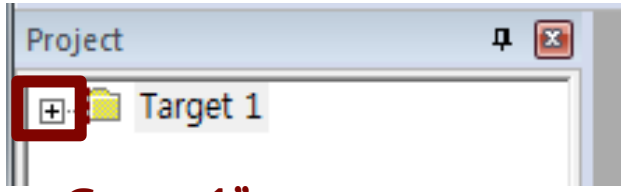
Project Creation (3/5)

- Select device for target
  - Arm ➔ ARM9E-S **(Little Endian)**
  - **Not Big Endian!**

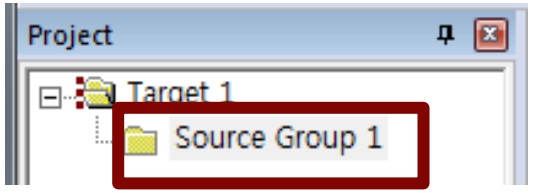


Project Creation (4/5)

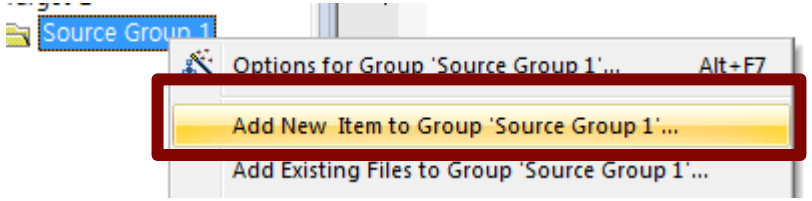
- Open “Target 1” by click + button



- Right click on “Source Group1”

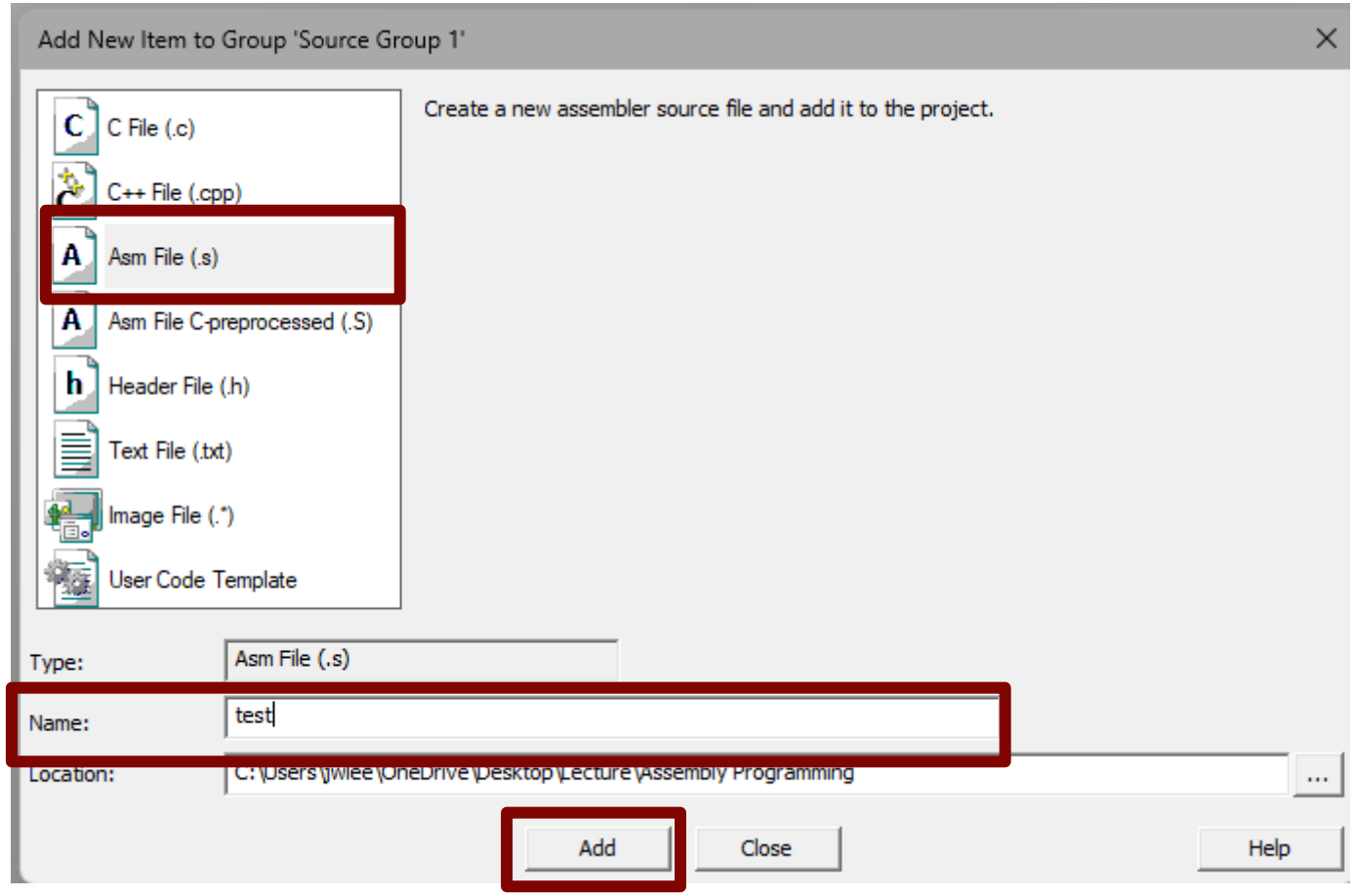


- Click “Add New Item to Group …”



