SigmaQuad Type I vs. Type II Timing Comparison

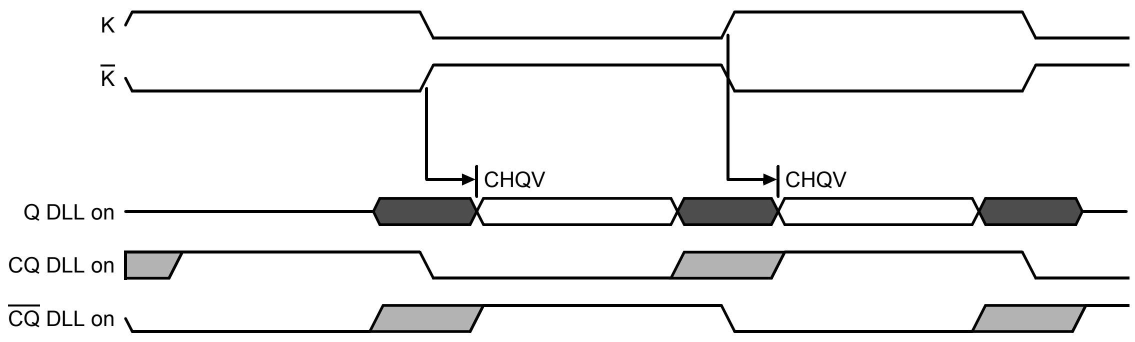
Introduction

SigmaQuad-II SRAMs implement a DLL (Delay Locked Loop). The DLL provides a larger data valid window by synchronizing the output data to the input clocks, C and $\overline{C}$ or K and $\overline{K}$, if C and $\overline{C}$ are tied high. SigmaQuad- II SRAMs have a Type I option, but this requires the DLL to be disabled and will be discussed in detail later.

Timing Difference Between SigmaQuad Type I and Type II

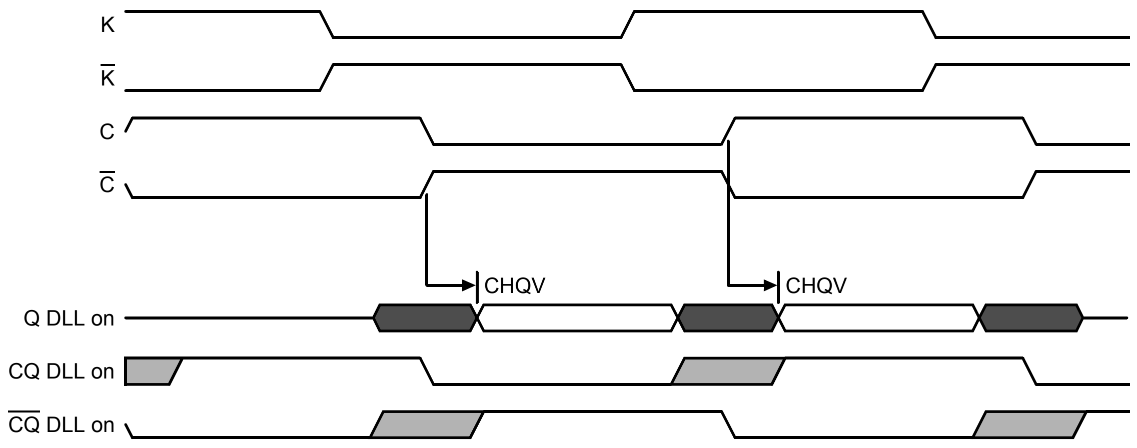
As previously mentioned, the SigmaQuad-II SRAMs have a DLL implemented that synchronizes the output data with the input clocks. The input clocks are K and $\overline{K}$, if C and $\overline{C}$ are tied high, or C and $\overline{C}$. The following figure shows a simplified output timing diagram for the SigmaQuad-II with C and $\overline{C}$ tied high.

DLL Turned On (C and $\overline{C}$ Tied High)



The following figure shows a simplified timing diagram for the SigmaQuad-II and includes the use of C and $\overline{C}$.

