TECH 3812
Lab #5
Intro to SPI

Ver 1.0

Background:

For the experiment we will be using an Arduino programmed to be a master (controller) that will only be sending (via MOSI) simple ascii characters to the Rigol scope for analysis.

Jumpers (will be used to modify the settings of the SPI interface so you can compare and contrast the different modes and speeds. NOTE: The Arduino MUST be reset for any changes in jumpers to be recognized!

Recall there are a total of 4 modes for a typical SPI interface as shown below:

CPOL	CPHA	Data Read and Change Time	SPI Mode
0	0	Read on rising edge, changed on a falling edge	0
0	1	Read on falling edge, changed on a rising edge	1
1	0	Read on falling edge, changed on a rising edge	2
1	1	Read on rising edge, changed on a falling edge	3

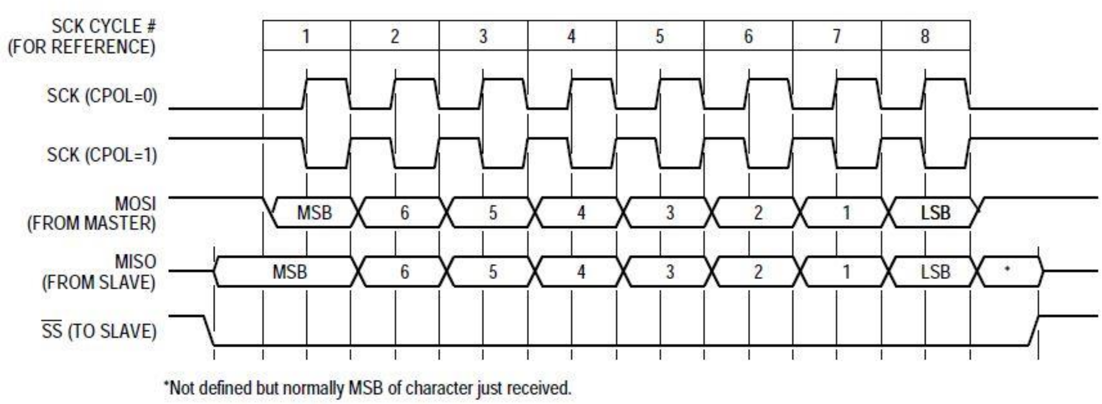

Figure 8-1. CPHA Equals Zero SPI Transfer Format

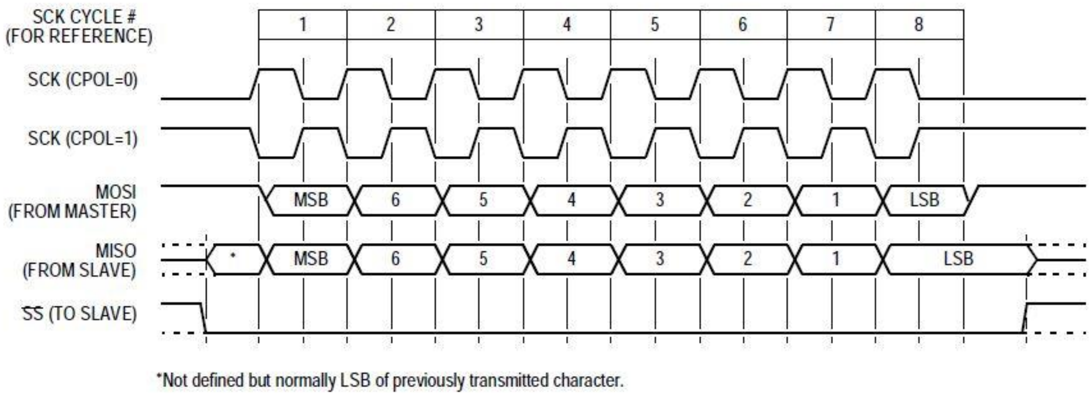

Figure 8-2. CPHA Equals One SPI Transfer Format

Procedure:

1. Take the pre-programmed Arduino and connect it to the Rigol Scope in the following manner:

Arduino Pin and Function	Rigol Scope Digital Input
13 (SCK)	0
11 (MOSI)	1
10 (SS)	4
GND	GND
GND*	

   * Connect the Arduino ground to a small breadboard. You will then use this ground to connect to other pins on the Arduino as instructed latter on in the instructions.

2. Load the setup file for the Rigol scope (provided on the USB thumb drive).

   Add info on the setup and the various decode options on the Rigol scope

3. Power up the Arduino board using a USB cable connected to a nearby PC. **DO NOT** power the Arduino from the Rigol scope.

4. Adjust the Rigol scope so you can see at least two pulses on the SS line. A three byte message is being transmitted, starting at the SS going low and ending at SS going high. Capture the image (making sure you have invert turned on when you do the capture). Print out and analyze this waveform and prove that the data shown by the protocol decoder is correct. Draw lines at each correct edge of the SCK and show the data being transmitted by the MOSI line (1 or 0). Once you have the binary/hex values, look up the ascii characters for each value and show that on the capture. Please mark this image "MODE 3"

5. Now add a jumper from GND to pin A2 (Analog in) and RESET the ARDUINO. You should notice a change in the OSCOPE output. You should also note that the protocol decoder is havening issues with decoding this signal, this is because we have gone from a Mode 3 with a rising edge clock (SCK) to mode 2 with a falling edge SCK. To correct this, press the DECODE1 button, select the SCK channel, and select falling edge for that channel. This should fix the issue.

   This is one of the issues discussed in class with SPI. If the modes do not match between the master (the Arduino) and the slave (Rigol scope), communications will not be possible. By changing to the falling edge, you are changing the mode on the Rigol scope to match the master's settings.