Project Creation (5/5)

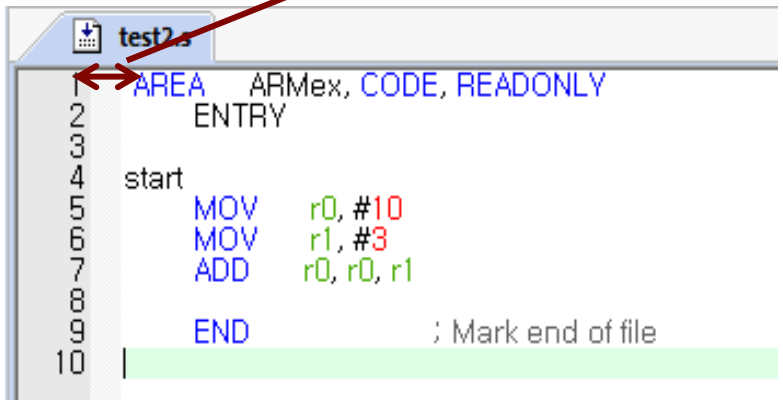
- Click ‘Asm File (.s)’ and write a file name



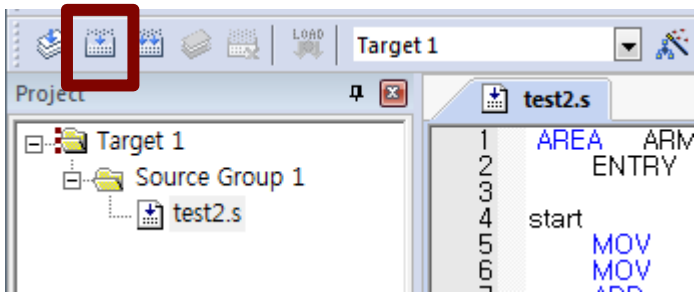
Example Execution (1/2)

- Write code like below figure

Tab to indent



- Click the 'build' button



Example Execution (2/2)

- Check the 'Build Output'

Build Output

```
Build target 'Target 1'
assembling test2.s...
linking...
Program Size: Code=12 RO-data=0 RW-data=0 ZI-data=0
".\test.axf" - 0 Errors, 0 Warning(s).
```

Debug (1/6)

Registers Window

Disassembly Window

Registers

Register	Value
Current	
R0	0x00000000
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000000
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x00000000
Mode	Supervisor

Project | Registers

Disassembly

```
0x00000000  E3A0000A  MOV     R0,#0x0000000A
     6:          MOV        r1,#3
0x00000004  E3A01003  MOV     R1,#0x00000003
     7:          ADD        r0,r0,r1
0x00000008  E0800001  ADD     R0,R0,R1
0x0000000C  00000000  ANDEQ   R0,R0,R0
0x00000010  00000000  ANDEQ   R0,R0,R0
0x00000014  00000000  ANDEQ   R0,R0,R0
0x00000018  00000000  ANDEQ   R0,R0,R0
0x0000001C  00000000  ANDEQ   R0,R0,R0
0x00000020  00000000  ANDEQ   R0,R0,R0
0x00000024  00000000  ANDEQ   R0,R0,R0
```

test.s

```
1   AREA    ARMex, CODE, READONLY
2   ENTRY
3
4 Start
5   MOV     r0,#10
6   MOV     r1,#3
7   ADD     r0,r0,r1
8
9   END             ;Mark end of file
10
```

Command

```
*** Error: 'C:\Keil\ARM\BIN\DARMP3.DLL' not found
Running with Code Size Limit: 32K
Load "C:\\Users\\CHS\\Desktop\\test\\test.axf"

*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 12 Bytes (0%)

WS 1, 0x0048

>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
```

Call Stack + Locals

Name	Location/Value	Type
__asm_0x0	0x00000000	void f()

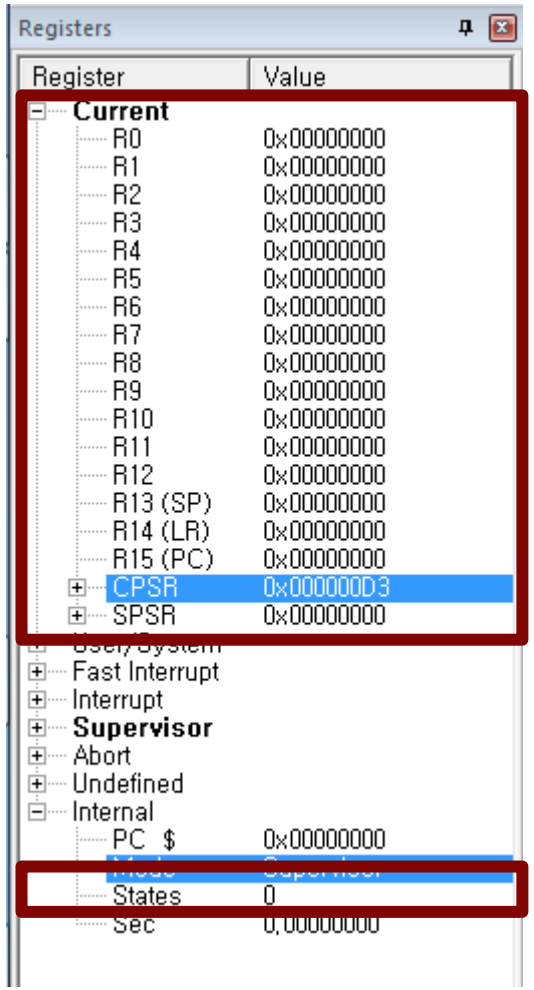
Call Stack + Locals | Watch 1 | Memory 1

