Creating an LPC valid image checksum directly in the image

As described in the FAQ "LPC Image Checksums", NXP LPC MCUs use a word in the vector table of the CPU to store a checksum that is examined by the bootloader to identify that the code in Flash is a valid image.

This word is set automatically when downloading an AXF image using the LPCXpresso flash drivers. But if a binary image is to be programmed via other means then, in LPCXpresso IDE, v7.8.0 and earlier, a utility needs to be run as a post-build step to patch the appropriate value into the binary.

However, LPCXpresso IDE v7.9.0 and later provide a mechanism to allow the checksum value to be directly written into the (AXF) image at build time - thus removing the need for the post build step. This is achieved in two steps:

Startup code

The startup code generated by LPCXpresso IDE 7.9.0 and later has been modified so that the value of the linker symbol **__valid_user_code_checksum** is placed in the appropriate location of the vector table.

```
WEAK extern void __valid_user_code_checksum(); // <- *** NEW IN LPCXPRESSO 7.9.0

 :

 :

void (* const g_pfnVectors[])(void) = {

    // Core Level - CM3

    &_vStackTop,                    // The initial stack pointer

    ResetISR,                       // The reset handler

    NMI_Handler,                    // The NMI handler

    HardFault_Handler,              // The hard fault handler

    MemManage_Handler,              // The MPU fault handler

    BusFault_Handler,               // The bus fault handler

    UsageFault_Handler,             // The usage fault handler

    __valid_user_code_checksum,     // LPC MCU Checksum <- *** NEW IN LPCXPRESSO 7.9.0

    0,                              // Reserved

 :

 :
```

This means that if you have an existing project created before LPCXpresso IDE v7.9.0, you will need to update the startup file to use the latest version.

Also note that the vector tables entries for Cortex-M0/M0+ parts will differ slightly from that displayed above, as they have few defined fields in the vector table.

Linker script

The LPCXpresso IDE v7.9.0 manged linker script mechanism has been enhanced so that it generates the appropriate value for the **__valid_user_code_checksum**, which is then embedded into the generated image by the startup code (as above). This value is actually calculated from the values of the proceeding vector table entries:

```
PROVIDE(__valid_user_code_checksum = 0 -

     (_vStackTop

      + (ResetISR + 1)

      + (NMI_Handler + 1)

      + (HardFault_Handler + 1)

      + (( DEFINED(MemManage_Handler) ? MemManage_Handler : 0 ) + 1)   /* MemManage_Handler may not be defined */

      + (( DEFINED(BusFault_Handler) ? BusFault_Handler : 0 ) + 1)     /* BusFault_Handler may not be defined */

      + (( DEFINED(UsageFault_Handler) ? UsageFault_Handler : 0 ) + 1) /* UsageFault_Handler may not be defined */

      ) );
```

Note that the definition of __valid_user_code_checksum in the linker script generated for Cortex-M0/M0+ parts will differ slightly from that displayed above, as they have fewer defined entries in the vector table.

Notes

- The new checksum mechanism will only function if the project's start up file places the __valid_user_code_checksum at the appropriate location in the vector table AND the managed linker script mechanism is enabled for the project.   The absence of __valid_user_code_checksum in the vector table will not cause a problem, but the linker will be unable to automatically place the checksum into the image. Similarly, there will not be a problem if **__valid_user_code_checksum** is defined, but the managed linker script mechanism is not used.
- The new checksum mechanism is only supported for Cortex-M projects. For ARM7/9 based MCUs, the separate checksum utlity must still be used to patch the binary.