# Keeping your Switches Alive with Deadtime

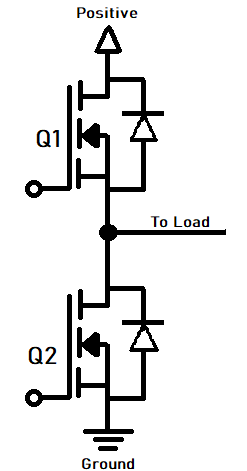
By

Klint Mahne

# Thumbs & Switches

Imagine you are using a hammer to strike nails into boards, and while you are holding the nails you graze your thumb ever so slightly. You may feel it, but because it’s just a graze it doesn’t seem like that big of a deal. After 50 times of this happening, you may notice some throbbing. After 500 the continuous strikes in the same spot add up, and your thumb is quite sore. Now imagine the 500 times happening in half a second – a 1Hz speed. That is what is happening to your switch at a 1hz switching speed, except instead of a hammer making contact it instead a massive current surging through and instead of your thumb, it’s your electronic device taking the beating. 1Hz is also quite slow for many switches, so now multiply the frequency by a few tens, or even hundreds, of thousands. Ouch.   
  
So, what exactly is happening when the ‘hammer strikes’, per say? To understand this, it is important to understand the configuration of the switches. For ease of explanation and understanding, I am going to use MOSFETs as the switch going forward, but know that it could just as easily be BJTs, IGBTs, SCRs or GTOs. They do have differences, however for the high-level explanation those difference won’t matter. In addition to this, we should also assume that the switches are being operated in a manner that maximizes their fully on or fully off time in order avoid the linear range in which the most power is dissipated as heat. The final assumption is that we are using PWM to control the output voltage, so if you’re not familiar with PWM it may be beneficial to familiarize yourself with it, however it is not crucial in understanding the concept of shoot-though.

# The Cycles

When using the MOSFETs to control the output of voltage regulation/conversion, you will typically see them in a “half-bridge” formation (also known as bridge-arms). Converters will normally have two or three of these half-bridge formations, and each one of these is individually at risk of shoot-through. For the purpose of this tutorial’s graphical representation, we will just use a single half-bridge. Below is the graphical representation of a bridge arm:

Q1 is tied to the positive rail, and Q2 is tied to the negative rail. The load is connected to the source of Q1 and the drain of Q2. Let’s take a look at the different cycles providing power to the load.

**POSITIVE HALF**

For the positive half-cycle, Q1 conducts from the positive rail and the current goes out into the load. Q2 will be off during this time .

**NEGATIVE HALF**

For the negative half-cycle, Q2 conducts as the load supplies the current and it flows into the ground.

Diagram, schematic

Description automatically generated

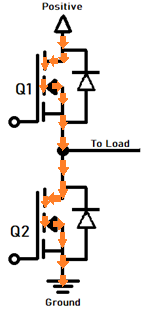
Diagram, schematic

Description automatically generated

Now, if you had put both of these bridge arms together (positive half connected to the positive terminal of the load and the negative half connected to the negative terminal of the load), you have a full bridge. Note, though, that the positive bridge arm still requires its Q2 to be off, and the negative side still requires its Q1 to be off. Before we continue, there is one more piece to tie this together - when a MOSFET is on it will have very small amounts of resistance. How does that come into play? If a MOSFET is on and its resistance is lower than the load, then current will want to flow through it instead of the load. These are the conditions that lead to shoot-through.

**SHOOT-THROUGH**

As you can see below, the current travels from the Q1 and bypasses the load to go directly into the Q2 of the same bridge arm. This short is essentially the ‘hammer striking the thumb’ and the condition that we want to avoid (unless, of course, your objective is to destroy your MOSFETs).



# Preventing Shoot-Through

Now we know what shoot-through is, what it looks like, and why it is bad, we need to know what we can do to prevent it. It’s simple – make sure that only one MOSFET in a single bridge arm is active a time! We accomplish this by implementing deadtime. Deadtime is about what it sounds like – a small period of time where neither Q1 nor Q2 MOSFETs are on. There are multiple ways to achieve deadtime in your circuit. The easiest way to do this is integrating a MOSFET driver that includes ‘not gates’ – hardware that prevents one MOSFET from turning on until the other is fully off. The deadtime rates may differ between drivers and however your circuit is applied, so be sure to take that into account if looking for a driver. In some cases, the microchip using logic to drive the gate (either through a dedicated driver or through a direct connection) may have hardware and software implementations of deadtime. For instance, the STM32Cube HAL allows you to configure the deadtime between pulses when using PWM. You can also use an resistor, capacitor, and diode to delay the turn on time of a MOSFET, giving the other MOSFET time to turn off.