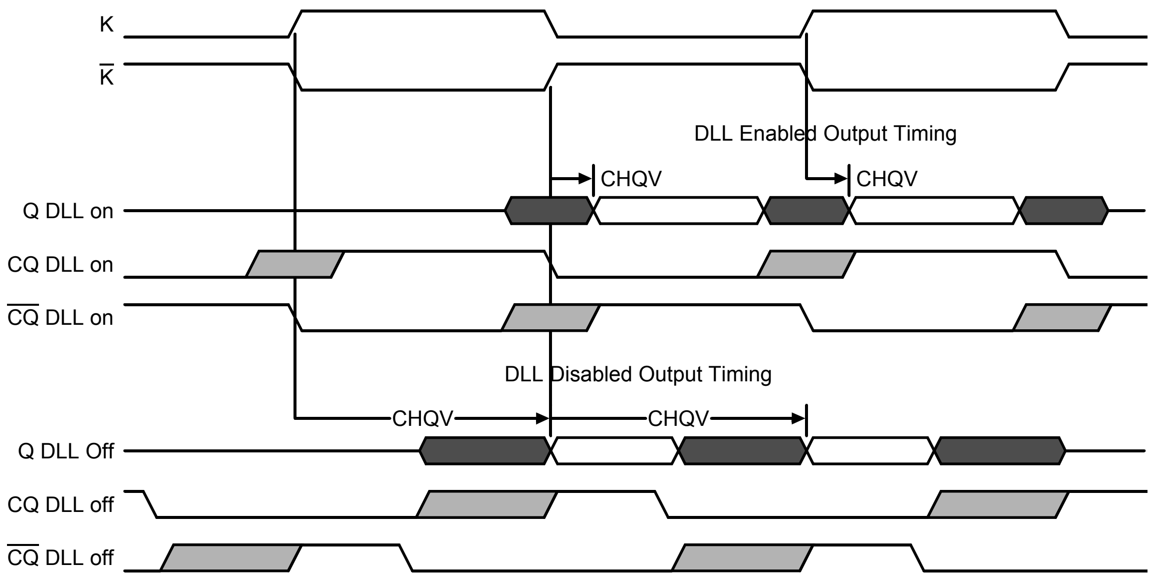
DLL Turned On (C and $\overline{C}$ active with delay)



Type II Timing with DLL Enabled vs. Disabled

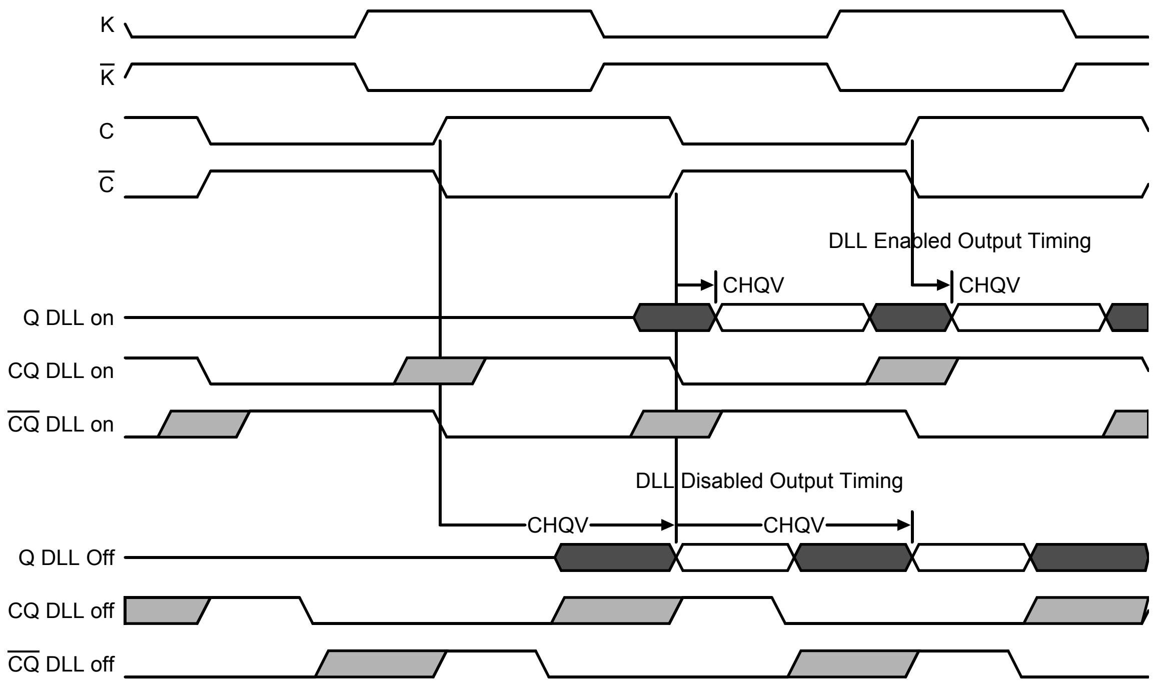
The SigmaQuad-II SRAMs have the option for turning off the DLL using the $\overline{\text{Doff}}$ pin (ball location H1). The following figure shows the changes in output timing when the DLL is enabled and when the DLL is disabled, with C and $\overline{\text{C}}$ tied high.