Simulation | t1: 0.00000000 sec | L:5 C:1 | CAP NUM

Command Window

Debug (2/6)

- **Registers Window**
  - We can check values in the registers here
  - Execution time or cycles
  - Program Counter



Debug (3/6)

- **Disassembly Window**
  - We can see translated machine language

```
Disassembly
0x00000000  E3A0000A  MOV     R0,#0x0000000A
      6:        MOV        r1,#3
0x00000004  E3A01003  MOV     R1,#0x00000003
      7:        ADD        r0,r0,r1
0x00000008  E0800001  ADD     R0,R0,R1
0x0000000C  00000000  ANDEQ   R0,R0,R0
0x00000010  00000000  ANDEQ   R0,R0,R0
0x00000014  00000000  ANDEQ   R0,R0,R0
0x00000018  00000000  ANDEQ   R0,R0,R0
0x0000001C  00000000  ANDEQ   R0,R0,R0
0x00000020  00000000  ANDEQ   R0,R0,R0
0x00000024  00000000  ANDEQ   R0,R0,R0
```

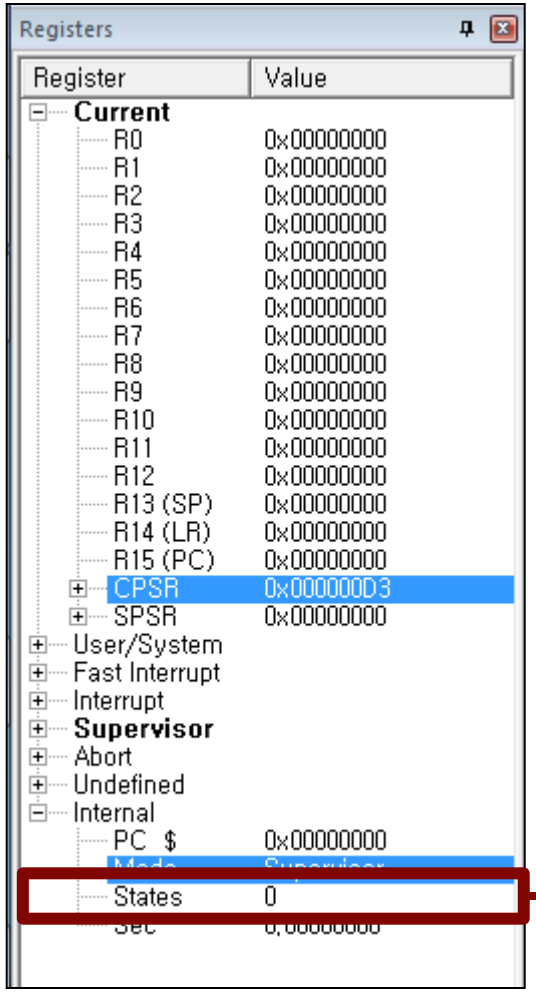
Debug (4/6)

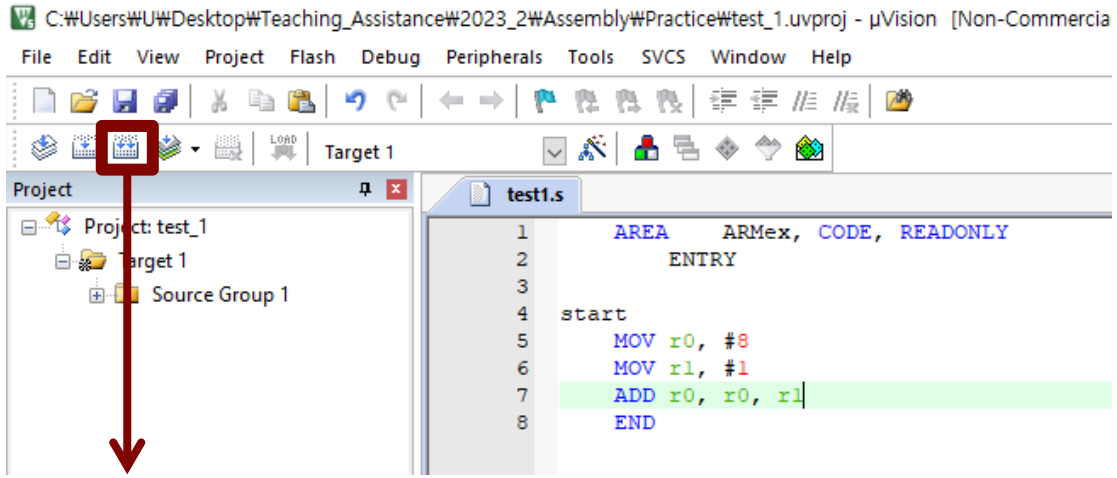
- We can check code size in the Command Window

```
Command
Running with Code Size Limit: 32K
Load "C:\\Users\\CHS\\Desktop\\test\\test.axf"

*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 12 Bytes (0%)

WS 1, 0x0048

>
ASSIGN BreakDisable BreakEnable BreakKill BreakList Break
```

