Format A and B cache unlock procedure

To unlock the locked down portion of the cache, write to the CP15 c9 register with:

- WAY == 0, for Formats A and B
- L == 0, for Format B only.

Format C lockdown

Cache lockdown Format C is a different form of cache way based locking. It enables the allocation to each cache way to be disabled or enabled. This provides some additional control over the cache pollution caused by particular applications, in addition to a traditional lockdown function for locking critical regions into the cache.

A locking bit for each cache way determines whether the normal cache allocation mechanisms can access that cache way.

For caches of higher associativity, only cache ways 0 to 31 can be locked.

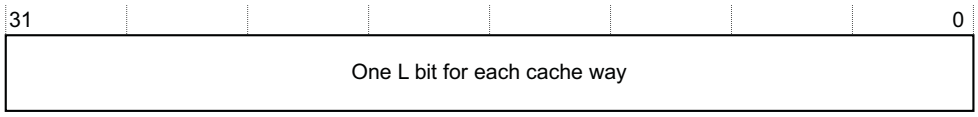
A maximum of N-1 ways of an N-way cache can be locked. This ensures that a normal cache line replacement can be performed. Handling a cache miss is UNPREDICTABLE if there are no cache ways that have L==0.

The 32 bits of the lockdown register determine the L bit for the associated cache way. The value of <opc2> determines whether the instruction lockdown register or data lockdown register is accessed.

The cache lockdown register is normally modified in a read, modify, write sequence. For example, the following sequence sets the L bit to 1 for way 0 of the instruction cache:

```
; In the following code, <Rn> can be any register whose value does not need to be kept.
        MRC     p15, 0, <Rn>, c9, c0, 1    ; Read Instruction Cache Lockdown Register
        ORR     <Rn>, <Rn>, #0x01
        MCR     p15, 0, <Rn>, c9, c0, 1    ; Write Instruction Cache Lockdown Register
                                           ; The write sets way 0 L bit for the instruction cache
```

The Format C lockdown register bit assignments are:



Bits[31:0] The L bits for each cache way. If a cache way is not implemented, the L bit for that way is RAO/WI. Each bit relates to its corresponding cache way, that is bit N refers to way N.

0 Allocation to the cache way is determined by the standard replacement algorithm (reset state)

1 No Allocation is performed to this cache way.

The Format C lockdown register must only be changed when it is certain that all outstanding accesses that can cause a cache linefill have completed. For this reason, a Data Synchronization Barrier instruction must be executed before the lockdown register is changed.

Format C cache lock procedure

The procedure for locking down into a cache way i with N cache ways using Format C involves making it impossible to allocate to any cache way other than the target cache way i. The architecture defines the following method for locking data into the caches:

1. Ensure that no processor exceptions can occur during the execution of this procedure, for example by disabling interrupts. If for some reason this is not possible, all software and data used by any exception handlers that can get called must be treated as software and data used by this procedure for the purpose of steps 2 and 3.

2. If an instruction cache or a unified cache is being locked down, ensure that all the software executed by this procedure is in an Non-cacheable area of memory, including the Tightly Coupled Memory, or in an already locked cache way.

3. If a data cache or a unified cache is being locked down, ensure that all data used by the following software (apart from the data that is to be locked down) is in an Non-cacheable area of memory, including the Tightly Coupled Memory, or is in an already locked cache way.

4. Ensure that the data or instructions that are to be locked down are in a Cacheable area of memory.

5. Ensure that the data or instructions that are to be locked down are not already in the cache, using cache clean, invalidate, or clean and invalidate instructions as appropriate.

6. Write to the CP15 c9 register with CRm == 0, setting L=0 for bit i and L=1 for all other bits. This enables allocation to the target cache way i.

7. For each of the cache lines to be locked down in cache way i:
   - If a data cache or a unified cache is being locked down, use an `LDR` instruction to load a word from the memory cache line. This ensures that the memory cache line is loaded into the cache.
   - If an instruction cache is being locked down, use the CP15 c7 prefetch instruction cache line operation to fetch the memory cache line into the cache.

8. Write to the CP15 c9 register with CRm == 0, setting L = 1 for bit i and restoring all the other bits to the values they had before this routine was started.

