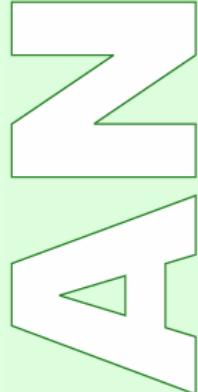




GENTOS

Application Note #006 (AN006-V1.3)



[GENTOS] How to Reduce the Code Size using CORERIVER Compiler (gencc)

V1.3

February 2006

- ◆ CORERIVER Semiconductor reserves the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice.
- ◆ Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.
- ◆ The CORERIVER Semiconductor products listed in this document are intended for usage in general electronics applications. These CORERIVER Semiconductor products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury.

Contents

1. Assign Parameters as Global Variables
2. Replace Function to Its Content
3. Assign Variable using Suitable Data Size
4. Remove the option "--debug"
when Compiling

1. Assign Parameters as Global Variables

- ◆ The parameter passing by a function with many parameters causes the code size to be increased.
- ◆ To reduce the code size, assign some parameters in functions as global variables.

Normal
Code

```
typedef unsigned char   BYTE; // 1 Byte
typedef unsigned int    WORD; // 2 Bytes

//-----
void main(void) {
    BYTE input0, input1, output; // Local Variables

    sub_function(input0, input1, output);
}

//-----
void sub_function(BYTE input0, BYTE input1, BYTE output) {
    // Some C code is here.
}
```

Recommended
Code

```
BYTE input0, input1, output; // Global Variables

//-----
void main(void) {

    sub_function();
}

//-----
void sub_function(void) {
    // Some C code is here.
}
```

2. Replace Function to Its Content

- ◆ The code size can be increased by the functions called rarely or the functions having small code size.
- ◆ Replace function to its content directly for small code size.
- ◆ The content of function can be defined as a MACRO.

Normal
Code

```
//-----
void main(void) {
    Main Statement Part 1
    sub_function();
    Main Statement Part 2
}

//-----
void sub_function(void) {
    Sub Statement Part
}
```

Recommended
Code

```
//-----
void main(void) {
    Main Statement Part 1
    Sub Statement Part          // Replace Function to
                                // its content.
    Main Statement Part 2
}
```

3. Assign Variable using Suitable Data Size

- ◆ A variable with unsuitable data size (large) makes the code size to be larger.
- ◆ Reassign some variables using suitable data size.

Normal
Code

```
typedef unsigned char   BYTE; // 1 Byte
typedef unsigned int    WORD; // 2 Bytes

WORD  loop_cnt;           // 2 Bytes, Global Variable

//-----
void main(void) {
    for(loop_cnt=0; loop_cnt<10; loop_cnt++) {
        :
    }
}
```

Recommended
Code

```
BYTE  loop_cnt;           // 1 Byte, Global Variables

//-----
void main(void) {
    for(loop_cnt=0; loop_cnt<10; loop_cnt++) {
        :
    }
}
```

4. Remove the Option “`--debug`” when Compiling

- ◆ You can get the reduced code size by removing the option “`--debug`” when compiling.
- ◆ But, in this case, you cannot use the debugger on MDS equipment because the compiler loses the debug information.
- ◆ Instead of debugging, you can use the menu “Emulation\Download & Run” to run and verify the code.