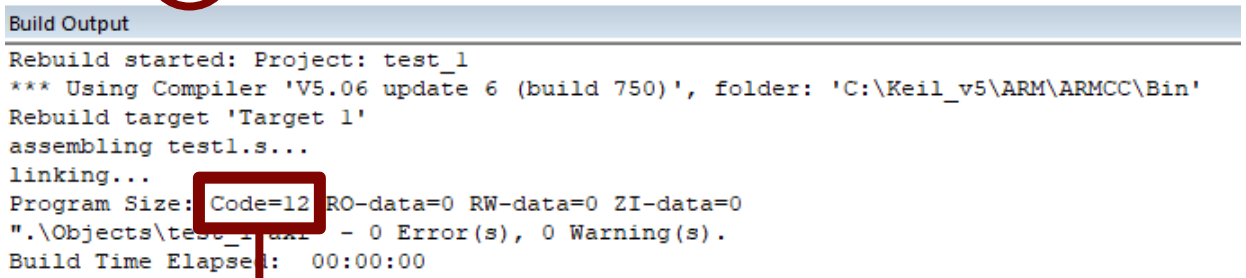
Debug (5/6)

- Check performance





① Rebuild



② Code size (Byte)

State

Debug (6/6)

- Shortcuts
  - Start debug mode ➔ Ctrl+F5
  - Break point ➔ F9
  - Check line ➔ F10
  - End debug mode ➔ Ctrl+F5

Example (1/5)

- Code

```
    AREA    ARMex, CODE, READONLY
    ENTRY

Start
    MOV     r0,#10  ;store integer 10 to register 0
    MOV     r1,#3   ;store integer 1   to register 1
    ADD     r0,r0,r1 ;add register0's value and register1's value and store result to register 0

    MOV     pc,lr   ;go to first instruction
    END             ;Mark end of file
```

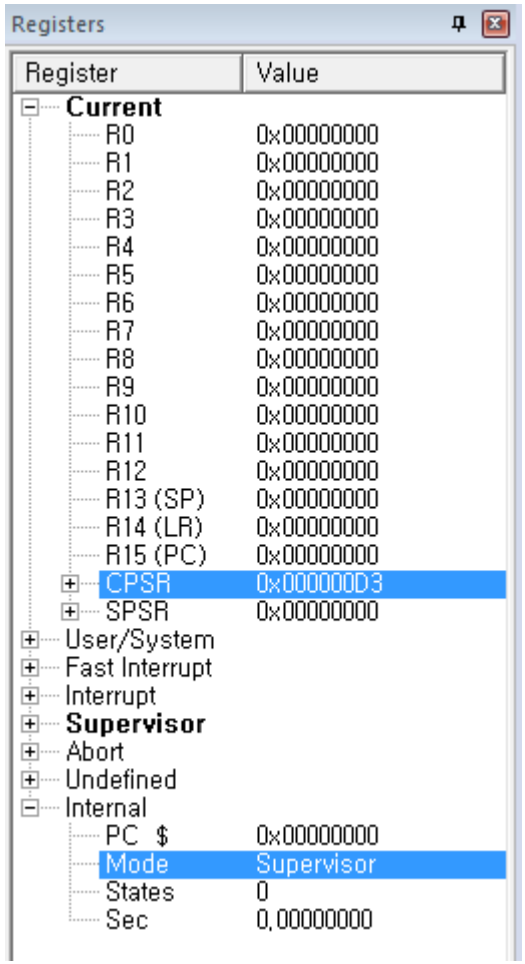
- Build

```
Build Output
assembling test.s...
test.s(10): warning: A1608W: MOV pc,<rn> instruction used, but BX <rn> is preferred
linking...
Program Size: Code=16 RO-data=0 RW-data=0 ZI-data=0
".\test.axf" - 0 Errors, 1 Warning(s).
```

- You can ignore the warning

Example (2/5)

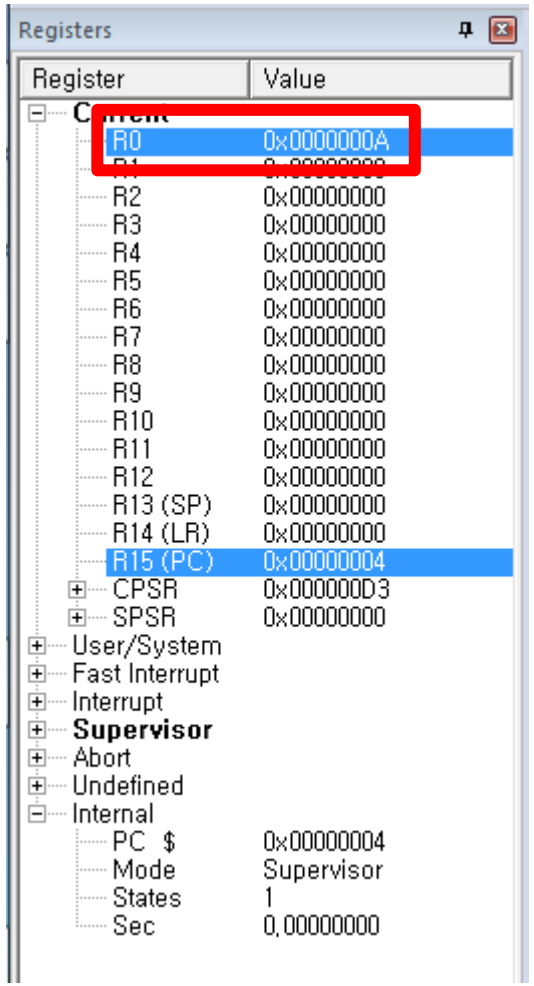
- Start debug




```
    AREA    ARMex, CODE, READONLY
    ENTRY

Start
    MOV     r0,#10  ;store integer 10 to register 0
    MOV     r1,#3   ;store integer 1   to register 1
    ADD     r0,r0,r1 ;add register0's value and register1's value and store result to register 0

    MOV     pc,lr   ;go to first instruction
    END             ;Mark end of file
```

