TI ARM Clang Complier/Linker v4.0

Linker Generated CRC example

References

- Chapter 10.9 Linker generated CRC tables and CRC over memory ranges
- 10.9.2.6 Interface when using the crc() operator

Add: #include "C:\ti\ccs1280\ccs\tools\compiler\ti-cgt-armllvm_4.0.3.LTS\include\c\crc_tbl.h"

10.9.2.6. Interface When Using the crc() Operator

CRCs over memory ranges are stored in a table format similar to that shown in Interface When Using the crc_table() Operator for the CRCs over sections. However, the table format is different than that of CRC tables.

The following figure shows the storage format for CRCs over memory ranges with example values:

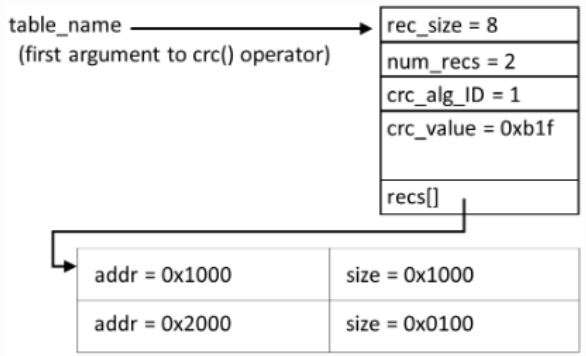
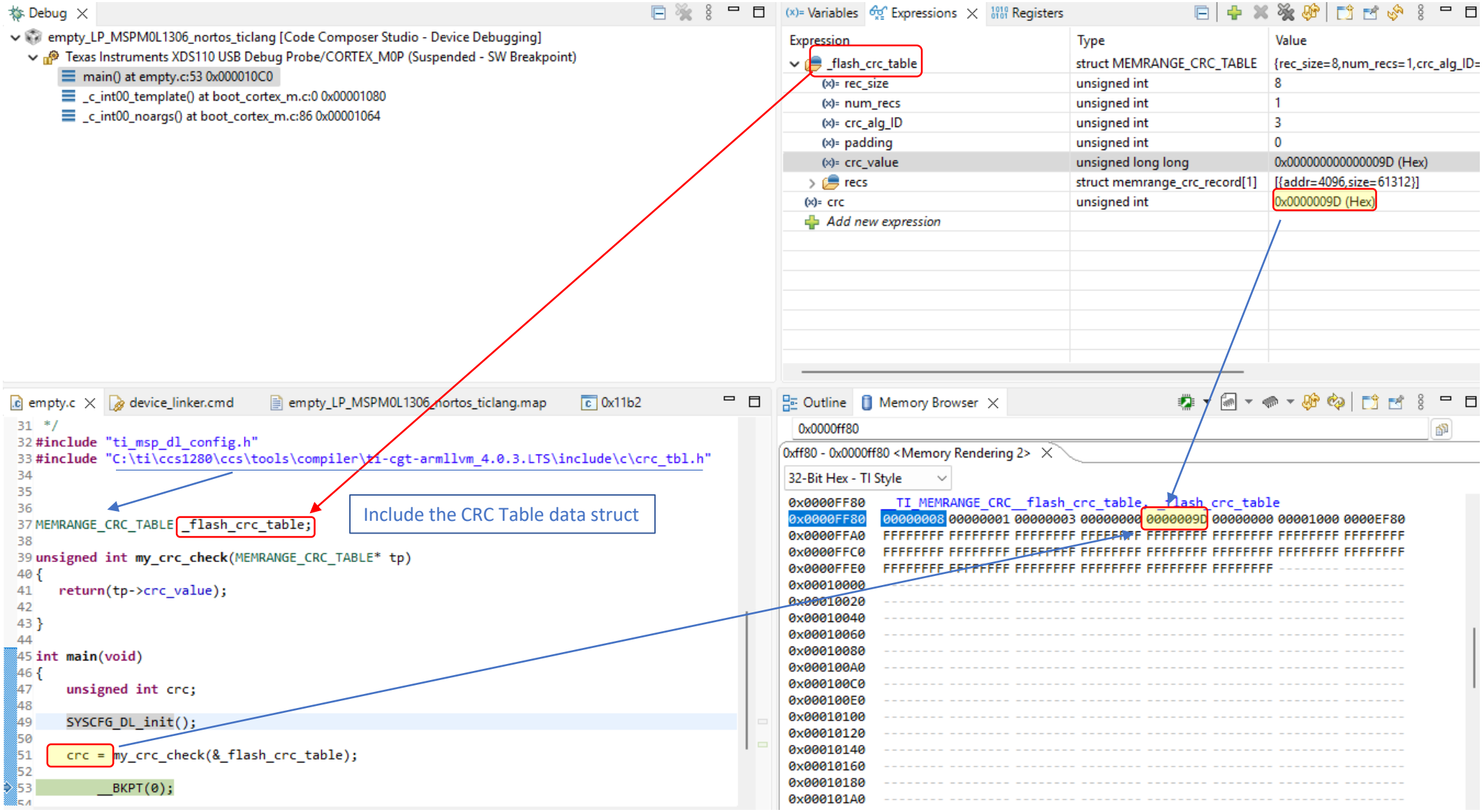


Figure 10.8 *CRC Storage Format*

The table header stores the record count and size, as well as the algorithm type and the CRC result. Each table entry encodes the start address and length of a memory range that was used to compute the CRC.

