1 Introduction

This document provides two sets of options in the CodeWarrior tools to produce optimal code generation for the Power Architecture e200 core. One set optimizes for speed; another set optimizes for size.

For more information on the build tools, refer to the following compiler documents:

- CW C++ Notes 4.x.txt
- CW Embedded PPC Notes 4.0.x.txt

2 Optimization for Speed

Compiler (mwcceppc) options

```
-proc 2en -fp spfp_only -vle -ppc_asm_to_vle -char
unsigned -sym dwarf-2,full -wchar_t off -enum min -
04 -opt speed -inline auto,smart -ipa program -
use_lmw_stmw on -use_isel on -flag no-switch_tables
-flag switch_op -lang c99 -flag c9x_alias_by_type -
sdata 71 -sdata2 127 -prefix
ansi_prefix.PPCEABI.bare.h -schedule on -
spe_vector
```
Linker (mwldeppc) options

- proc Zen -fp spfp_only -lRuntime.PPCEABI.V.SP.UC.a -lMSL_C.PPCEABI.bare.V.SP.UC.a -lMSL_C++.PPCEABI.bare.V.SP.UC.a -sym dwarf-2.full -code_merging all,aggressive -far_near_addressing -vle_enhance_merging -vle_bl_opt -char unsigned -sdata 71 -sdata2 127 -map

**TIP** Experiment with the values of the `-sdata` and `-sdata2` options. The compiler and linker need to have the same thresholds. Make them as large as possible until link errors are generated saying that the small data errors have overflowed. The threshold for `sdata` is independent of the threshold for `sdata2`. The default is 8 for each. If there is lots of small data, this value may need to be lowered to less than 8. One way to determine the optimal threshold is to look at `.data`, `.bss` and `.rodata` sections in the link map file. Ideally, only CodeWarrior library data is in these sections. If any application code in these sections, try to make the thresholds match the largest application data. If a link error is generated saying that relocation 109 does not match the section of an object and the object is in a library, bump down the threshold or rebuild the library with larger thresholds that match the application small data thresholds. If no link error is generated, it is OK if library and application thresholds do not match. `.rodata` corresponds to `sdata2` and `.data` and `.bss` corresponds to `sdata`.

**TIP** Change `-spe_vector` to either

- `-spe_addl_vector` or
- `-spe2_vector`

depending on processor support. For example z3 and some z6 processors, e.g. MPC5566 (Viper), support `-spe_addl_vector`

**TIP** Change `-schedule` on to either

- `-pragma "schedule z750"` or
- `-pragma "schedule z760"

if the targeted processor is in the z750 or z760 families.

**TIP** The option `-ipa program` generally produces the best code both when trying to obtain fastest speed or smallest size.

### 3 Optimization for Size

Compiler (mwcceppc) options

- proc Zen -fp spfp_only -vle -ppc_asm_to_vle -char unsigned -sym dwarf-2.full -enum min -04 -func_align 4 -opt space -inline auto,smart -ipa program -use_lmwy stmwx on -Cpp_exceptions off -RTTI off -wchar_t off -bool off -use_isel on -flag no-switch_tables -flag switch_op -lang c99
-flag c9x_alias_by_type -sdata 32767 -sdata2 127 -prefix ansi_prefix.PPCEABI.bare.SZ.h - schedule on -spe_vector

Linker (mwldeppc) options

-proc Zen -fp spfp_only -lRuntime.PPCEABI.V.SP.UC.a -lMSL_C.PPCEABI.bare.SZ.V.SP.UC.a -lMSL_C++.PPCEABI.bare.SZ.V.SP.UC.a -sym dwarf-2,full -code_merging all,aggressive - far_near_addressing -vle_enhance_merging -vle_bl_opt -char unsigned -sdata 32767 -sdata2 127 - map

---

**NOTE** The tips in the previous Section 2, “Optimization for Speed” also apply to this section.