Whitepaper

Understanding Write Combining on Arm

Version 1.0
Released 2 November 2020

Write Combining (WC) is a specialized memory type defined by the x86-64 architecture that is used for gathering multiple stores into burst transactions over the system bus. WC is commonly used on x86-64 platforms for interaction with I/O and other peripheral devices [1].

In this whitepaper we provide an overview of the Arm architecture memory types that provide WC-like capabilities. In addition, the whitepaper provides advice for Arm system implementers on providing compatibility with existing hardware leveraging WC capabilities for performance optimizations.

This whitepaper is intended for Arm SoC and system components designers and system software developers.

Memory Gathering Capabilities on Arm

The Arm architecture defines two memory types that don't use caches and support gathering capabilities: Normal Non-Cacheable (Normal NC) memory and Device memory with GRE attributes (Device-GRE). The GRE attribute corresponds to the conjunction of the following properties: Gathering, Reordering, and Early write acknowledgment. Detailed information about the Arm architecture memory types can be found in the Arm specification [2], section B2.7.

In this whitepaper we focus on the gathering attribute which permits the following behavior, as it is defined by the Arm architecture [2], section B2.7.2:

"Multiple memory accesses of the same type, read or write, to the same memory location to be merged into a single transaction.

Multiple memory accesses of the same type, read or write, to different memory locations to be merged into a single memory transaction on an interconnect."

Normal NC is a relaxed order memory type that provides gathering capabilities, allows speculation over the memory, and does not impose any alignment constraints. Device-GRE is also a relaxed order memory type allowing gathering, but it disallows read speculation and imposes strict alignment constrains. Table 1 summarizes the key differences between the memory types.

Table 1: An overview of memory gathering capabilities across architectures. Y - Yes, N – No, micro-arch – depends on a micro architecture

	X86-64 WC	Arm Normal NC	Arm Device-GRE
Relaxed Order	Y	Y	Y
Gathering	Y	Y	Y
Supports Read Speculation	Y	Y	N
Supports Unaligned Access	Y	Y	N
Maximum Gathering Size	64 Byte	micro-arch	micro-arch

Based on the above table, one can see that Normal NC characteristics are identical to the x86-64 WC memory type. Device-GRE memory type has similar gathering properties, but since it does not support unaligned accesses, it cannot serve as a direct replacement for x86 WC on Arm. In addition, Device-GRE does not support read speculation nevertheless the read speculation capability has no impact on software functional correctness. It is important to point out that software that guarantees aligned access can use Device-GRE memory as a replacement for x86-64 WC. It is advised that driver developers verify that the target OS provides kernel APIs for allocation Normal NC and Device-GRE memory. For example, current Linux kernel (5.5-rc7) do not expose kernel APIs for Device-GRE memory allocation.

Another critical characteristic of memory gathering is the partial write and maximum gathering sizes.

X86-64 WC can potentially gather continuous sequences of stores into 64 Byte blocks, which is the maximum gathering size for current x86 implementations. It is important to note that gathering is an opportunistic optimization and under certain circumstances may not be successful, which will result in partial writes of 8 Byte chunks [1] on a system bus. Peripheral devices cannot exclusively rely on WC gathering functionality and must be fully functional in the presence of partial writes.

The Arm architecture does not define the partial write or maximum gathering sizes for Normal NC or Device-GRE memories (Table 1). Normal NC or Device-GRE memory regions can represent physical memories that reside on system components that are outside of the scope of the Arm architecture and depend on the underlying system design and implementation. PCIe root complex and PCIe endpoints are examples of such system components; PCIe Base Address Registers (BARs) can be mapped as a Normal NC or Device-GRE memory type in the Linux kernel. The Arm architecture defines single copy atomicity rules for naturally aligned memory access of 4 bytes and above [2] (section B2.2). These rules combined with upcoming Server Base System Architecture specification provides mapping of single copy atomicity rules to the PCIe bus transaction protocol [3]. A combination of the two specifications provides device drivers with mechanisms to control PCIe transaction size.

To enable performance-portable and efficient interaction with peripheral devices that leverage gathering capability for performance optimizations, it is advised for Arm system implementations to use gathering semantics as defined by the Arm architecture. For a continuous sequence of stores to Normal NC or Device-GRE memory, the implementation can gather up to the limits defined by microarchitecture constraints with the goal to increase the efficiency of the transactions on the system bus and PCIe. Device drivers can use the Data Gathering Hint (DGH) (see section C6.2.7.9) or one of the memory barrier instructions [2] (section E2.7.2) to notify the underlying hardware about the intent to stop the gathering.

Enabling Memory Gathering in Software

This is an example of 64 Byte copy from q0-q3 registers to the destination address referenced in *x0*. When the destination address is mapped as Normal NC the micro-architecture is expected to trigger the gathering flow which will result in a 64 Byte transaction on the bus. The `dgh` instruction is used to notify the CPU about the intent to stop gathering.

```
str     q3, [x0]

str     q2, [x0,#16]

str     q1, [x0,#32]

str     q0, [x0,#48]

dgh
```

Summary

The Arm architecture defines two memory types that provide memory gathering capabilities – Normal NC and Device-GRE. Out of the two memory types, Normal NC memory characteristics are identical to the x86-64 WC memory type. Memory gathering is an opportunistic optimization and under certain circumstances may not be successful. Peripheral devices cannot exclusively rely on gathering functionality and must be fully functional in the presence of partial writes. Arm's DGH instruction provides a light-weight mechanism for notifying the CPU about the intent to stop gathering.

References

[1] **https://software.intel.com/sites/default/files/managed/39/c5/325462-sdm-vol-1-2abcd-3abcd.pdf** *Vol. 3A 11-8*

[2] ARMv8-A Reference Manual, (Issue F.a)

[3] SBSA Specification, **https://developer.arm.com/architectures/platform-design/server-systems**