

EEC172 Prelab 2

3/8/2024

Name:

SID:

Difficulty
Rating

Question 1.

(Short Answer) What are the 2 signal lines in I2C and the 4 signal lines in SPI called?

Difficulty
Rating

Question 2.

(Free Response) SPI and I2C busses can support multiple devices on their clock and data lines. How do the controller (main/master) devices initiate communications with a given peripheral (subnode/slave) device?

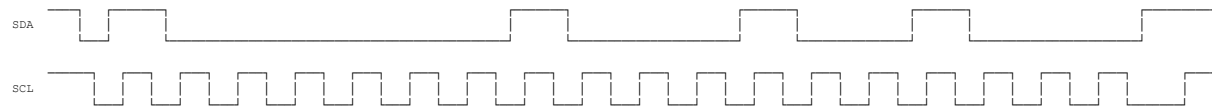
Difficulty
Rating

Question 3.

(True/False) In both the SPI and I2C protocols, the controller device generates the clock signal and is responsible for generating enough cycles to transmit its own data and receive all the data that the peripheral device generates.

Question 4.

(Free Response) Determine the 7-bit address and 8-bit integer that is being transmitted in the following I2C dataframe signal. Is the data being written to or read from the peripheral?

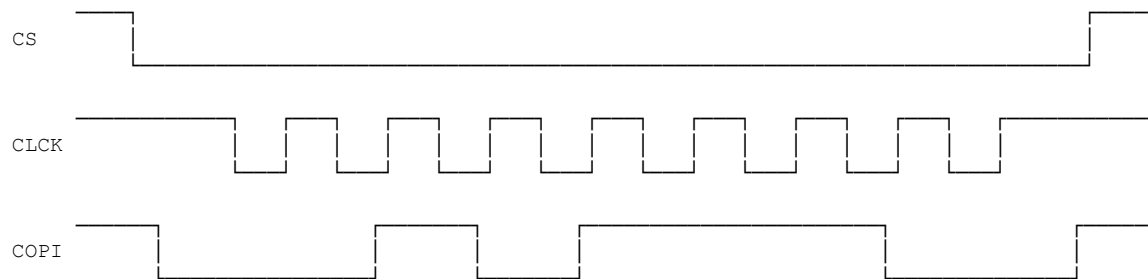


Difficulty

Rating

Question 5.

(Free Response) You see the following digital waveform on the SPI bus. You are expecting your micro-controller to read 0x2E from the peripheral, but instead it is seeing 0x5C. What do you think is causing this bug and why? What could we do to fix the problem? Assume that the hardware waits enough time for the signal to stabilize when sampling (reads the signal value after the clock edge).



Difficulty

Rating

Question 6.

(Free Response) We are designing a system that has many peripheral devices that must transfer a relatively large amount of data at a high speed. Because of this, we would like to use SPI; however, we are limited by the number of GPIO pins on our microcontroller. We have 3 free GPIO ports, but have 4 peripheral devices to communicate with. We don't have enough GPIO pins to assign one to each peripheral, but we also don't want to use a daisy chain configuration, since that requires more clock signals to receive data. So we still need an independent CS configuration where the clock, COPI, and CPO lines are still shared between all devices.

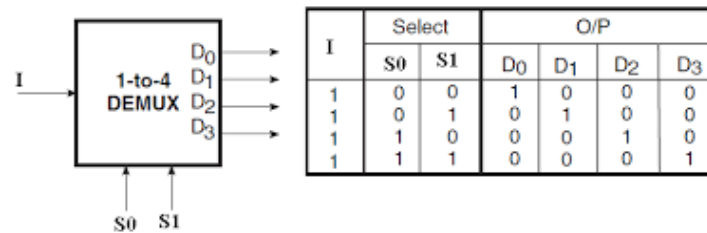


Figure 1: Demux Symbol and Logic Table

Fortunately, we can solve our problem with a 1-to-4 demultiplexer. A 1-to-4 demultiplexer (demux) takes 1 input signal and 2 selection lines, and based on the values of the selection lines, connects the input signal to one of 4 outputs. How can we use the 1-to-4 demux to enable communication with all 4 peripheral devices? Explain with a connection diagram.