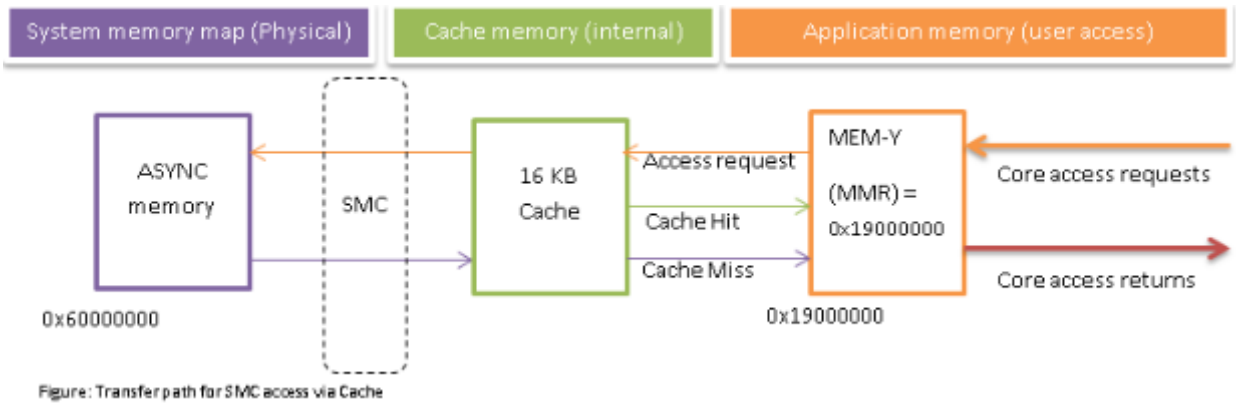


Figure : Transfer path for SMC access via Cache

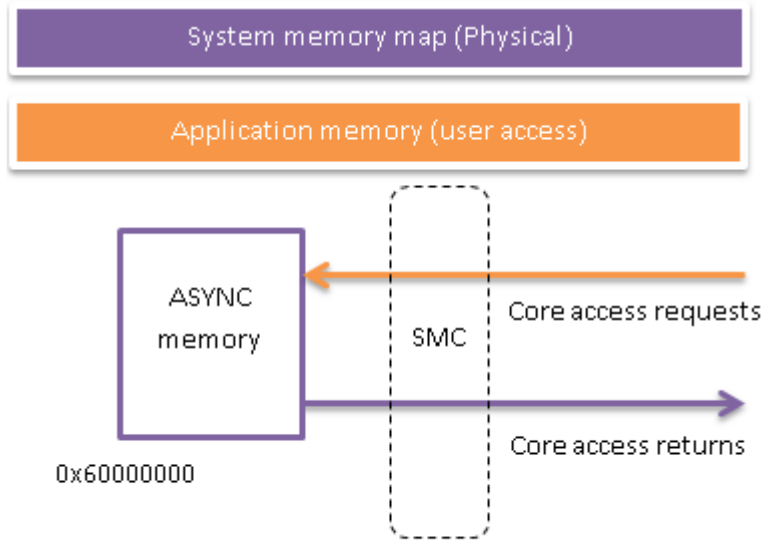
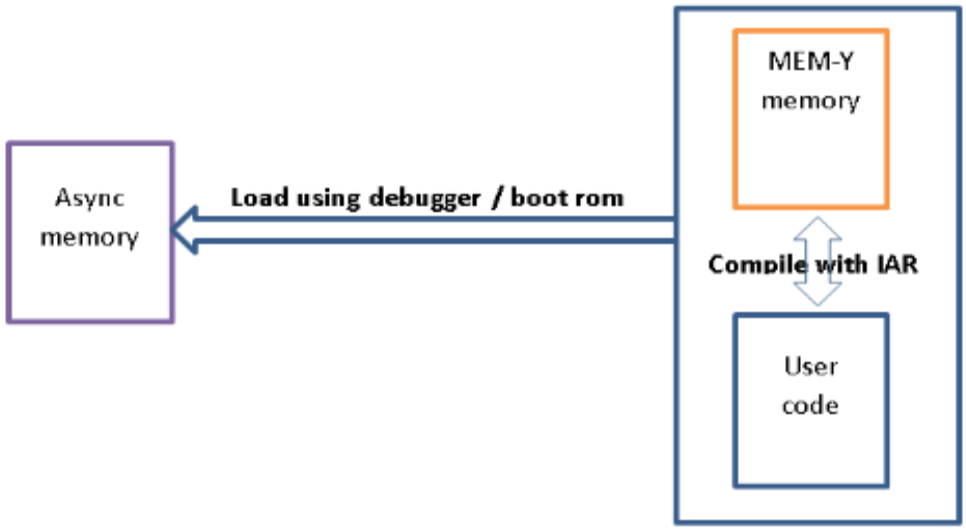


