

Impact of Undershoot on Bus Switches and Corresponding Solutions in PCI Hot-Plug Applications

By Kay Annamalai

This application brief explains the need for undershoot protection for bus switches where their use is in busses with heavy ringing, such as PCI busses.

The block diagram of a simple n-channel FET switch is shown in Figure 1. It consists of a source connected to drain through a channel. The channel is enabled by means of HIGH level at the gate. Control is provided by an active LOW \overline{EN} signal. When the \overline{EN} signal is HIGH, the gate is LOW and the switch is disabled. When \overline{EN} is LOW, the gate is HIGH and the switch conducts. When the switch is disabled, and the input has an undershoot signal greater than the switch threshold voltage (0.7V) for conduction, the switch becomes enabled, since there is a forward bias between the gate and the source. This will cause an unwanted signal to go through the drain. For the PCI bus, the A side of the switch is connected to the PCI bus and the B side to the PCI slot. As a result of undershoots on the PCI bus, the signals on the A side gets connected to the B side, resulting in improper operation to the slot.

To alleviate false operation to the PCI slot, a bus switch with undershoot protection is required. This is shown in Figure 2. In this circuit, the \overline{EN} signal goes through undershoot protection circuitry before connecting to the gate. During undershoot conditions, the channel is prevented from conducting. Also, the substrate is negatively biased using a charge pump. By reverse biasing the substrate, there is no conducting path from substrate to source during undershoot, thus preventing any latchup conditions.

This protection circuitry has been employed in all of the following Pericom devices:

- PI5C6801C – 10 bit Bus switch
- PI5C16215C – 20 bit Bus switch
- PI5C32160C – 32:16 Mux/Demux Bus switch

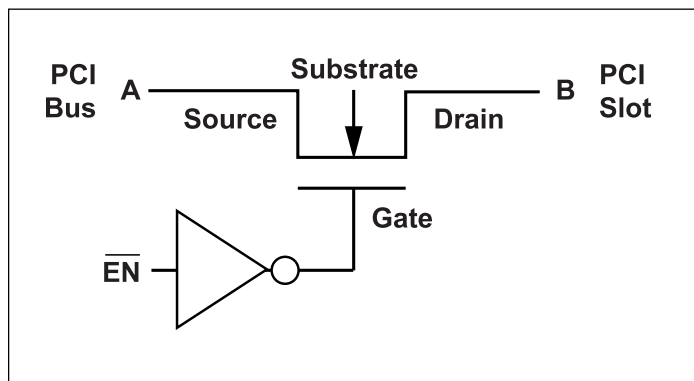


Figure 1. Regular n-Channel FET Switch.

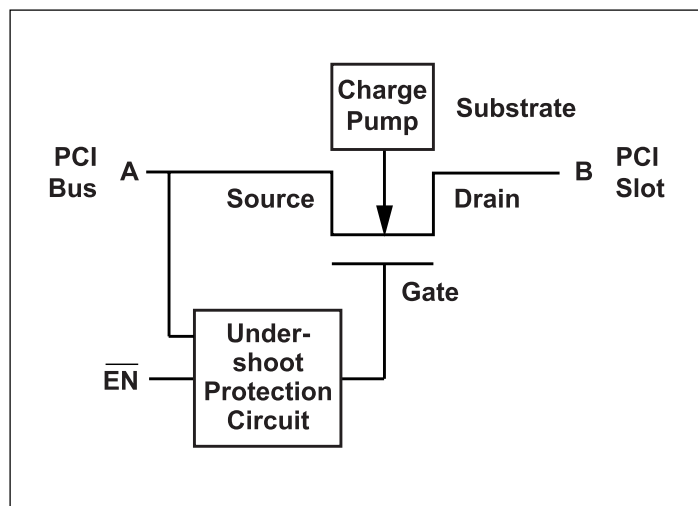


Figure 2. Undershoot Protected n-Channel FET Switch