CML Microcircuits
COMMUNICATION SEMICONDUCTORS

Application Note

Writing a Function Image^{TM} to Serial Memory

Publication: AN/Gen/FI/Programmer_FI/1 January 2007

1 Introduction

The CMX703x and CMX704x devices require a Function Image^{TM} to be loaded from an external memory as part of an initialisation routine. Because it is likely that a Function Image^{TM} will be programmed into the target board once or very few times, a low endurance memory is sufficient. The small footprint and compact bus width make standalone EEPROM or the newer, flash-based, Serial PROM a good choice. Once mounted in the target system, it is preferable to program this serial memory using the CMX703x or CMX704x.This Application Note discusses how to use the in-circuit programming capability of Thick Stub Function Images^{TM}.

2 Terminology

Target device – the CMX703x or CMX704x in the target system.
Device Function Image^{TM} – Function Image^{TM} used to set the functionality of the target device.
Thick Stub – Thick Stub Function Image^{TM} (fi_eeprom_ xx.h or fi_flash_xx.h) for programming a device Function Image, via C-BUS, into an EEPROM or serial flash memory connected to the SPI of the target device.

3 Loading the Thick Stub

The latest version of the Thick Stub should be downloaded from the CML Portal (My CML). This is loaded into the target device using the method below and detailed in the target device's Data Sheet.

Open the Thick Stub in a text viewer and note the checksum values for the Checksum registers ($B8), ($B9), ($A9) and ($AA) given in the file header. Download the Thick Stub into the target device and confirm that the load was successful by reading the Checksum registers and comparing with the expected values. For fi_eeprom_11.h these are;

$B8 (Block 1 Checksum High Word) = 0xFFDA
$B9 (Block 1 Checksum Low Word) = 0x9ADC
$A9 (Block 2 Checksum High Word) = 0x0000
$AA (Block 2 Checksum Low Word) = 0x0000

Concatenate checksum reads for the overall checksum thus:
Checksum = $B8 + $B9 + $A9 + $AA

Note the checksum will be different for other Function Images such as fi_flash_10.h.

Now write to the Programming register ($C8) the Activation Code high word followed by the Activation Code low word (0xEEEEEEEE). The target device is now ready to write a device Function Image^{TM} into the serial memory.

4 Preparing to write a device Function Image^{TM}

With the Thick Stub loaded into the target device it is possible to program a serial memory connected to the SPI port, via the C-BUS.

Nominally, the target devices use a 19.2MHz clock and an oscillator of this value is fitted on the PE0201 Evaluation Kits. If a different frequency is used then the clock divider settings in the Main CLK Ref register ($BD) must be configured as detailed in the target device's User Manual or the serial memory's timings will be violated.

The BOOTEN1 and BOOTEN2 pins will have been set to 11 prior to loading the Thick Stub. It is important these are unchanged during the following procedure.

```
D/*DBlock 1: DB1, Load seq 1*/
#define DB1_PTR        0x4010  /* Data block pointer*/
#define DB1_LEN        0x0187  /* Data block length, in 16 bit words*/
#define DB1_CHK_HI      0xFFDA
#define DB1_CHK_LO      0x9ADC
...
…
/*DBlock 2: DB2, Load seq 2*/
#define DB2_PTR        0x3000  /* Data block pointer*/
#define DB2_LEN        0x1000  /* Data block length, in 16 bit words*/
#define DB2_CHK_HI      0x0053
#define DB2_CHK_LO      0x060B
…
…
/*DBlock 3: ACTIVATE, Load seq 3*/
#define ACTIVATE_PTR    0x4010
#define ACTIVATE_LEN    0
/* Data block content*/
/* No data block or checksum for activate block*/
```

Figure 1. Example file headers for all data blocks

Get the parameters required by the Thick Stub:

Open the device Function Image^{TM} in a text reader.

Copy the following items from the file header:

1. The 2 data block start addresses (value following DBn_PTR, where n is the number of the block).
2. The 2 data block length (value following DBn_LEN, where n is the number of the block).

The host processor is now ready to programme a Function Image^{TM} into the serial memory via the SPI. The host control and data path for the download being C-BUS. Please refer to the target device's User Manual for details of C-BUS operation.
Wait until the PRG flag of the Status register ($C6) = 1 between each block write.

5 Writing the device Function Image^{TM} to serial memory

1. The data block start address is written to the Modem Address register ($B6).
2. The data block length is written to the special register ($B7).
3. 0xABCD is written to the Programming register ($C8) to initiate the data load.
4. Each data word is written to the Programming register ($C8) until the whole block has been written.
5. The process is then repeated until both data blocks are written.

This process is flowcharted in section 7.

6 Loading a Function Image^{TM} from external serial memory

1. BOOTEN1 must be set to one and BOOTEN2 must be set to zero.
2. A C-BUS General Reset is issued or the CMX7031 is power cycled.

The device Function Image^{TM} will automatically load as part of the target device's start-up procedure. It is good practice to check that the device Function Image^{TM} has been successfully loaded from the serial memory into the target device.

Refer to the device Data Sheet for the normal start-up procedure and operation.