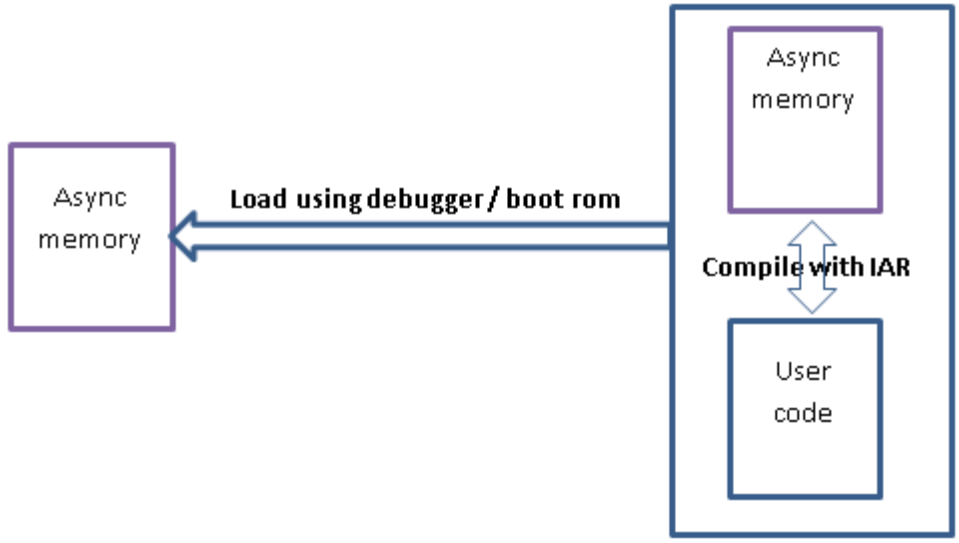
Figure: Transfer path for SMC access without using Cache

Developing application

Code building and loading for case of Cache



Code building and loading for case without Cache



Developing SMC connected SRAM based projects without using Cache

SMC connected SRAM can be accessed directly, similar to internal SRAM. Here is an example Script that can be dded in IAR ICF

```
// setup linker file
define symbol ASYNC_BASE         = 0x60000000;
define symbol ASYNC_SIZE         = 0x00080000;
define region ASYNC_code_region   = mem:[from ASYNC_BASE  size ASYNC_SIZE ];
place in ASYNC_code_region        { section .async_code};
```

```
// test code mapped in SMC SRAM
#pragma location = ". async_code "
void CallSomeCode() {...}
```

Developing SMC connected SRAM based projects using Cache

As mentioned in the first page, developing SRAM based projects that support Cache has requirement of: different compile memory (0x1900_0000) and load memory (0x6000_0000). IAR Tools support a way to assist code development in this scenario. The idea is to inform the Linker that the code for MEMY locations are made as 'initialize manually', but do not perform the actual initialization (memory translation in hardware takes care of that). Then, have the initialization source copy inside the Async memory itself that is retained for the internal access.

```
// setup linker file
define symbol ASYNC_BASE          = 0x60000000;
define symbol ASYNC_SIZE          = 0x00080000;
define symbol MEMY_BASE           = 0x19000000;
define symbol MEMY_SIZE           = 0x01000000;
define region MEMY_code_region     = mem:[from MEMY_BASE  size ASYNC_SIZE ];
define region MEMY_INIT           = mem:[from ASYNC_BASE size ASYNC_SIZE];
place in MEMY_code_region         { section .memy_code };
place in MEMY_INIT                { section .memy_code_init };
initialize manually               { section .memy_code };
```

```
// test code mapped in SMC SRAM
#pragma location = ".memy_code"
void CallSomeCode() {...}
```

```
// check the Map file after link
"P4":                                0x3c
 P4 s2                  0x19000000    0x3c  <Init block>
   .memy_code       inited   0x19000000    0x3c  Blinky.o [1]
                  - 0x1900003c    0x3c
"P5":                                0x3c
 Initializer bytes    ro data  0x60000000    0x3c  <for P4 s2>
                  - 0x6000003c    0x3c
```

Figure: Test example screenshot from IAR