DLL On vs. Off with C and $\overline{\text{C}}$ Tied High


K
$\overline{\text{K}}$
DLL Enabled Output Timing
CHQV
CHQV
Q DLL on
CQ DLL on
$\overline{\text{CQ}}$ DLL on
DLL Disabled Output Timing
CHQV
CHQV
Q DLL Off
CQ DLL off
$\overline{\text{CQ}}$ DLL off

When the DLL is enabled and is locked to a frequency, the clock-to-data valid time is relatively short. When the DLL is disabled, the Type II timing will revert to a SigmaQuad Type I timing. The previous fiugure shows the changes in timing with the DLL enabled and disabled. As seen in the following figure, the data driven out of the SigmaQuad device is referenced to the rising edge of K or $\overline{\text{K}}$, if C and $\overline{\text{C}}$ are tied high. The clock to data valid of the SigmaQuad device with the DLL disabled, t_{KHQV} and t_{CHQV}, is significantly larger than the when the DLL is enabled. Also, when the DLL is disabled the AC timing specifications are no longer guaranteed. An example of DLL enabled vs. disabled with the use of C and $\overline{\text{C}}$ is shown in the following figure.

DLL On vs. Off with C and $\overline{C}$ Active



DLL Constraints

There are a few constraints to keep in mind in order for the DLL to function properly. The first is phase jitter. Phase jitter is the maximum allowed cycle-to-cycle variation of the rising edges of the input clocks. This is defined in the datasheet as t_{KCvar}. If the t_{KCvar} spec is violated too much then the DLL has the potential to become unlocked. See Application Note AN1014 for additional information on t_{KCvar} specifications.

The next constraint is DLL lock time. Once the input clocks to the SRAM have become stable it takes 1K clock cycles, 64K clock cycles for ECCRAMs, before the DLL is able to lock onto the operating frequency. It is ideal to have the voltage supplies DC stable before supplying the inputs clocks to the SRAM. Also, another thing to keep in mind is that if the incoming clocks are not DC stable (i.e., with too much cycle-to-cycle variation), the DLL may lock onto the wrong frequency. To avoid this potential undesired issue, use the $\overline{\text{Doff}}$ pin to turn off the DLL until the clocks have stabilized at the desired operating frequency. Then turn the DLL on and the DLL will lock onto the frequency within 1K cycles, 64K cycles for ECCRAM.

When the DLL is enabled and the input cycle time is greater than the DLL maximum cycle time ($t_{KHKHmax}$), the datasheet specification for output clock to data valid timing is no longer guaranteed. For DLL input cycle times that are above the max $t_{KHKHmax}$ specification and less than 50 ns, the clock-to-data valid spec will be somewhere between 2 ns and 16 ns and will vary cycle to cycle. Once the DLL input cycle time is greater than 50 ns, the clock-to-data valid specification is around 2 ns, 6 ns for the ECCRAMs. Also, under these conditions the data driven out is still synchronized to the rising edge of K or $\overline{K}$, if C and $\overline{C}$ are tied high.