Example (3/5)

- r0=10

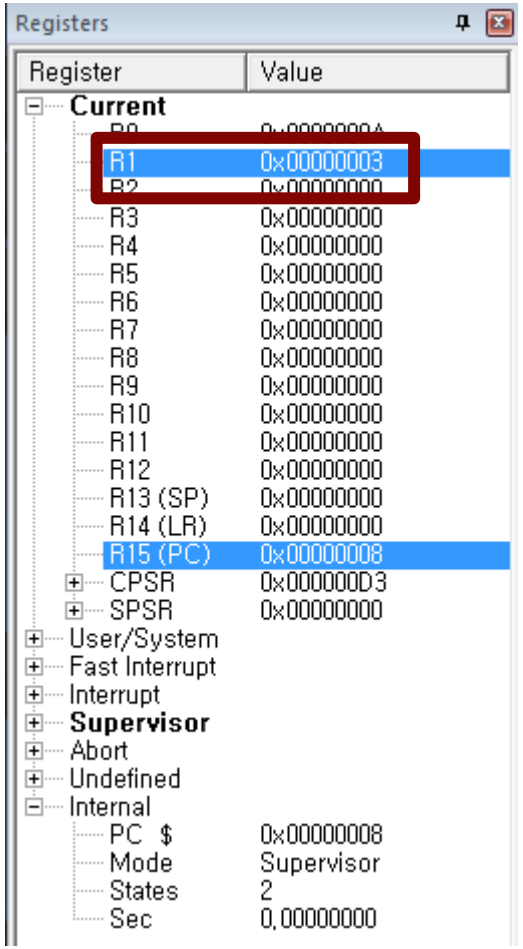


➔R0=10(0xA)

```
AREA    ARMex, CODE, READONLY
ENTRY

Start
  MOV   r0,#10  ;store integer 10 to register 0
  MOV   r1,#3   ;store integer 1  to register 1
  ADD   r0,r0,r1 ;add register0's value and register1's value and store result to register 0

  MOV   pc,lr   ;go to first instruction
  END           ;Mark end of file
```

Example (4/5)

- r1=3

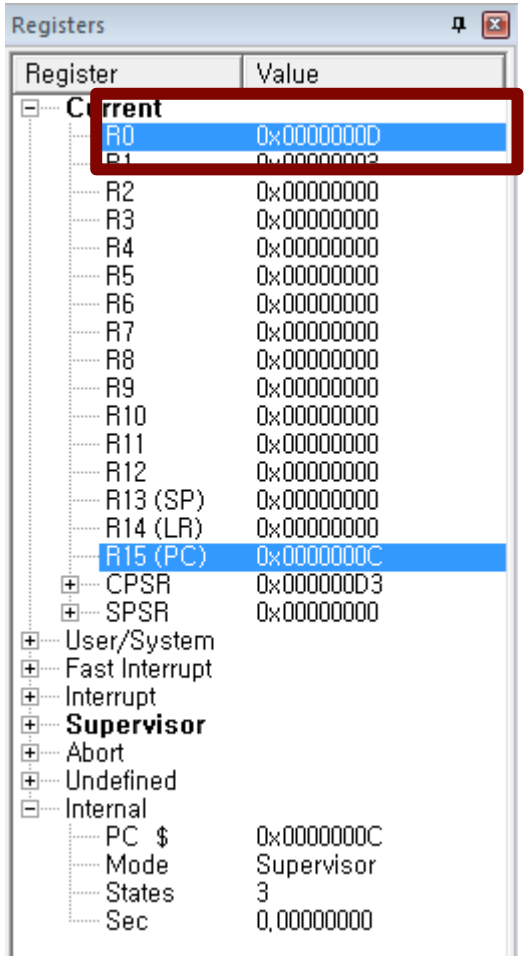


➔R1=3(0x3)

```
    AREA    ARMex, CODE, READONLY
    ENTRY

Start
    MOV     r0,#10  ;store integer 10 to register 0
    MOV     r1,#3   ;store integer 1   to register 1
    ADD     r0,r0,r1 ;add register0's value and register1's value and store result to register 0

    MOV     pc,lr   ;go to first instruction
    END             ;Mark end of file
```

Example (5/5)

