

Femtosense SPU-001 EVK2

Application Note 001

SPI SCK Logic Gating

The SPU-001 EVK2 uses standard SPI to communicate with both the SPU-001 chip and an on-board flash chip, where each chip has an individual CS signal to select it. When using EVK2, it is important that the SCK clock signal of the SPI bus generated by the host MCU is free of irregular pulses or noise that crosses the logic threshold, *even when the CS signals are not asserted*. Regardless of the state of the CS signals, the only behavior that EVK2 permits on the SCK signal is pulses in multiples of 8.

This becomes a problem if another chip sharing the SPI bus communicates with the host MCU using a protocol where bits are sent in non-multiples of 8, or if the SCK signal is not controlled when chips are not selected. An example of the latter is given below. The three red arrows show where the host allows the SCK signal to be uncontrolled while the CS signal is not asserted (logic high). This behavior causes the SPU to lose synchronization in a way that is difficult to recover from without a reset.



Figure 1. SPI bus traces generated by an Analog Devices MAX78000 host MCU and EVK2. The three red arrows highlight irregular levels in the SCK signal that disrupt the SPU chip on EVK2.

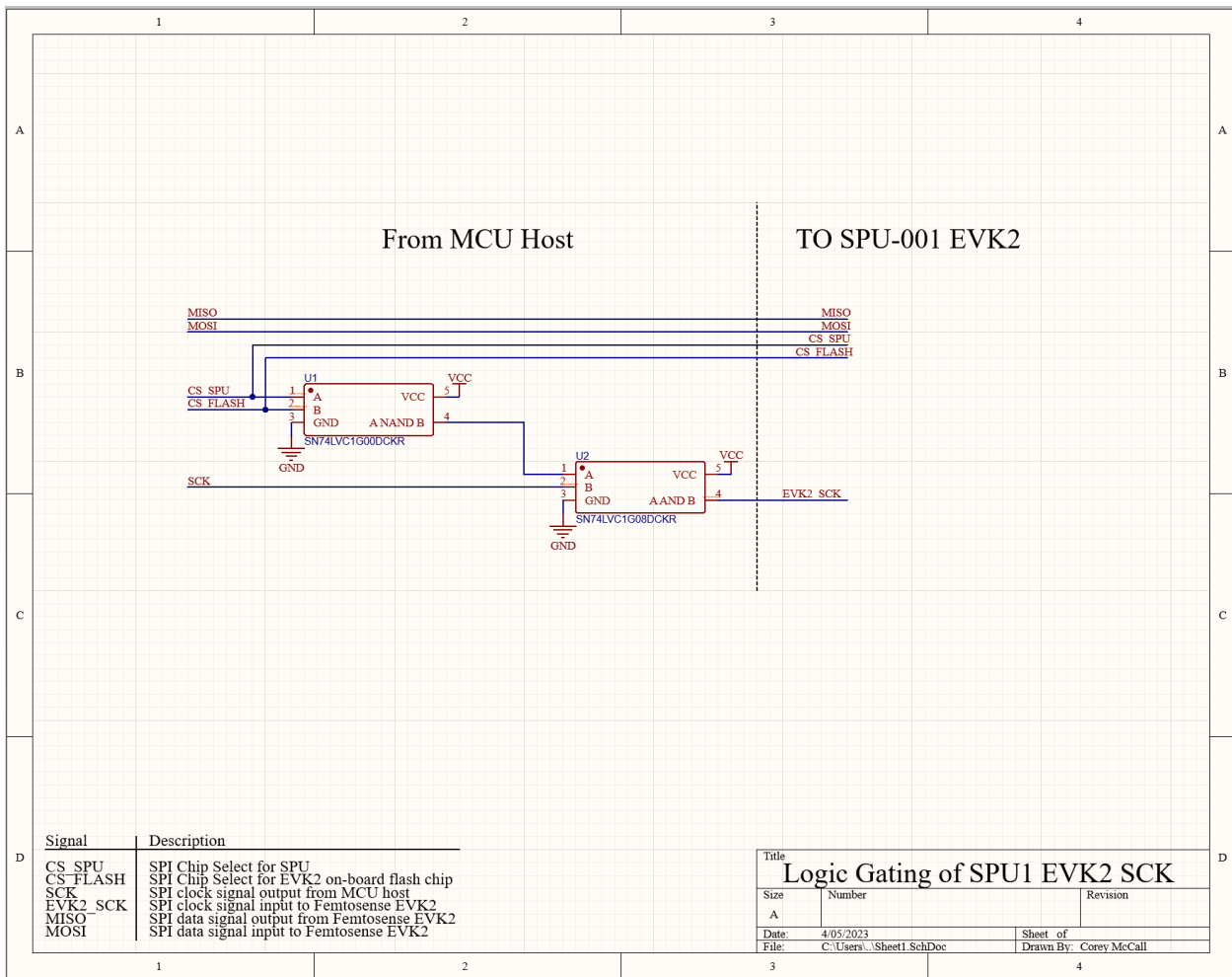
FemtoseNSE SPU-001 EVK2

Application Note 001

If this behavior is expected, one solution is to intercept the SCK signal between the host MCU and EVK2. A logic equation can be applied to the intercepted signal such that SCK is set to logic low unless one of the EVK2 CS signals are asserted:

$$EVK2_SCK = \overline{(CS_SPU \bullet CS_FLASH)} \bullet SCK$$

This can be implemented with discrete logic gates such as the TI SN74LVC1G00DCKR (NAND) and TI SN74LVC1G08DCKR (AND) as shown below. Note that only the SPI signals are shown.



Future versions of the FemtoseNSE SPU chip and EVK will not be susceptible to this issue.