

Práctica 1 - Parte 1

Para la resolución de estos problemas necesitará el manual de referencia del STM32F103xx (RM0008) y la hoja de datos del microcontrolador entre otros. Puede encontrar sugerencias de otros programas o herramientas necesarias en “enlaces útiles”

1. Cree un proyecto en STM32CubeIDE para el microcontrolador STM32f103c8 y verifique los siguientes puntos observando la salida de disassembly (archivo .list o la vista de disassembly en la perspectiva de debug)
 - a) La implementación de la multiplicación en una instrucción de assembler, del tipo $r = x * y$ donde las variables son `uint32_t` con valores dados en su declaración ¿se usa realmente una única instrucción para el proceso completo?
 - b) La implementación de la división de dos `uint32_t` en una instrucción
 - i) ¿Qué diferencia hay entre usar dos variables $r = x / y$ o usar una división constante como $r = x / 128$ o $r = x / 127$?
 - c) La implementación del guardado de una variable `int8_t` con valor -5 en una variable `int32_t`
2. Verifique en la implementación del punto anterior el camino completo de los datos. ¿Dónde se almacenan originalmente los valores que se dieron a las variables? ¿Se usa la memoria RAM en algún momento? Puede utilizar no sólo la vista “estática” del disassembly sino también la depuración instrucción por instrucción y la vista de registros. Utilice la vista Memory para verificar la memoria RAM.
3. Implemente una función `func1` que sume dos variables de tipo `uint32_t` y retorne el resultado y úsela en el código para sumar dos valores literales. Depure instrucción por instrucción con la vista de disassembly, registros, y memoria
 - a) ¿Cómo se le pasan los argumentos a la función?
 - b) ¿A qué segmento del mapa de memoria corresponde la dirección a donde salta?
 - c) ¿Cuál es la instrucción que usa para saltar y qué registro modifica?
 - d) ¿A qué dirección apunta el SP antes y después de saltar a la función? ¿Cuánto se reservó de stack?
 - e) ¿Cómo retorna el valor? ¿De donde se obtiene la dirección de retorno? ¿Se libera stack?
4. Reserve memoria dinámica para 100 bytes usando la función `malloc()`, asignando su retorno a un puntero de tipo `uint8_t`. ¿En qué ubicación se alojó este arreglo dinámico? ¿es la misma zona que otras variables? ¿Si no es la misma zona, cuál es la diferencia? Si lo es, ¿es en el mismo sector de esa zona?
5. Identifique en la documentación la descripción del periférico FMSC de la familia de dispositivos STM32f10xxx y si el microcontrolador STM32f103c8 en particular lo posee.

6. Si puede asignar la dirección de memoria correspondiente a cada registro, la siguiente función permite inicializar el periférico DWT y activar su función de conteo de ciclos para poder contar cuántos ciclos de clock se insumen durante la operación del microcontrolador.

```
void habilitar_contador_ciclos() {  
    DEMCR |= (1 << 24);  
    DWT_CTRL |= (1 << 0);  
    DWT_CYCCNT = 0;  
}
```

Rastree en la documentación, comenzando por el Reference Manual, el periférico DWT y los registros y bits utilizados.

- a) ¿A qué segmento del mapa de memoria pertenece? ¿Corresponde a ARM o a ST?
- b) ¿En qué documento se describe detalladamente la funcionalidad de cada registro?

7. Utilice la funcionalidad del punto anterior, la vista de disassembly, la vista de Expresiones (donde puede ver el valor de CYCCNT) y el debug paso a paso por instrucción para observar cuántos ciclos de clock toma cada una de las operaciones del punto (1).