- r0=r0+r1

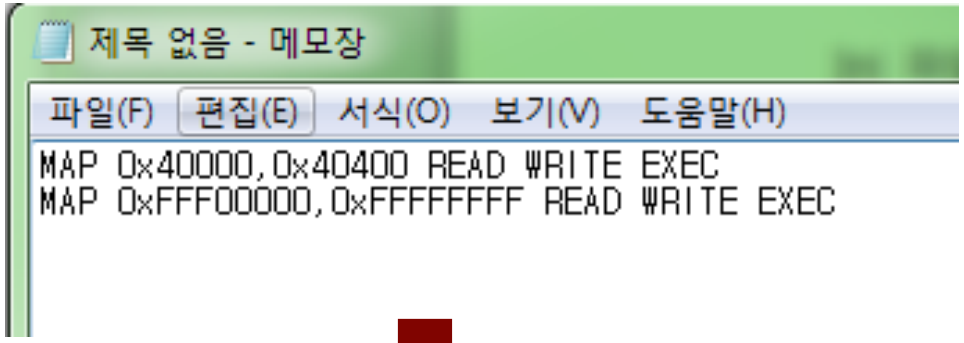


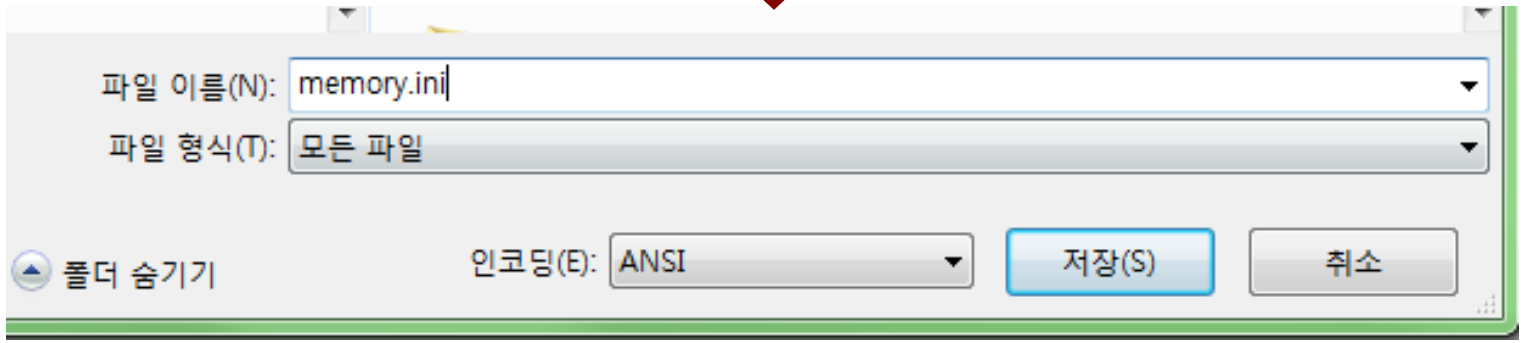
➔R0=13(0xD)

```
    AREA    ARMex, CODE, READONLY
    ENTRY

Start
    MOV     r0,#10  ;store integer 10 to register 0
    MOV     r1,#3   ;store integer 1  to register 1
    ADD     r0,r0,r1 ;add register0's value and register1's value and store result to register 0

    MOV     pc,lr   ;go to first instruction
    END             ;Mark end of file
```

Register INI File (1/6)

