This is simple example of the usage of Flash Array Integrity Check (FAIC) function that is available on all MPC56xx devices.

The FAIC reads data from selected and unlocked flash blocks and calculates the MISR signature. User can compare the MISR signature calculated by flash controller (in runtime) with MISR calculated by offline tool (this is done during development, not in runtime) from s-record file. If the MISR is identical, we know that the content of selected flash blocks corresponds to content in s-record file and that there are no ECC errors (single bit or double bit ECC errors).

Note: Exception is not triggered if double bit ECC errors are discovered by FAIC. Just the MISR value will be different.

There are two types of Array Integrity Sequence. The default sequence (AIS = 0) is meant to replicate sequences normal “user” code follows, and thoroughly checks the read propagation paths. This sequence is proprietary. The alternative sequence (AIS = 1) is just logically sequential. It should be noted that the time to run a sequential sequence is significantly shorter than the time to run the proprietary sequence.

The FAIC as well as offline MISR calculator are implemented in SSD flash drivers provided by Freescale – can be found on www.freescale.com

Description of SW example:

SW example is written for MPC5604B device but the principle is the same for all other devices.
Let’s say that I want to check data in all blocks of Code Flash except block L0. This block will contain MISR values calculated by offline tool, so we can’t run the FAIC on this block – the MISR value would change if we place new MISR values there.
Linker file is modified, so all the code and data of the project are placed to L0 block only. I have placed just some constant to the rest of flash, so the s-record file is not empty. The offline tool expects that the rest of flash (beside data in s-record file) is in erased state.

Now it is necessary to run the offline tool and calculate the MISR values. See the “bin” folder in sample project where I copied MISR_C90LC.exe from SSD drivers and where I created test.bat file.
The batch file contains these commands:

```
MISR_C90LC.exe -t CF -m 0x220 -ea HM -as 0 -lo 0x3E -mi 3 -hi 0 -ba 0 -i 0 -bi 0 -s internal_FLASH.mot -o output_as0.txt

MISR_C90LC.exe -t CF -m 0x220 -ea HM -as 1 -lo 0x3E -mi 3 -hi 0 -ba 0 -i 0 -bi 0 -s internal_FLASH.mot -o output_as1.txt
```

See the readme.txt file provided together with exe file for details. Both commands select all flash blocks except L0 to be checked. One command select Proprietary sequence, second Sequential sequence. The rest of parameters just say which device we are using. It is necessary to specify the FLASH_MCR register value, ECC algorithm (Hamming code in case of MPC5604B) and so on.
Then we also specify s-record file that contains the data and output file. Generated output files then contain calculated MISR values. We can take these values and we can put them to our project.

If we want to run the FAIC on our device, it is necessary:

- Run the FlashInit function
- Unlock all blocks that will be checked by FAIC
- Run the FAIC

If the FAIC pass, it will return C90FL_OK. That means data in s-record file correspond to data in flash.

For test purposes, it is possible to change any data in flash. The FAIC then will not pass and the function will return C90FL_ERROR_AIC_MISMATCH.
It also means that anytime the data or code in flash are changed (for example during development), it is necessary to calculate new MISR values using offline tool.