Format C cache unlock procedure

To unlock the locked down portion of the cache, write to the CP15 c9 register, setting L == 0 for each bit.

Format D lockdown

This format locks individual L1 cache line entries rather than using a cache way scheme. The methods differ for the instruction and data caches.

The instructions that access the CP15 c9 Format D Cache Lockdown Registers and operations are as follows:

```
MCR p15, 0, <Rt>, c9, c5, 0    ; fetch and lock instruction cache line,
                               ; Rt = MVA
MCR p15, 0, <Rt>, c9, c5, 1    ; unlock instruction cache,
                               ; Rt ignored
MCR p15, 0, <Rt>, c9, c6, 0    ; write Format D Data Cache Lockdown Register,
                               ; Rt = set or clear lockdown mode
MRC p15, 0, <Rt>, c9, c6, 0    ; read Format D Data Cache Lockdown Register,
                               ; Rt = lockdown mode status
MCR p15, 0, <Rt>, c9, c6, 1    ; unlock data cache,
                               ; Rt ignored
```

Note

Some format D implementations use CRm == {c1, c2} instead of CRm == {c5, c6}. You must check the Technical Reference Manual to find the encoding uses. The architecture did not require the implementation of CP15, and the Architecture Reference Manual only gave a recommended implementation. The actual CP15 implementation is IMPLEMENTATION DEFINED in ARMv4 and ARMv5.

The following rules determine how many entries in a cache set can be locked:

- At least one entry per cache set must be left for normal cache operation, otherwise behavior is UNPREDICTABLE.
- How many ways in each cache set can be locked is IMPLEMENTATION DEFINED. MAX_CACHESET_ENTRIES_LOCKED < NWAYS.
- Whether attempts to lock additional entries in Format D are allocated as an unlocked entry or ignored is IMPLEMENTATION DEFINED.

For the instruction cache, a fetch and lock operation fetches and locks individual cache lines. Each cache line is specified by its MVA. To lock instructions into the instruction cache, the following rules apply:

- The routine that locks lines into the instruction cache must be executed from Non-cacheable memory.
- The memory that holds the instructions being locked into the instruction cache must be Cacheable.
- The instruction cache must be enabled and invalidated before locking down cache lines.

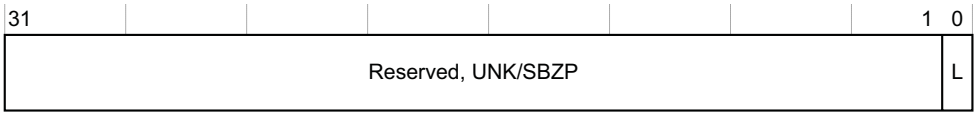
If these rules are not applied, results are UNPREDICTABLE. Entries must be unlocked using the global instruction cache unlock command.

Cache lines must be locked into the data cache by first setting a global lock control bit. Data cache linefills occurring while the global lock control bit is set are locked into the data cache. To lock data into the data cache, the following rules apply:

- The data being locked must not exist in the cache. Cache clean and invalidate operations might be necessary to meet this condition.
- The data to be locked must be Cacheable.
- The data cache must be enabled.

CP15 c9, Format D Data or unified Cache Lockdown Register, DCLR2, ARMv4 and ARMv5

The DCLR2, the Format D Data or unified Cache Lockdown Register, bit assignments are:



L, bit[0] Lock bit

0 no locking occurs

1 all data fills are locked while this bit is set.

Interaction with CP15 c7 operations

Cache lockdown only prevents the normal replacement strategy used on cache misses choosing to reallocate cache lines in the locked down region. CP15 c7 operations that invalidate, clean, or clean and invalidate cache contents affect locked down cache lines as normal. If invalidate operations are used, you must ensure that they do not use virtual addresses or cache set/way combinations that affect the locked down cache lines. Otherwise, if it is difficult to avoid affecting the locked down cache lines, repeat the cache lockdown procedure afterwards.

O.7.12 CP15 c9, TCM support

TCM register support is optional when CP15 and TCM are supported in ARMv4 and ARMv5. For details see *CP15 c9, TCM support* on page AppxL-2538.