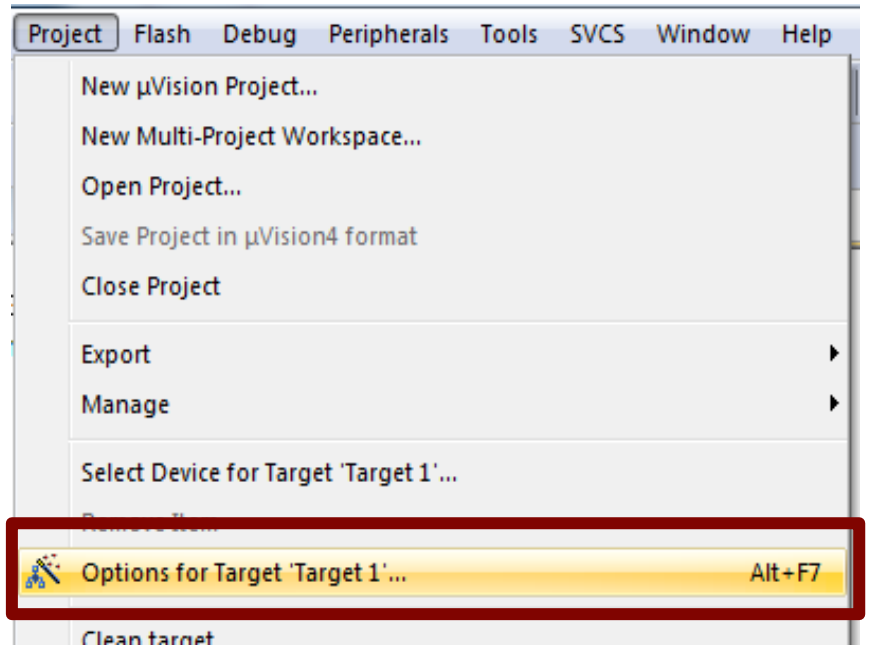
- Make ini file





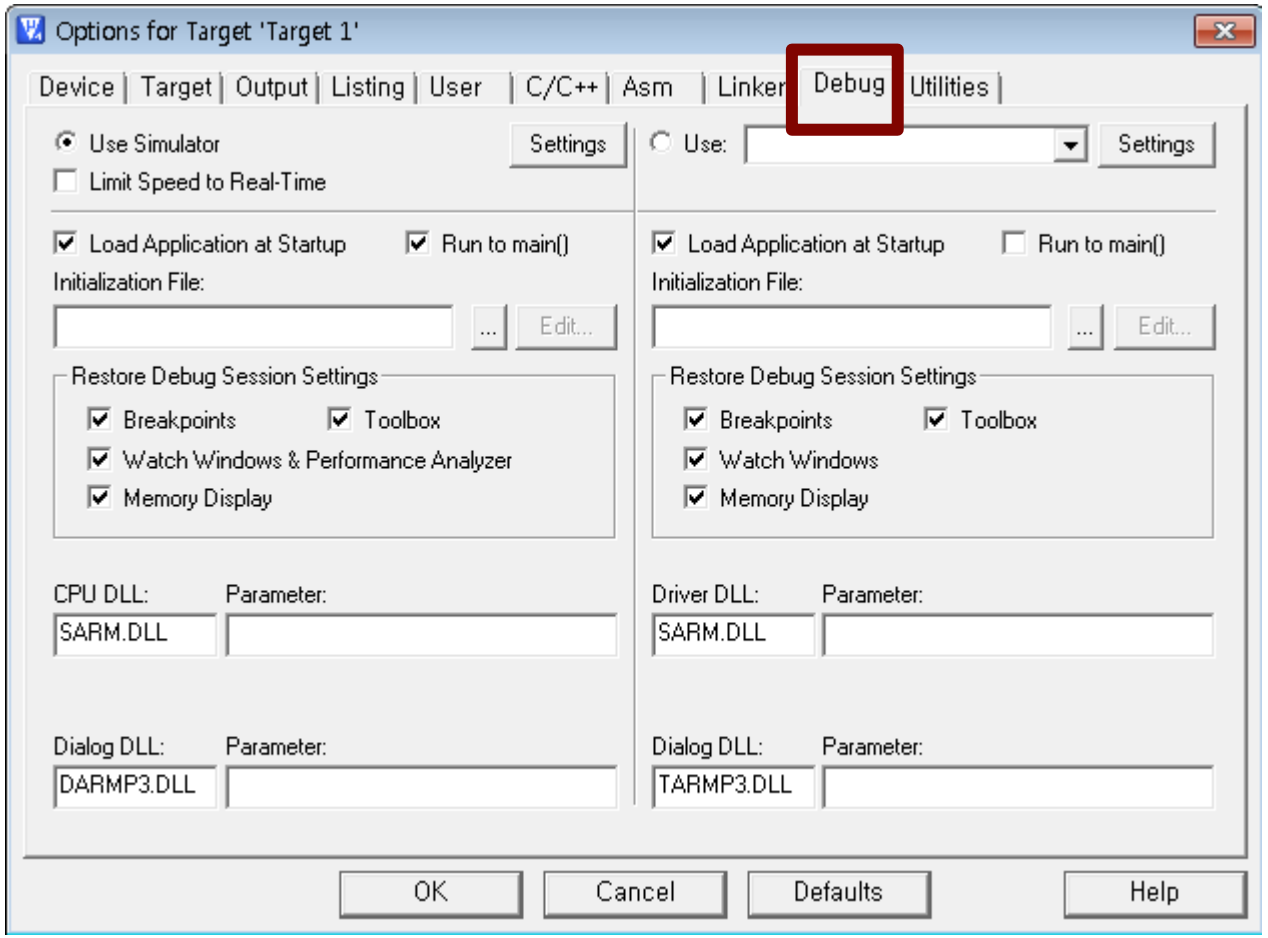
Register INI File (2/6)

