
How to adapt projects built with previous versions of COSMIC to version 4.4.5 and later

Introduction

COSMIC (COS-C-COMPILER) versions 4.4.5 onward have an improved COSMIC linker to minimize the risk of virtual register corruption at the execution of an interrupt routine. The linker performs an automatic call-tree analysis for every STM8 interrupt routine and generates a linker error each time the issue is present. This can result in compilation errors for code that compiled properly with previous versions of COSMIC.

This technical note describes how the detected risk is fixed by a small change in the interrupt handler code in COSMIC version 4.4.5 and later.

Contents

- 1 Fix COSMIC @svlreg missing for interrupt function error 4**
 - 1.1 Description 4
 - 1.2 How to fix @svlreg COSMIC linker error 4

- 2 Revision history 6**

List of figures

Figure 1.	Disassembly code: additional code to save c_lreg register	4
Figure 2.	COSMIC linker error (@svlreg keyword missing)	4
Figure 3.	Missing @svlreg COSMIC linker error fix	5

1 Fix COSMIC @svlreg missing for interrupt function error

1.1 Description

The COSMIC compiler creates three virtual registers in the fast memory in addition to the few CPU registers of the STM8. These registers are called `c_x`, `c_y` and `c_lreg`.

The `c_lreg` register is an internal pseudo register of the COSMIC STM8 compiler. It consists in 4 bytes allocated in page0. The compiler uses it for 32-bit calculations, both long and float, whenever the microcontroller resources are not sufficient since the STM8 has no 32-bit register.

The `c_lreg` register should be saved under interrupt. Since this takes time and code (4 bytes to push and pop) and since STM8 interrupt routines usually do not use 32-bits computation, COSMIC saves the `c_lreg` register in the stack as a non-default option, controlled by the `@svlreg` keyword.

Most of STM8 interrupt functions do not need the extra code and cycles for saving `c_lreg` (especially for timing constraint processing) and for the rare cases where this is necessary, it is the user's responsibility to add it to the interrupt routines each time it is needed as illustrated in [Figure 1](#).

Figure 1. Disassembly code: additional code to save `c_lreg` register

stm8s_it.c:140 @svlreg INTERRUPT_HANDLER(EXTI_PORTD_IRQHandler, 6)			
0xe6f5	< .PORTD_IRQHandler>	0xBE02	LDW X, 0x02
0xe6f7	< .ORTD_IRQHandler+2>	0x89	PUSHW X
0xe6f8	< .ORTD_IRQHandler+3>	0xBE00	LDW X, 0x00
0xe6fa	< .ORTD_IRQHandler+5>	0x89	PUSHW X

To highlight this new COSMIC feature for customers and to avoid any eventual regression of user's applications, the COSMIC linker has been enhanced to automatically detect interrupt functions missing the `@svlreg` keyword. The COSMIC linker generates a linker error for each detection of the missing `@svlreg` keyword.

1.2 How to fix @svlreg COSMIC linker error

When using COSMIC version 4.4.5 or later, users can be confronted to applications that do not compile anymore while they compiled correctly with an older version of the COSMIC compiler. Some of the new compilation errors can result from the user's own interrupt code as shown in [Figure 2](#).

Figure 2. COSMIC linker error (@svlreg keyword missing)

```
Running Linker
clnk -l"C:\Program Files (x86)\COSMIC\FSE_Compilers\Lib" -o Debug\STM8-128-EVAL_Demo.sm8
-mDebug\STM8-128-EVAL_Demo.map Debug\STM8-128-EVAL_Demo.lkf
#error clnk Debug\STM8-128-EVAL_Demo.lkf:1 @svlreg missing for function f_EXTI_PORTD_IRQHandler
The command "clnk -l"C:\Program Files (x86)\COSMIC\FSE_Compilers\Lib" ... Debug\STM8-128-EVAL_Demo.lkf"
has failed, the returned value is: 1
exit code=1.
```

The fix for such errors is very simple. The user must simply add the missing `@svlreg` keyword to those interrupt routines where the compiler detects it is missing. An example is provided in [Figure 3](#). Once failing interrupt routines have been fixed, normal compilation is

restored with the additional security that no unlucky interrupt can crash the application in an unexpected way.

Figure 3. Missing @svlreg COSMIC linker error fix

```
#if defined(_COSMIC_)
/* 'svlreg option' is added to force the saving of the virtual long register */
@svlreg INTERRUPT_HANDLER(EXTI_PORTE_IRQHandler, 7)
#else
INTERRUPT_HANDLER(EXTI_PORTE_IRQHandler, 7)
#endif
```

2 Revision history

Table 1. Document revision history

Date	Revision	Changes
15-Feb-2018	1	Initial release.

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2018 STMicroelectronics – All rights reserved