7 Procedure for programming the serial memory

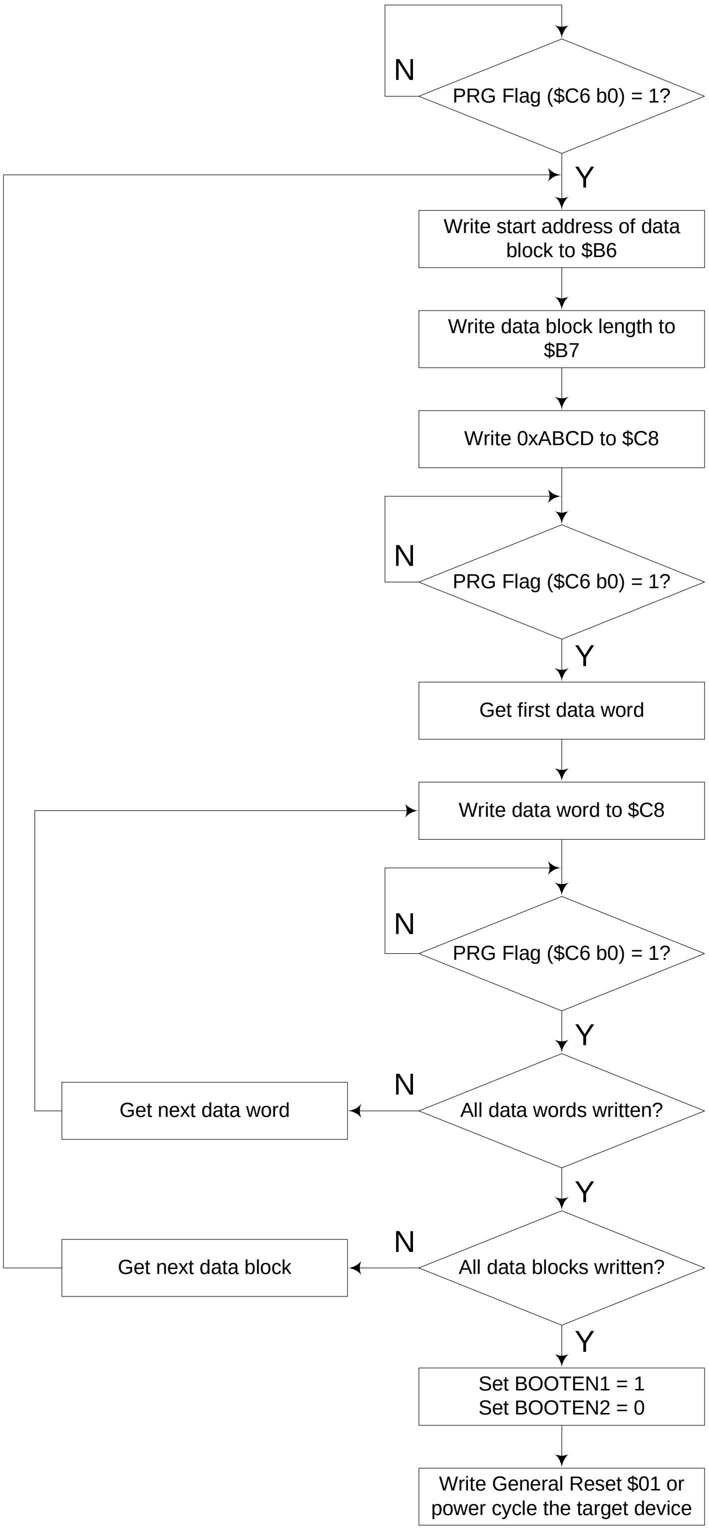

Figure 2. Program Flow for programming a Function Image^{TM} into serial memory using C-BUS

8 Conclusion

The ability of the CMX703x and CMX704x to accept data from the host processor via C-BUS and to program an external serial memory is helpful when implementing an ATE test system or for in-field firmware updates.
In an ATE system, the Function Image^{TM} can be downloaded to a product under test using high speed write to temporary flash storage. The product's host processor will then transfer the Function Image^{TM} to serial program memory, which exercises considerable portions of the product's system. This method also requires less interfacing to the ATE and reduced code development time.
The CMX703x and CMX704x device's bootloader and Thick Stub virtually eliminates 'fail-safe firmware update code' development effort and implementation costs.

CML does not assume any responsibility for the use of any algorithms, methods or circuitry described. No IPR or circuit patent licenses are implied. CML reserves the right at any time without notice to change the said algorithms, methods and circuitry and this product specification. CML has a policy of testing every product shipped using calibrated test equipment to ensure compliance with this product specification. Specific testing of all circuit parameters is not necessarily performed.

CML Microcircuits (UK) Ltd *COMMUNICATION SEMICONDUCTORS*	**CML Microcircuits (USA) Inc.** *COMMUNICATION SEMICONDUCTORS*	**CML Microcircuits (Singapore) Pte Ltd** *COMMUNICATION SEMICONDUCTORS*	
		Singapore	**China**
Tel: +44 (0)1621 875500 **Fax:** +44 (0)1621 875600 **Sales:** sales@cmlmicro.com **Tech Support:** techsupport@cmlmicro.com	**Tel:** +1 336 744 5050 800 638 5577 **Fax:** +1 336 744 5054 **Sales:** us.sales@cmlmicro.com **Tech Support:** us.techsupport@cmlmicro.com	**Tel:** +65 67450426 **Fax:** +65 67452917 **Sales:** sg.sales@cmlmicro.com **Tech Support:** sg.techsupport@cmlmicro.com	**Tel:** +86 21 6317 4107 +86 21 6317 8916 **Fax:** +86 21 6317 0243 **Sales:** cn.sales@cmlmicro.com.cn **Tech Support:** sg.techsupport@cmlmicro.com
- www.cmlmicro.com -			