The following header file excerpt shows the C structures the linker creates to manage the CRC information:

```
typedef struct memrange_crc_record {
    uintptr_t           addr;       /* starting address */
#if defined(__LARGE_CODE_MODEL__) || defined(__LARGE_DATA_MODEL__)
    uint32_t            size;       /* size of data in 8-bit addressable units */
#else
    uint16_t            size;       /* size of data in 8-bit addressable units */
#endif
} MEMRANGE_CRC_RECORD;
typedef struct memrange_crc_table {
    uint16_t            rec_size;   /* 8-bit addressable units */
    uint16_t            num_recs;   /* how many records are in the table */
    uint16_t            crc_alg_ID; /* CRC algorithm ID */
    uint32_t            crc_value;  /* result of crc */
    MEMRANGE_CRC_RECORD recs[1];
} MEMRANGE_CRC_TABLE;
```


Debug
empty_LP_MSPM0L1306_nortos_ticlang [Code Composer Studio - Device Debugging]
Texas Instruments XDS110 USB Debug Probe/CORTEX_M0P (Suspended - SW Breakpoint)
main() at empty.c:53 0x000010C0
_c_int00_template() at boot_cortex_m.c:0 0x00001080
_c_int00_noargs() at boot_cortex_m.c:86 0x00001064
Variables
Expressions
Registers
Expression
Type
Value
_flash_crc_table
struct MEMRANGE_CRC_TABLE
{rec_size=8,num_recs=1,crc_alg_ID=
rec_size
unsigned int
8
num_recs
unsigned int
1
crc_alg_ID
unsigned int
3
padding
unsigned int
0
crc_value
unsigned long long
0x000000000000009D (Hex)
recs
struct memrange_crc_record[1]
[{addr=4096,size=61312}]
crc
unsigned int
0x0000009D (Hex)
Add new expression
empty.c
device_linker.cmd
empty_LP_MSPM0L1306_nortos_ticlang.map
0x11b2
*/
#include "ti_msp_dl_config.h"
#include "C:\ti\ccs1280\ccs\tools\compiler\ti-cgt-armllvm_4.0.3.LTS\include\c\crc_tbl.h"
MEMRANGE_CRC_TABLE _flash_crc_table;
unsigned int my_crc_check(MEMRANGE_CRC_TABLE* tp)
{
return(tp->crc_value);
}
int main(void)
{
unsigned int crc;
SYSCFG_DL_init();
crc = my_crc_check(&_flash_crc_table);
__BKPT(0);
Include the CRC Table data struct
Outline
Memory Browser
0x0000ff80
0xff80 - 0x0000ff80 <Memory Rendering 2>
32-Bit Hex - TI Style
0x0000FF80 TI_MEMRANGE_CRC__flash_crc_table__flash_crc_table
0x0000FF80 00000008 00000001 00000003 00000000 0000009D 00000000 00001000 0000EF80
0x0000FFA0 FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
0x0000FFC0 FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
0x0000FFE0 FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
0x00010000
0x00010020
0x00010040
0x00010060
0x00010080
0x000100A0
0x000100C0
0x000100E0
0x00010100
0x00010120
0x00010140
0x00010160
0x00010180
0x000101A0

Linker .cmd file

```
-uinterruptVectors
--stack_size=256
#define _BOOT_START_ADDRESS_    0x00000000
#define _BOOT_SIZE_             0x00000800
#define _APP_START_ADDRESS_     0x00001000


MEMORY
{
    //FLASH           (RX)  : origin = 0x00000000, length = 0x0000FFF8
    FLASH_BOOT      (RX)  : origin = _BOOT_START_ADDRESS_, length = _BOOT_SIZE_
    SRAM            (RWX) : origin = 0x20000000, length = 0x00001000
    BCR_CONFIG      (R)   : origin = 0x41C00000, length = 0x000000FF
    BSL_CONFIG      (R)   : origin = 0x41C00100, length = 0x00000080

    CRC_RESULT      (RX)  : origin = 0x0000FF80, length = 0x80

GROUP
{
    FLASH_APP  : origin = _APP_START_ADDRESS_, length = 0x00010000 - _APP_START_ADDRESS_ - 0x80

}crc(_flash_crc_table, algorithm=CRC8_PRIME)
}

SECTIONS
{
    .intvecs:   > 0x00000000
    .text    : palign(8) {} > FLASH_APP
    .const   : palign(8) {} > FLASH_APP
    .cinit   : palign(8) {} > FLASH_APP
    .pinit   : palign(8) {} > FLASH_APP
    .rodata  : palign(8) {} > FLASH_APP
    .ARM.exidx    : palign(8) {} > FLASH_APP
    .init_array   : palign(8) {} > FLASH_APP
    //.binit        : palign(8) {} > FLASH_APP
    .TI.ramfunc   : load = FLASH_APP, palign(8), run=SRAM, table(BINIT)

    .vtable :   > SRAM
    .args   :   > SRAM
    .data   :   > SRAM
    .bss    :   > SRAM
    .sysmem :   > SRAM
    .stack  :   > SRAM (HIGH)

    .BCRConfig  : {} > BCR_CONFIG
    .BSLConfig  : {} > BSL_CONFIG

    .TI.memcrc > CRC_RESULT//, type = COPY
}
```