

To avoid any significant impact on performance because of multiple targets, FlashCopy creates dependencies between the targets. Dependencies can be considered as “hidden” FlashCopy mappings that are not visible to and cannot be managed by the user. A dependency is created between the most recent target and the previous one (in order of start time). Figure 11-13 shows an example of a source volume with three targets.

When the three targets are started, Target T0 was started first and considered the “oldest.” Target T1 was started next and is considered “next oldest,” and finally Target T2 was started last and considered the “most recent” or “newest.” The “next oldest” target for T2 is T1. The “next oldest” target for T1 is T0. T1 is newer than T2, and T0 is newer than T1.

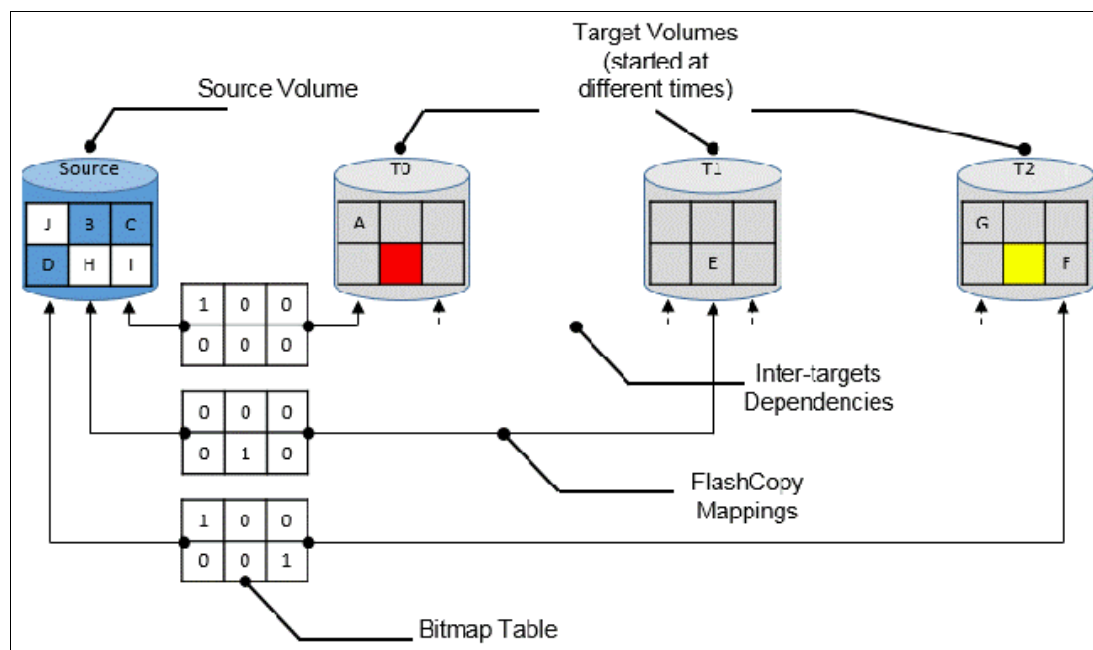


Figure 11-13 FlashCopy dependencies example

Source read with Multiple target FlashCopy

There is no specific behavior for read operations on source volumes when there are multiple targets for that volume. The data is always read from the source.

Source write with Multiple Target FlashCopy (Copy on Write)

A write to the source volume does not cause its data to be copied to all of the targets. Instead, it is copied to the most recent target volume only. For example, consider the sequence of events that are listed in Table 11-3, for a source volume and three targets started at different times. In this example, there is no background copy. The “most recent” target is indicated with an asterisk.

Table 11-3 Sequence example of write IOs on a source with multiple targets

	Source Volume	Target T0	Target T1	Target T2
Time 0: mapping with T0 is started	A B C D E F	___* ___	Not started	Not started
Time 1: change of “A” is made on source (->“G”)	G B C D E F	A __* ___	Not started	Not started
Time 2: mapping with T1 is started	G B C D E F	A __ ___	___* ___	Not started