

Relocation Overflow Error

From Texas Instruments Wiki

Jump to: [navigation](#), [search](#)

Contents

[hide]

- [1 Relocation Overflow Error](#)
- [2 Why does this happen?](#)
- [3 Solutions](#)
- [4 References](#)

Relocation Overflow Error[\[edit\]](#)

Sometimes in the development of a C6x DSP based project errors like seen below may crop up.

```
error: relocation overflow occurred at address 0x12345678 in section
      '.text' of input file
      '<path to a .obj file>'.
The 16-bit relocated address 0xFFFF is too large to encode in the 15-
bit field.
```

You may need to add a mask to the assembly instruction or use other target specific assembly features if you really only need the lowest 15 bits of this symbol.
Please see the section on Relocation in the Assembly User's Guide.

Why does this happen?[\[edit\]](#)

This means that you have run into the reason that near and far variables exist, but to explain fully we must go into some details on how memory accesses happen in the C6x.

Essentially, on the C64x processor, you have 32-bit instructions. Inside the 32-bit instruction, you can only access memory based on the offset of a CPU register. One of the CPU registers is used as the data pointer register for near memory when writing C code. Whenever you access near memory, the CPU will use 15 bits of the 32-bit instruction as the offset from the data pointer register. For this reason, the .bss near data memory section is limited to $2^{15} = 32\text{k}$ bytes of memory that can be accessed in a single CPU cycle, since it is always addressable.

However, to get access to the remainder of the memory map, you have far memory as well. Far memory requires that you use an MVKL and MVKH set of instructions to build the absolute 32-bit address of the memory you wish to access. For this reason, all data memory outside of the near memory section will take 3 CPU cycles to access, because the compiler must build the instructions to access the memory.

So what our compiler will do in default is make all global single element variables near variables, and all aggregate array memory far variables. Typically, this will work just fine but in your case this error is saying that some code in your executable is trying to access a variable in a 'near' fashion that is not near enough (i.e. it cannot fit the offset it would need within the 15 bit offset field).

Solutions[[edit](#)]

If you would like to improve upon this performance, or if you run out of near memory space, as it appears has happened with this error, there are some options available to you.

- Inside your CCS "Project" -> "Build Options", go to the "Compiler" tab, and then the "Advanced" category. Here, you will find an option called "Memory Models". If you select "--mem_model:data=far", this will resolve your issue because it will make all global memory far memory data. However, please note that this is not the most efficient way to do things (all memory accesses now take 3 instructions).
- Another thing you can do is use the "near" and "far" keywords with all of your global variables. This way, you can individually configure variables to be either near or far memory. This is what I would recommend doing as it will allow you to still take advantage of near memory for the 32kB worth of variables that you access most often. This is the most efficient solution.
- If this is stemming from library code and you already have tried the --mem_model:data=far option than this could mean the library needs to be rebuilt. Since the v5.1.0, the C6000 compilers default to near (16 bit) accesses and use trampolines to deal with jumps of greater than 1 million bytes. If an object file (or library) was compiled using an earlier compiler, the code will not be compatible if not built for near accesses and will need to be recompiled.

References[[edit](#)]

[C6000 Memory Models Article](#)

[Relocation Overflow Error thread on E2E Forums](#)

[TMS320C6000 Optimizing Compiler Guide SPRU187](#)

[TMS320C6000 Assembly Language Tools Guide SPRU186](#)

Keystone=

{{

1. switchcategory:MultiCore=

- For technical support on MultiCore devices, please post your questions in the [C6000 MultiCore Forum](#)
- For questions related to the BIOS MultiCore SDK (MCSDK), please use the [BIOS Forum](#)

Please post only comments related to the article **Relocation Overflow Error** here.

- For technical support on MultiCore devices, please post your questions in the [C6000 MultiCore Forum](#)
- For questions related to the BIOS MultiCore SDK (MCSDK), please use the [BIOS Forum](#)

Please post only comments related to the article **Relocation Overflow Error** here.



Links



[Amplifiers & Linear Audio Broadband RF/IF & Digital Radio Clocks & Timers Data Converters](#)

[DLP & MEMS High-Reliability Interface Logic Power Management](#)

[Processors](#)

- [ARM Processors](#)
- [Digital Signal Processors \(DSP\)](#)
- [Microcontrollers \(MCU\)](#)
- [OMAP Applications Processors](#)

[Switches & Multiplexers Temperature Sensors & Control ICs Wireless Connectivity](#)

Retrieved from

"https://processors.wiki.ti.com/index.php?title=Relocation_Overflow_Error&oldid=19117"

Category:

- [Compiler](#)

Navigation menu

Personal tools

- [Log in](#)
- [Request account](#)

Namespaces

- [Page](#)
- [Discussion](#)



Variants

Views

- [Read](#)
- [View source](#)
- [View history](#)



More

Search



Navigation

- [Main Page](#)
- [All pages](#)
- [All categories](#)
- [Recent changes](#)
- [Random page](#)
- [Help](#)

Toolbox

- [What links here](#)
- [Related changes](#)
- [Special pages](#)
- [Printable version](#)
- [Permanent link](#)
- [Page information](#)
- This page was last edited on 5 November 2009, at 10:41.
- Content is available under [Creative Commons Attribution-ShareAlike](#) unless otherwise noted.
- [Privacy policy](#)
- [About Texas Instruments Wiki](#)
- [Disclaimers](#)
- [Terms of Use](#)