- Activate Project tab ➔ Click Options for Target



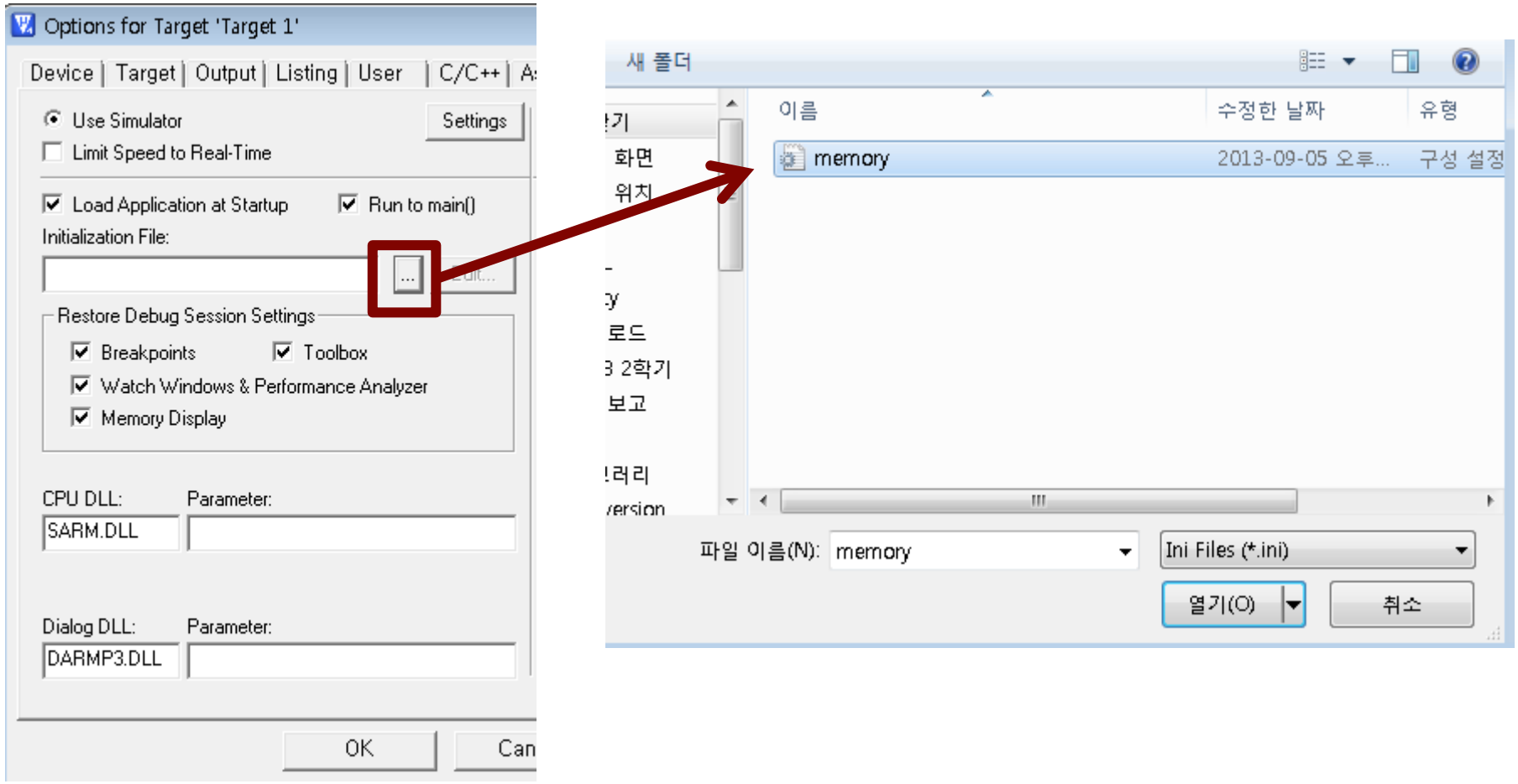
Register INI File (3/6)

- Debug tab Click



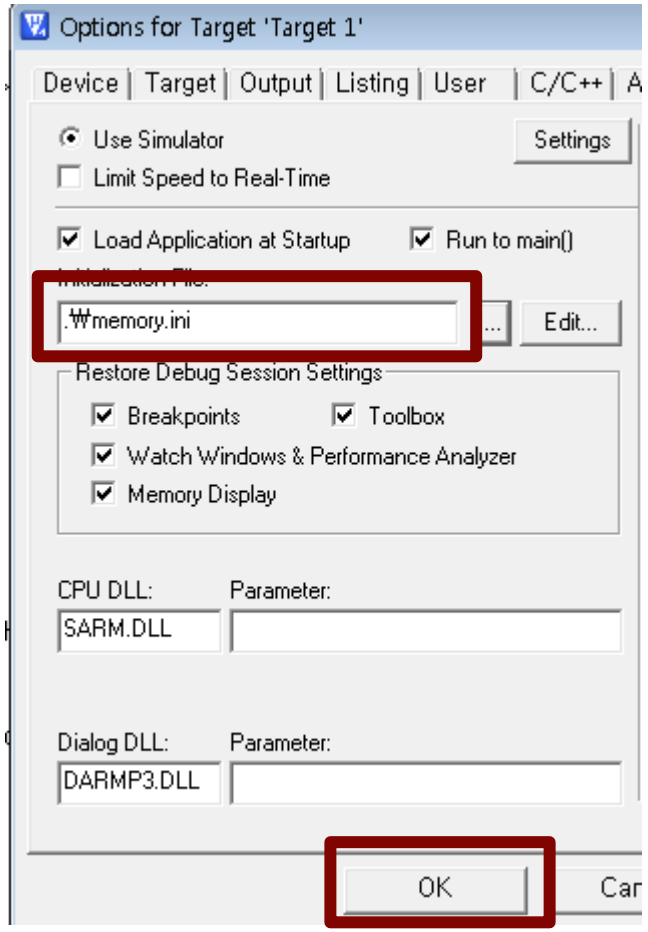
Register INI File (4/6)

- Click "…" button and register ini file



Register INI File (5/6)

- Check and click ok



Register INI File (6/6)

- You can check following logs when you start debug

```
Include "C:\\Users\\CHS\\Desktop\\Assembly\\debug.ini"
MAP 0x40000,0x40400 READ WRITE EXEC
MAP 0xFFF00000,0xFFFFFFFF READ WRITE EXEC
BS \\Assambly\Assembly.s\96
```

Basic Instructions

Table 1

Opcode [24:21]	Instruction	Description	Operation
0000	AND	Logical bit-wise AND	Rd:=Rn AND Op2
0001	EOR	Logical bit-wise exclusive OR	Rd:= Rn EOR Op2
0010	SUB	Subtract	Rd:= Rn - Op2
0011	RSB	Reverse subtract	Rd:= Op2-Rn
0100	ADD	Add	Rd:= Rn+Op2

Reference

- Table 1:ARM system-on-chip architecture second edition, page 120