Course A Power Electronics

Simulation

The problem often arises that electronic assemblies such as microcontrollers, sensors, small actuators or motors have to be supplied with 5V from a higher-level voltage source, e.g. 20V. Since linear regulators, which are nothing more than variable series resistors, generate excessive losses, buck converters are often used. With the use of modern semiconductors, efficiencies of > 98% can be achieved.
For our first experiment, we design a buck converter with the following properties:

Input voltage range: 15V ... 20V
Regulated output voltage: 5V
Rated output power: 25W

Ideal World

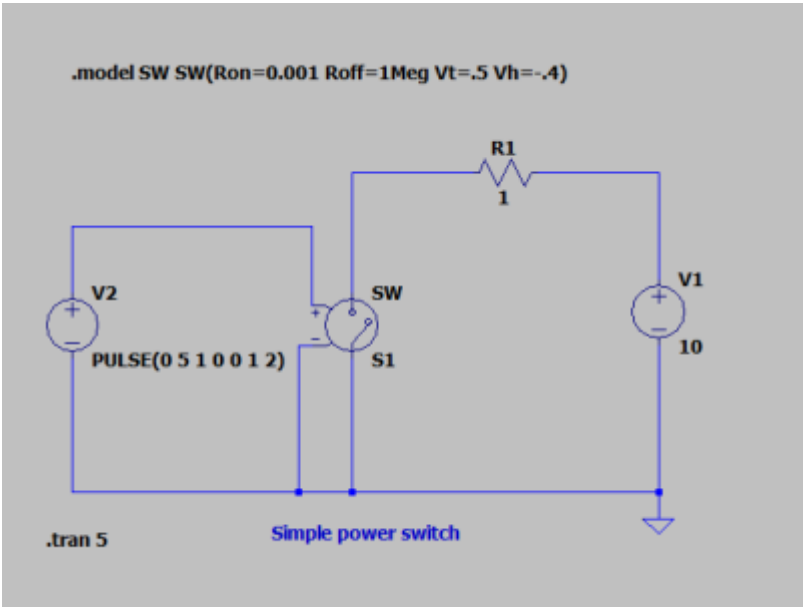
Unfortunately, a simulation is often very similar to installing real hardware: soldering everything together and applying voltage in the hope that it will work is utopian. Or in a simulation: parameterize all components, connect them and press the run button usually doesn't work 
This usually doesn't work even if we only use ideal components. We have to gradually start approaching the matter and build up the simulation step by step. And each step must confirm the results we expected at the beginning. If this is not the case, the simulation should be reduced or simplified again until the result meets our expectations in order to achieve a trustworthy result. Don't let a simulation lie to you. It is just as important as the simulation itself to check whether the results are plausible. If, for example, currents > 100A or voltages in the MS range occur in the simulation, you should not trust the result under any circumstances.

Task 1a) Create the circuit diagram of a buck converter consisting of a switching element,

diode, coil, capacitor and load. Determine the load resistance so that the required power is set at the output. Use LTSpice to create a simulation of a buck converter with the boundary conditions described above, consisting of an ideal switch, an ideal diode, coil and output capacitor.
Task 1b) Create a table and calculate the expected voltage and current values for each component. Sketch the voltage and current curves.
Task 1c) Use LTSpice to create a simulation of a buck converter with the boundary conditions described above, consisting of an ideal switch, an ideal diode, coil and output capacitor.
Task 1d) Compare the simulation result with the expected results. What do you notice? What explains the deviations?
Task 1e) You may have noticed that although we asked LTSpice to use an ideal diode, it was not chosen as ideal. Use the simulation to calculate the efficiency of your buck converter.

For the simulation please use the following parameters:
Switching frequency: Last two digits of your Matrikelnummer in kHz
Inductor: Last three digits of your Matrikelnummer in µH (if the value is less than 40, it is doubled until it is greater than 40)
Capacitor: You can either use a 330µF or 4700µF capacitor

Note: Within LTSpice you have to configure the switch parameters first:

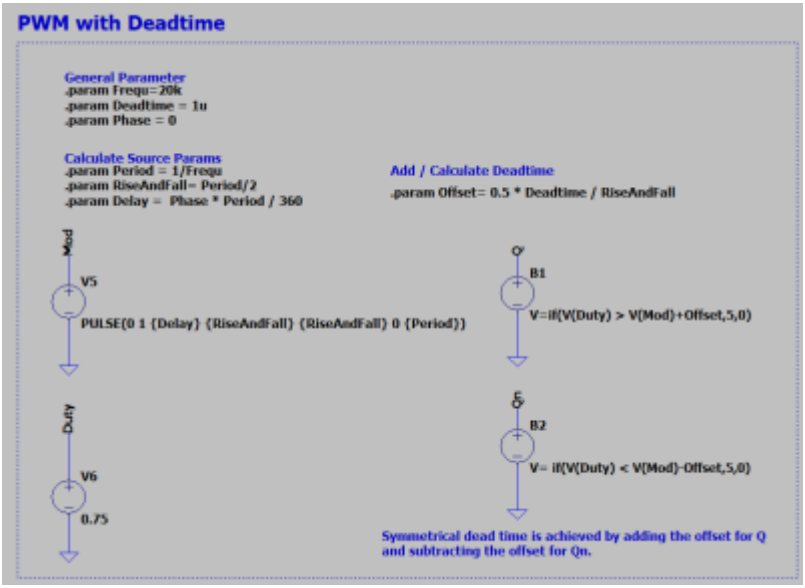


Diodes always incur power loss, primarily caused by the voltage drop across the device. Thus, the freewheeling diode is often replaced by a second power switch, leading to the commonly used half-bridge topology. This is a clever solution, especially when using MOSFETs as switches. MOSFETs always come with an intrinsic diode, known as the body diode. Therefore, even if the MOSFET is not activated, the body diode can serve as the freewheeling diode. The body diode, of course, exhibits the same poor power loss characteristics as the previously used diode. Therefore, we activate the MOSFET to allow the current to flow through the MOSFET instead of the diode.The rule to avoid short circuits is as follows: whenever the high-side MOSFET is turned off, the low-side MOSFET is turned on. However, this introduces another problem. Since the switching speed of the devices is not infinite, a certain amount of delay is required before turning on the low-side MOSFET after the high-side MOSFET is turned off, and vice versa. This delay is referred to as dead time.

Task 2a) Replace the diode in your simulation with an additional ideal switch and run the simulation again to determine the required power and voltage values.
Task 2a) Once again, compare the simulation results with your expected values. Ensure that

your simulation produces reasonable results, as values in the range of kA or MV are outside the expected scope!

Note: Dead time can be generated by shifting the comparator values with a negative or positive offset. In the following example, Qn is the inverse PWM signal of Q, with variable dead time.



Real Components

Adding Mosfets

LTSpice, like any network simulation program, is capable of accounting for the real behavior of MOSFETs. However, as we know, the parasitic elements of almost all semiconductors are strongly dependent on temperature. For MOSFETs, parasitics such as the parasitic capacitance are even dependent on the current voltage value. Nevertheless, having a simple, realistic behavioral model is better than having none at all. The models can normally be downloaded from the manufacturer's website. Here

irf3205s.zip

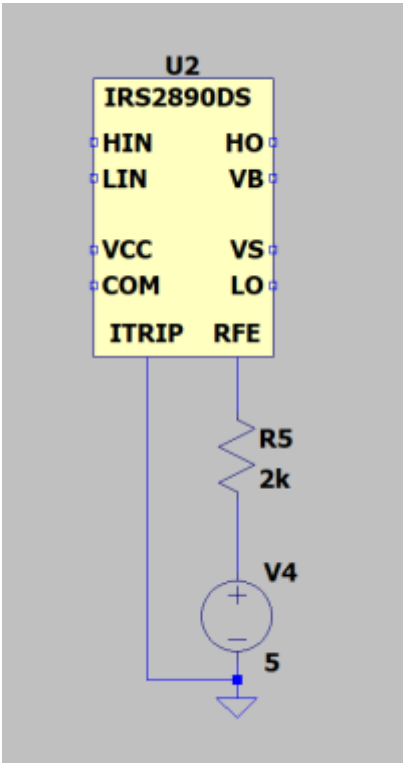
you can find the model for your IRF320 MOSFET.
Datasheet:

irf320.pdf

Task 3a) Replace the ideal switches with the MOSFET model. Be sure to set the correct gate-source voltages.
Task 3b) Determine the required dead time for the MOSFETs to ensure safe operation. Observe and document what happens if the dead time is too short. Take a screenshot for your documentation.
Task 3c) Verify if the values match the timing specifications provided in the datasheet. Determine the appropriate dead time required to prevent unwanted short circuits.
Task 3d) What is the maximum switching frequency the MOSFETs can operate at? What happens if we further increase the switching frequency?

Adding Gate Driver

In a simulation, it is very easy to generate an isolated voltage source that is independent of the common ground. In reality, generating isolated sources requires significant effort, as transformers are typically needed to decouple the grounding. The problem arises with the gate supply of the high-side MOSFET, as the drain potential continuously switches between the reference ground and the input voltage. A commonly used solution is the use of a so-called bootstrap circuit. Put simply, the circuit charges a capacitor which is then connected to the mosfet independently of the ground potential. Integrated circuits such as the IRS2890DS driver we use perform this task for us and can control one half-bridge at a time.



Gate drivers are often equipped with additional features. Our driver can detect overcurrents and if necessary, will turn off all MOSFETs for safety. Since overcurrents in a simulation tool will not damage your PC or notebook, you can ignore this feature in the simulation. Pull up the RFE output and connect ITRIP to ground.

Datasheet:

infineon-irs2890ds-ds-v01_00-en.pdf

Task 4a) Add the driver circuit to your simulation and check the functionality. The general behavior should be the same as bevore!

As a rule of thumb, the bootstrap capacitor is selected in the range of the input capacitance Ciss of the MOSFET to be driven. However, many developers start with a 100nF capacitor and check the function in the design. Gate series resistors are used either to reduce the switching speed of the mosfets or to dampen

unwanted oscillations. A value between 1 and 10 ohms is often used as the starting value.

Estimat Parasitics

DIY Board

Additional Components