

Synchronizing Multiple AD9777s Using DATACLK Input Mode

By Steve Reine

As stated in its data sheet, there are two primary methods for clocking the AD9777. The first is with the PLL enabled, where the clock applied to the differential clock inputs is the same speed as the input data rate. The internal PLL then performs the internal clock multiplication needed to operate the interpolation filters and DAC. The second method is to disable the PLL and to apply an external clock at the DAC output sample rate. In this mode, internal dividers provide the clocks needed for the internal digital filters and also generate the DATACLK and ONEPORTCLK clocks needed for external data synchronization.

For applications with the most stringent noise requirements, the AD9777 can be operated with the internal PLL disabled. See the AD9777 data sheet for more details on performance in this mode. With the PLL disabled, the DATACLK output can be used to synchronize external data at the AD9777 digital inputs. Setup and hold times for this mode are given in the data sheet. This works very well when only a single AD9777 needs to be synchronized with external data. In applications where multiple AD9777s are used and their external digital data must be synchronized to a single clock, using DATACLK as an output will not work correctly. DATACLK out may fall on

one of several different clock edges of the differential input clock, and there is no way to synchronize multiple AD9777s to make sure that DATACLK out falls on the same edge for every part. To overcome this obstacle, the AD9777 can be programmed so that the DATACLK pin functions as an input. In this mode, the user drives the AD9777 with two clocks, the external differential DAC clock, as well as the external clock on the DATACLK pin. Note that this mode is available only when the PLL is disabled and the AD9777 is in two-port data mode. To set the device to this mode, program (through the SPI port) input register 02h, Bit 3, to a Logic 1.

As shown in Figure 1, an internal state machine is used to perform the synchronization between the DAC and data clocks. Synchronization occurs when a rising edge is applied to the state machine by way of the DATACLK pin. When this rising edge occurs in advance of a DAC clock rising edge by a time t_{SMS} , the input data is synchronized to the third following edge of the DAC clock. Figures 2a–2c show the timing in all three interpolation modes. Note the timing required between the DAC and data clock and that the set up and hold times of the digital inputs are with respect to the third following edge of the DAC clock.

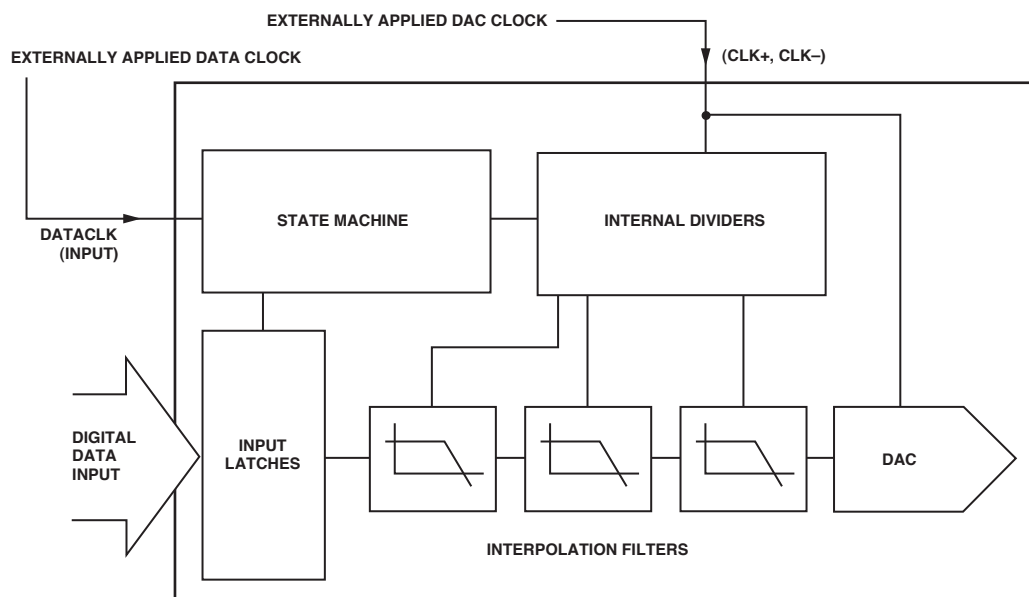


Figure 1. Driving the AD9777 with Separate DAC and Data Clocks Using DATACLK Input

