Operating FIFOs On Full and Empty Boundary Conditions

TECHNICAL NOTE TN-08

by Suneel Rajpal and Frank Schapfel

The IDT7201, IDT7202, IDT7203 and IDT7204 (512 x 9, 1,024 x 9, 2,048 x 9 and 4,096 x 9) FIFOs have only four control lines: Read, Write, Reset and Retransmit. The focus of this tech note is the relation of the Read ($\overline{R}$) and Write ($\overline{W}$) lines to the FIFO's empty and full conditions.

These high-speed FIFOs can perform asynchronous and simultaneous read and write operations. $\overline{R}$ and $\overline{W}$ assert and deassert the Empty Flag ($\overline{EF}$) and Full Flag ($\overline{FF}$). Therefore, special conditions exist when a full FIFO continues to be written to and a read operation takes place. Also, special timings occur when an empty FIFO continues to be read to and a write operation takes place. These operations are called the FIFO boundary conditions.

Read and Write increment the read and write pointers on their respective rising clock edges. The read and write pointers affect the Empty Flag and Full Flag counters. The Empty Flag timings are shown in Figure 1. When the FIFO has only one word in it, the falling edge of $\overline{R}$ causes $\overline{EF}$ to be asserted. After the clock cycle is completed ($\overline{R}$ goes HIGH again), $\overline{EF}$ will remain asserted and the internal read counter is not affected by subsequent read cycles. $\overline{EF}$ is deasserted by the next rising edge of $\overline{W}$, after which another read pulse can be applied to do a read operation. In asynchronous systems, read and write operations take place at any time; $\overline{EF}$ is set by one signal and deasserted by another asynchronous signal.

When $\overline{R}$ is being clocked on an empty FIFO, the outputs will be in high-impedance. If a write operation is performed during asynchronous read cycles, a possible violation of the read pulse width minimum can occur, as shown in Figure 2. $\overline{EF}$ is deasserted, but there is an insufficient read pulse minimum width. To prevent the minimum read pulse width violation, initiate a read operation only after $\overline{EF}$ is HIGH, or guarantee a long enough read pulse width minimum time. A violation of the timing causes an internal glitch on the FIFO Read which can cause the read pointer to be "out of sync". Then the data inside the FIFO may be scrambled or may be garbage. The Empty Flag and Full Flag counters may also be upset by the internal glitch, which upsets FIFO memory usage. The only way to recover from this violation is to do a master reset.

A similar situation arises at the full FIFO boundary condition. When the FIFO is one word from being full, the falling edge of $\overline{W}$ causes the $\overline{FF}$ to be asserted. After the write cycle is completed ($\overline{W}$ goes HIGH again), $\overline{FF}$ will remain asserted and the internal write counter is not affected by subsequent write cycles. The $\overline{FF}$ flag is deasserted by the next rising edge of $\overline{R}$, as shown in Figure 3, after which another write pulse can be applied to do a write operation.

When the FIFO is full and $\overline{W}$ is being clocked, data sent to the FIFO will be ignored and the write pointer will not incre-ment. Here, as in the earlier case, if these write cycles are asynchronous during a read operation, a possible violation of the write pulse width minimum can occur, as shown in Figure 4. Here, $\overline{FF}$ is deasserted but a sufficient write pulse minimum width is not met. To prevent the problem, initiate a write operation only after $\overline{FF}$ is HIGH, or guarantee a long enough write pulse width minimum time. A violation of the timing causes an internal glitch on the FIFO write line. This can cause the write pointers to be "out of sync" where the data inside the FIFO may be scrambled or may be garbage. The Empty Flag and Full Flag counters may also be upset by the internal glitch. Again, the only way to recover from this condition is to do a master reset.

In summary, these FIFOs are designed to transfer only valid data from input to output. To ensure that valid data is written into and read from, empty and full FIFOs handshake through the flag mechanism. When there is no output data available, the reading side must wait until the end of a write. In a full FIFO, the writing side must wait for the reading side to create an "empty" location. Incomplete read and write cycles can not only invalidate data, but can cause the pointers to be out of synchronization, requiring a master reset to renew data transfer.

