

5.8.1 Dynamic read allocate mode

When a memory region is marked as Write-Back Write-Allocate, it normally allocates a cache line on either a read miss or a write miss. However, there are some situations where allocating on writes is undesirable, such as executing the C standard library `memset()` function to clear a large block of memory to a known value. Writing large blocks of data like this can pollute the cache with unnecessary data. It can also waste power and performance if a linefill must be performed only to discard the linefill data because the entire line was subsequently written by the `memset()`.

To prevent this, the Cortex-M7 *Bus Interface Unit* (BIU) includes logic to detect when a full cache line has been written by the core before the linefill has completed. If this situation is detected on three consecutive linefills, it switches into dynamic read allocate mode. When in dynamic read allocate mode, loads behave as normal and can still cause linefills, and writes still lookup in the cache but, if they miss, they write out to external memory rather than starting a linefill.

The BIU continues in dynamic read allocate mode until it detects either a cacheable write burst to external memory that is not a full cache line, or there is a load to the same line as is currently being written to external memory.

Dynamic read allocate mode can be disabled by setting the `ACTLR.DISRAMODE` to 1. See [Auxiliary Control Register on page 3-6](#).