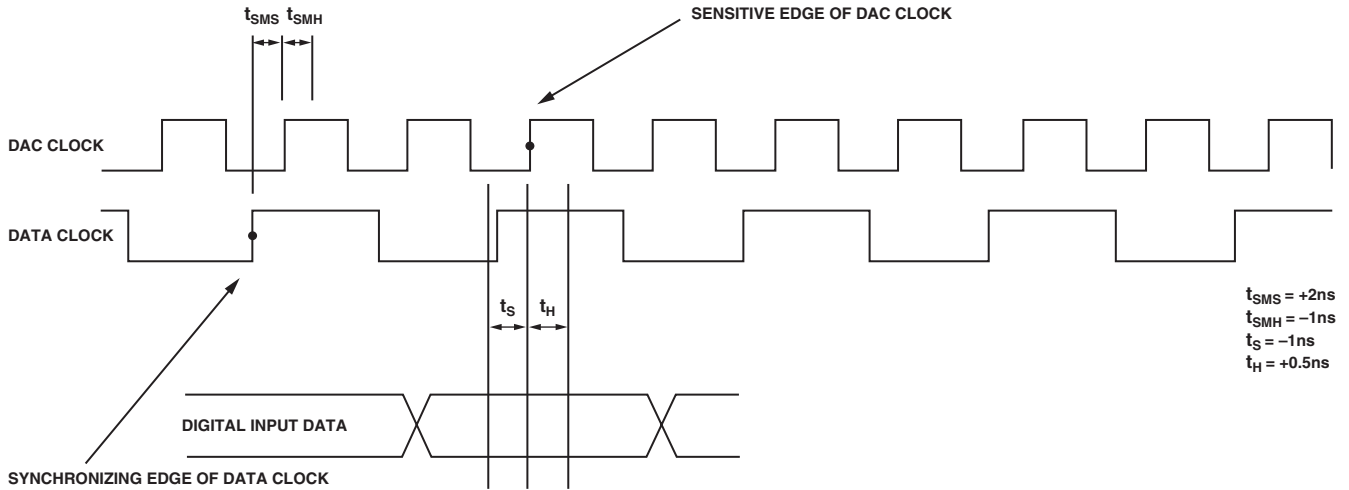


Figure 2a. Data Clock Input Mode Timing with 2× Interpolation

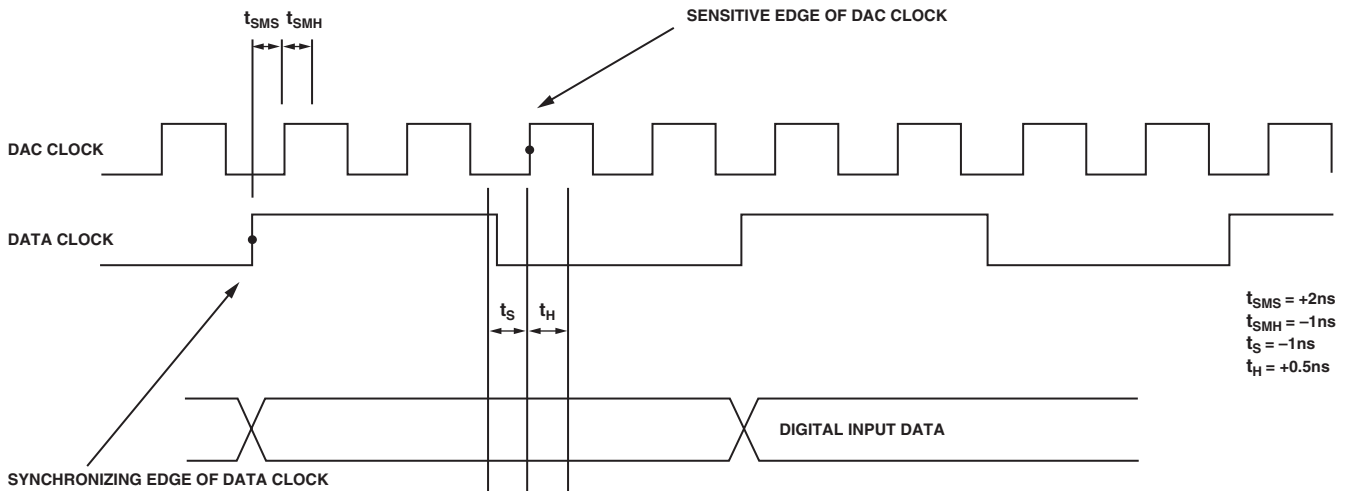


Figure 2b. Data Clock Input Mode Timing with 4× Interpolation

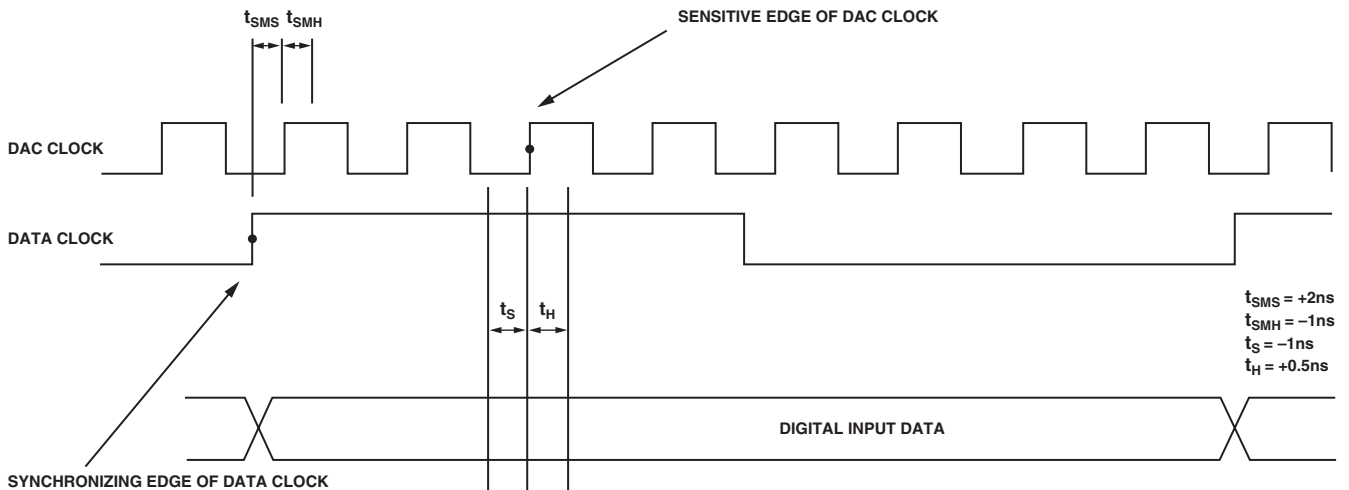


Figure 2c. Data Clock Input Mode Timing with 8× Interpolation

If the rising edge of the data clock occurs after the time t_{SMH} , the synchronization will be delayed by one cycle of the DAC clock.

An internal state machine controls the DAC latch timing in all modes of operation. This state machine is updated on every rising edge of the DAC clock. The output of the state machine updates the AD9777 input latches at a rate dependent on the DAC clock speed and the interpolation rate. With DATACLK programmed as an input, the state machine is reset on every rising edge of DATACLK. For this reason, if DATACLK were to stop, and the input data were to continue uninterrupted, the AD9777 may appear to be operating normally (i.e., state machine is still running). However, synchronization will likely be lost very quickly and the keep out windows defined by t_S , t_H , t_{SMS} , and t_{SMH} may shift in time randomly.