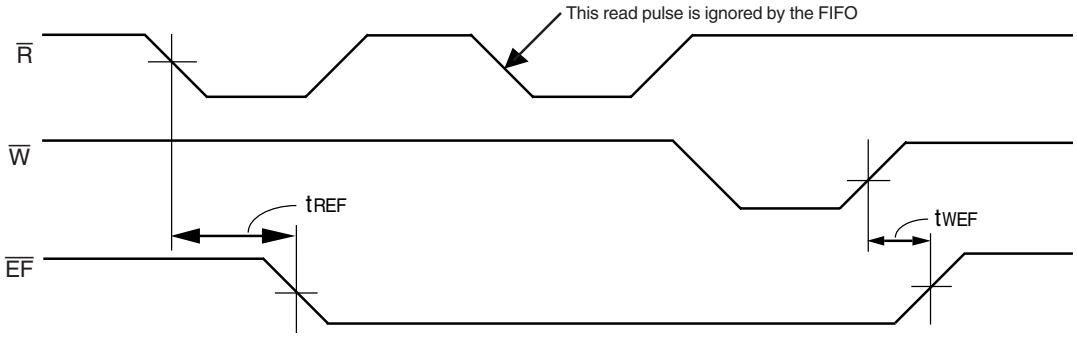
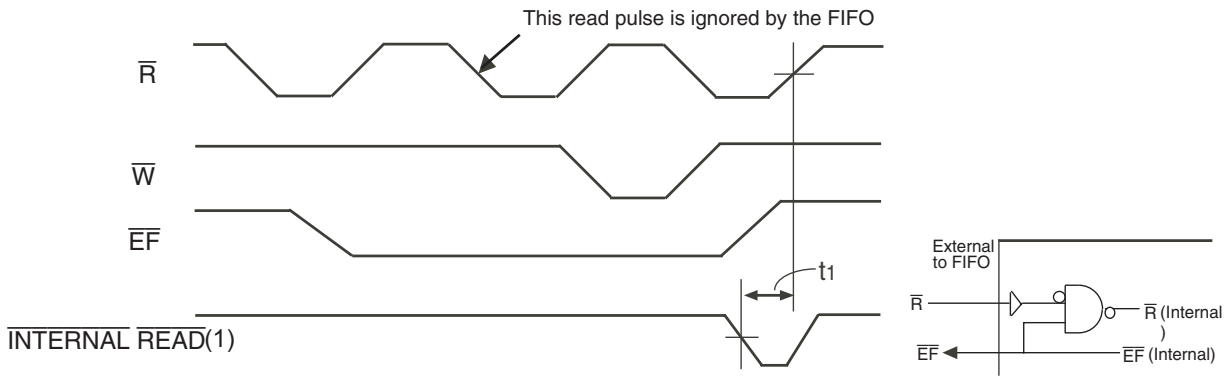


Figure 1. Empty Flag from Last Read to First Write



NOTES:

1. Pulse within the FIFO used to clock the write pointer and the Empty and Full Flag counters.
2. If $t_1 < t_{RPW}$ (minimum read pulse width low), then the read pointer, Empty Flag and Full Flag counters may be out of sync. See Figure 15 of IDT7201/7202LA data sheet.