6. Once again, Adjust the Rigol scope so you can see at least two pulses on the SS line. A three byte message is being transmitted, starting at the SS going low and ending at SS going high. Capture the image (making sure you have invert turned on when you do the capture). Print out and analyze this waveform and prove that the data shown by the protocol decoder is correct. Draw lines at each correct edge of the SCK and show the data being transmitted by the MOSI line (1 or 0). Once you have the binary/hex value, look up the ascii character for that and show that on the capture. Please mark this image "MODE 2"

7. Move the jumper from A2 and place it in A3 and RESET THE ARDUINO. Once again you should notice a change in the OSCOPE output, but this time, since the mode has changed from Mode 2 to Mode 1, you should not need to adjust the Rigol scope SCK edge, since both are Falling edge.

   Since the change from Mode 2 to Mode 1 only has a PHASE change between the clock and the data, no adjustments are needed to obtain the correct data from the Rigol scope SPI Decoder.

8. Once again, Adjust the Rigol scope so you can see at least two pulses on the SS line. A three byte message is being transmitted, starting at the SS going low and ending at SS going high. Capture the image (making sure you have invert turned on when you do the capture). Print out and analyze this waveform and prove that the data shown by the protocol decoder is correct. Draw lines at each correct edge of the SCK and show the data being transmitted by the MOSI line (1 or 0). Once you have the binary/hex value, look up the ascii character for that and show that on the capture. Please mark this image "MODE 1"

9. Now leave the jumper from GND to A3 and add a jumper from GND to pin A2 (Analog in) and RESET the ARDUINO. You should notice a change in the OSCOPE output. You should also note that the protocol decoder is havening issues with decoding this signal as before. Once again , this is because we have gone from a Mode 2 with a falling edge SCK to Mode 1 with a rising edge SCK. To correct this, press the DECODE1 button, select the SCK channel, and select rising edge for that channel. This should fix the issue.

10. Once again, Adjust the Rigol scope so you can see at least two pulses on the SS line. A three byte message is being transmitted, starting at the SS going low and ending at SS going high. Capture the image (making sure you have invert turned on when you do the capture). Print out and analyze this waveform and prove that the data shown by the protocol decoder is correct. Draw lines at each correct edge of the SCK and show the data being transmitted by the MOSI line (1 or

0). Once you have the binary/hex value, look up the ascii character for that and show that on the capture. Please mark this image "MODE 0"

11. Now take the capture from step 4, but this time use the WRONG SCK edge and mark the first byte of date. Mark this image "MODE 3 – Incorrect Decode". Looking at the waveform, answer the following question (on the bottom of the printout):

    Q: What problem do you see trying to decode the one's and zeros of this SPI Communications. Relate your answer to our discussion of the issues with setting up SPI.

12. Looking at the captures from steps 4, 6, 8 and 10. Answer the following Question (please put your answer on the back of the MODE 3 output.

    Q. At the point SS Signal goes from a High to a Low, describe the VERY SUTTLE differences between the SCK and MOSI in relation to one another. How does this relate to the Modes?

> As we discussed in class, SPI also has the ability to communicate at different clock rates. The next part of the experiment is about the SPEED of the transmission and the Clock Rate.

13. Now disconnect the two ground wires on A3 and A2 and RESET THE ARDUINO. Change the time scale so that you see 2 full clock pulses. Using the Cursor Functions on the scope, measure the time for ONE COMPLETE CLOCK CYCLE.

    To do this: Press “Cursors” button to call up the cursors menu. Set “Mode” to “Manual” (you can leave “Display Mode” as x and “Source” as “LA” - Logic Analyzer). Select “Cursor A” and use the multifunction Knob to move the cursor to the start of a clock cycle. Select “Cursor B” and use the multifunction know to move the cursor to the end of the clock cycle. The box should show the cycle time as ΔX.

    Capture the image on the screen.

14. On the capture, calculate the frequency of the clock. Also calculate the time it takes to send the 24 bit message. Mark this page as "Speed 3"

15. Connect GND to A0 and RESET THE ARDUINO. Change the time scale so that you see 2 full clock pulses. Using the Cursor Functions on the scope, measure the time for ONE COMPLETE CLOCK CYCLE. Capture the image on the screen.

16. On the capture, calculate the frequency of the clock. Also calculate the time it takes to send the 24 bit message. Mark this page as "Speed 2"

17. Move the GND from A0 to A1 and RESET THE ARDUINO. Change the time scale so that you see 2 full clock pulses. Using the Cursor Functions on the scope, measure the time for ONE COMPLETE CLOCK CYCLE. Capture the image on the screen.

18. On the capture, calculate the frequency of the clock. Also calculate the time it takes to send the 24 bit message. Mark this page as "Speed 1"

19. Leaving GND connected to A1 and RESET THE ARDUINO, add a connection from GND to A0 as well. Change the time scale so that you see 2 full clock pulses. Using the Cursor Functions on the scope, measure the time for ONE COMPLETE CLOCK CYCLE. Capture the image on the screen.

20. On the capture, calculate the frequency of the clock. Also calculate the time it takes to send the 24 bit message. Mark this page as "Speed 0"

One Last Thing

21. Disconnect A0 and A1 and RESET THE ARDUINO. Adjust the time scale to zoom in on the SS line pulse at the start of the message. Using Cursors, measure the Off Time of the clock before SS line pulsed, the ON TIME at the SS pulse and the OFF TIME After the SS line pulse. Record these three times In a table under your answer for #12.

    Q. There should be a difference in these times. Come up with a plausible explanation why this is so.

Submit a paper copy with a suitable title page, the 9 captures and your answers to the questions. Due start of lab next week