

Design Assignment 2 – Assembly Language Programming

1 INTRODUCTION

You will write an assembly language routine to implement the following procedure:

```
uint32_t read_nvic_cpuid_base( uint32_t *reg_ptr,
                               uint32_t *implementer,
                               uint32_t *variant,
                               uint32_t *partno )
{
    uint32_t reg_val;
    reg_val      = *reg_ptr;
    *implementer = (reg_val >> 24) & 0xFF;
    *variant     = (reg_val >> 20) & 0xF;
    *partno      = (reg_val >> 4) & 0xFFF;
    return reg_val;
}
```

2 CREATE A PSOC PROJECT

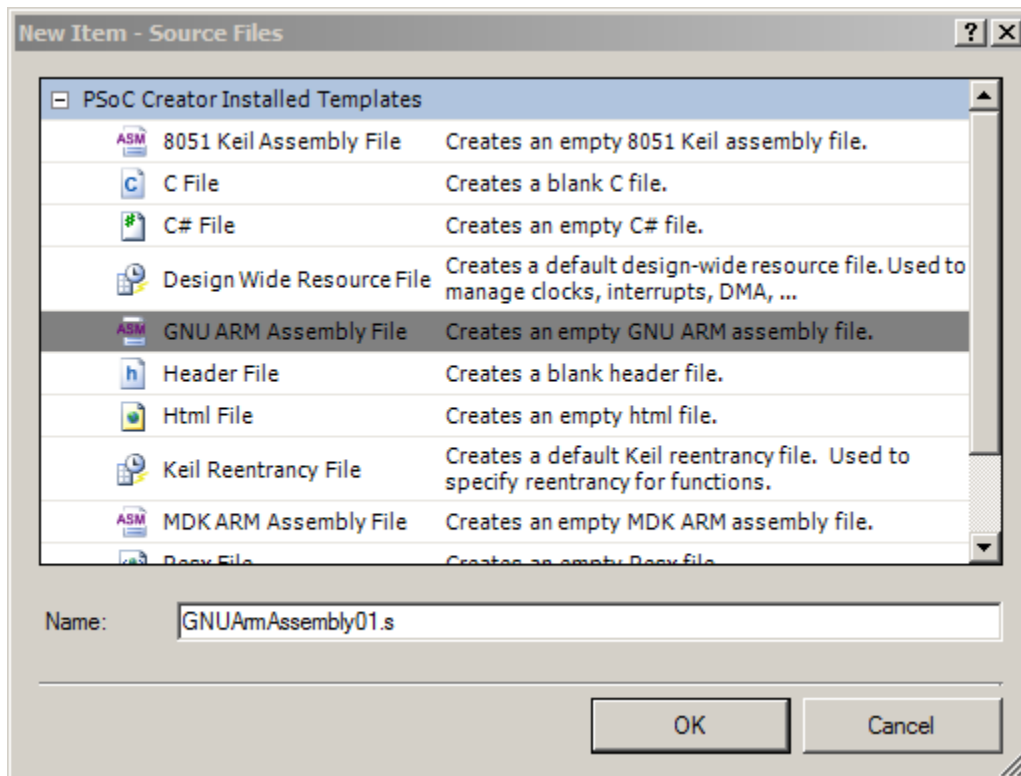
Create a new PSoC project for this Design Assignment. In your main.c file declare the prototype for the procedure **outside** of the main() procedure.

Within main() declare uint32_t variables for reg_val, reg_ptr, implementer, variant, partno.

You will call your assembly language procedure as follows:

```
#define NVIC_CPUID_BASE ( /* Fill in address here */ )
uint32_t *reg_ptr;
reg_ptr = NVIC_CPUID_BASE;
reg_val = read_nvic_cpuid_base( reg_ptr, &implementer, &variant, &partno );
```

Create your assembly file, right click on Add ... → New Item ..., then select GNU ARM Assembly File:



You will get the following boilerplate code:

```
/*
    .global FunctionName
    .func FunctionName, FunctionName
    .thumb_func
FunctionName:
    BX lr
    .endfunc
*/
```

Uncomment the boilerplate, and replace each of the four instances of `FunctionName` with `read_nvic_cpuid_base`.

Add your code and test and debug. Remember that you will need to save and restore any registers that you need to use in your assembly language routine. Save registers via `PUSH { regs }` and restore them via `POP { regs }`.

Single step your code, and examine and record the values stored in the variables declared in `main()`.

3 WRITE A REPORT

Write a report, which will be due by the date announced in class.

The report is to include, but not limited to the following:

- a) Introduction – brevity is a virtue, but highlight any features of the lab that were new to you.
- b) Software Architecture – very briefly describe the software “architecture” of your project, including as appropriate:
 - a. Functions and files that you defined
 - b. Macros or any new types or structures that you defined
 - c. Input/output, if any
 - d. Initialization
 - e. Infinite loop
 - f. Real-time aspects
- c) Procedures & Results
 - a. Brief subsection outlining each procedure that led to results.
 - b. Present results as table, picture, or other form, as most appropriate
 - c. Brief discussion of results including equations and graphs, as appropriate
- d) The report must be understandable to another engineer or supervisor not working on this project.
- e) A conclusion of your results and discussion of anything you found especially interesting or not expected from your work on this project.

Specific items that I will look for:

- a. How are parameters passed (call by value vs. call by reference)?
- b. How is value returned?
- c. How do you return from assembly language subroutine?

REPORT FORMAT:

- One report per team.
- **Cover sheet with Title, Class, Names, etc.**
- Microsoft Word file or files.
- All source code files checked in to GitHub.