Figure 2. Violation of tRPW During Boundary Conditions

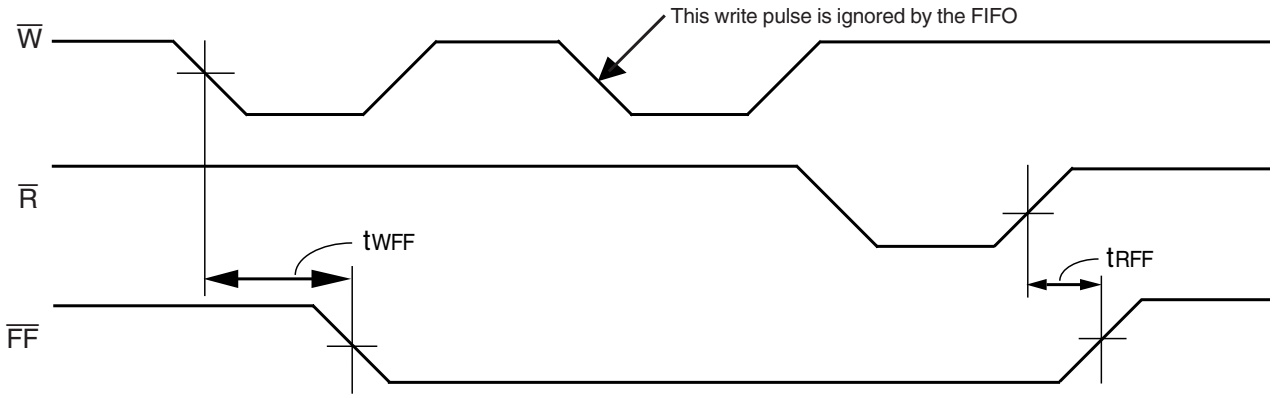
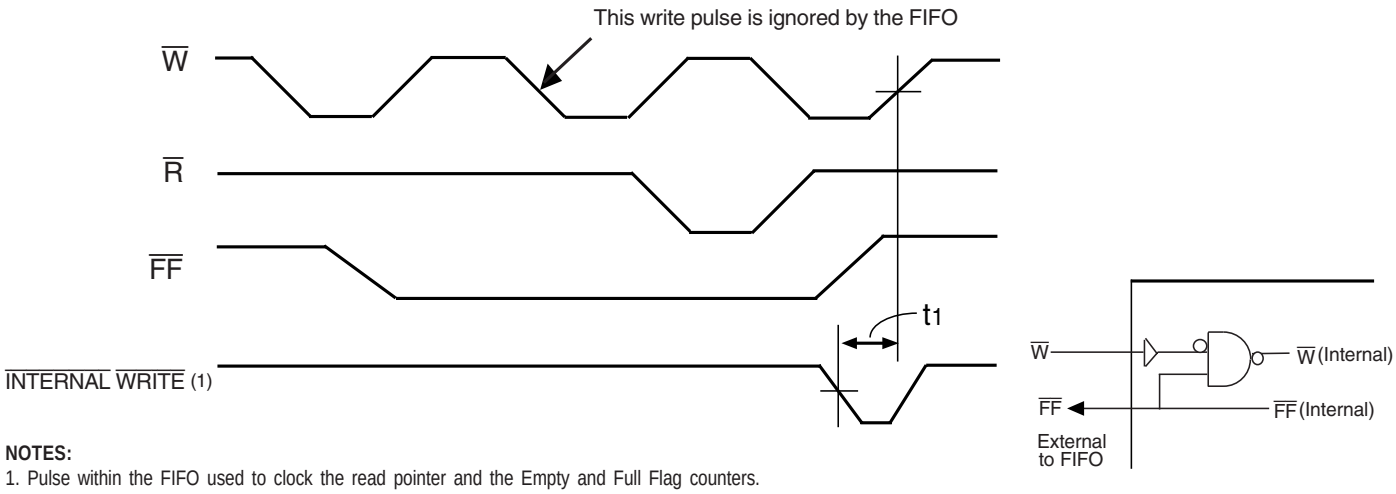


Figure 3. Full Flag from Last Write to First Read



NOTES:

1. Pulse within the FIFO used to clock the read pointer and the Empty and Full Flag counters.
2. If t_1 < t_{WPW} (minimum write pulse width low), then the write pointer, Empty Flag and Full Flag counters may be out of sync. See Figure 16 of IDT7201/7202LA data sheet.

Figure 4. Violation of t_{WPW} During Boundary Conditions

CORPORATE HEADQUARTERS
6024 Silver Creek Valley Road
San Jose, CA 95138

for SALES:
800-345-7015 or 408-284-8200
fax: 408-284-2775
www.idt.com

for Tech Support:
408-360-1753
email: FIFOhelp@idt.com