Interfacing to the QF4A512 Digital SPI Port

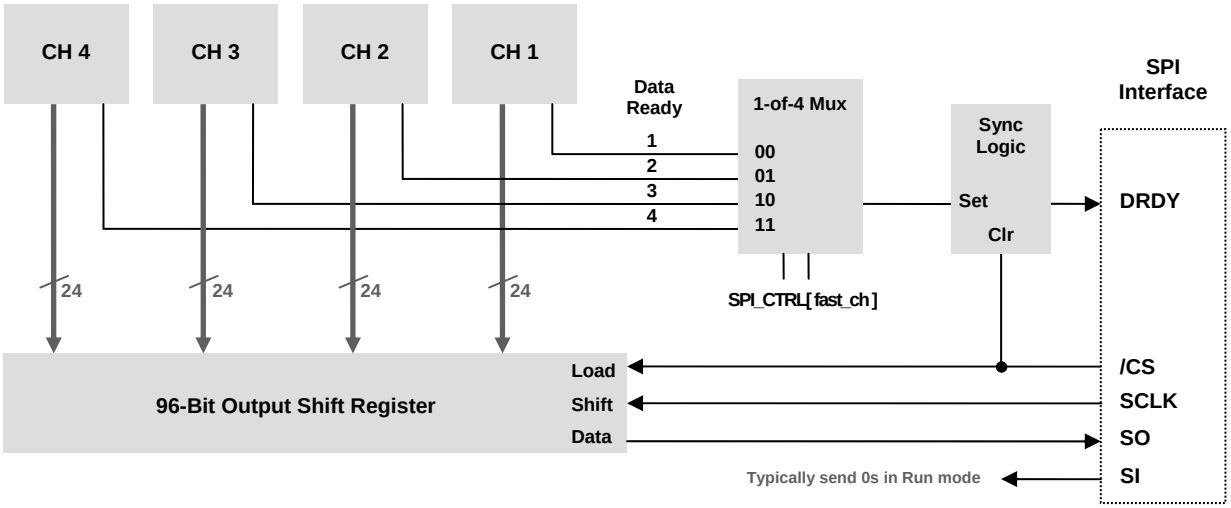
1) Introduction

The digital interface to the QF4A512 is a Serial Peripheral Interface (SPI) in the slave configuration. In Run mode, a continuous stream of filtered conversion results is available at the SPI port. Applications may use 1, 2, 3 or all 4 channels simultaneously.

2) Overview

Figure 1 shows the interface logic in the QF4A512 used to implement the SPI interface.

Figure 1 – Internal SPI Logic for Run Mode Operation (all 4 Channels Enabled)



The process of emitting frames out the SPI port is as follows.

Figure 2 – QF4A512 Frame Sequence during Run Mode Operation

1. The fastest enabled channel in the **ENABLE_2 (0Bh)** register finishes processing data and sets its Data Ready signal high
2. If the same channel is selected in the **SPI_CTRL[fast_ch] (15h)** register, DRDY will latch high, interrupting the host CPU
3. The host CPU detects DRDY high and activates /CS, which loads the channel data into the Output Shift Register and clears DRDY
4. The host serially clocks data out the SO pin
5. The host completes the cycle by deactivating /CS

Background Info

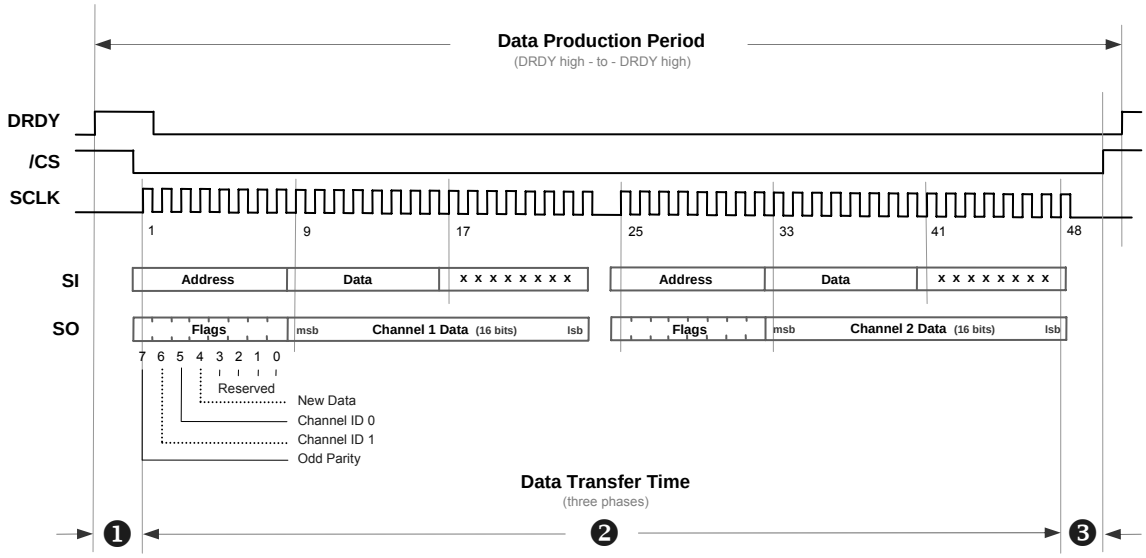
The SPI bus is not an official standard, but the implementation invented at Freescale (formerly Motorola) has become the *de facto* standard. See - www.freescale.com/files/microcontrollers/doc/ref_manual/S12SPIV4.pdf.

Debug Note

Be sure the design allows /CS to be active (low) for at least four SYS_CLK (not SCLK) cycles during each SPI bus cycle. Although this would rarely be an issue, it can take up to three SYS_CLK cycles for DRDY to be reset inside the QF4A512. If /CS is de-asserted too soon, DRDY might not be cleared in time for the next cycle.

The format of the data from the Output Shift Register is as follows.

Figure 3 – QF4A512 SPI Bus Sequence during Run Mode with Two-Channels Active



Notes -

1. The Host SPI port will only emit SCLK when actually transferring data.
2. Data is valid on the rising edge of SCLK.
3. Parity flag available if **SPI_CTRL[sif_par]** is set.
4. Over/under range flags available if **SPI_CTRL[adc_stat]** is set.

The Data Production Period is the time it takes the QF4A512 to produce consecutive frames of sampled, processed data. Data Transfer Time is the time it takes to remove a frame of data from the QF4A512. It consists of three phases,

❶ The time from DRDY high to the first rising SCLK edge. This phase will vary depending on how quickly the host CPU can detect the DRDY-high transition and start a SPI cycle.

❷ The time to clock the data out the SPI port. This time is –

Equation 1 – Data Transfer Phase 2 in Terms of SCLK Period

$$❷ = \text{Number of Active Channels} \times (\text{SCLK Period} \times 24\ \text{Bits/Channel}) + \text{SPI Cycle Gap})$$

❸ The time from the last rising SCLK to /CS high, ending the data transfer cycle. This phase will vary depending on how quickly the host CPU can detect the end of its SPI cycle and take /CS high. On some processors /CS is controlled by the SPI hardware, but on small microcontrollers /CS is often a general-purpose IO pin that is controlled programmatically.

Each channel generates 24-bits per sample, including 8 flag bits and 16 data bits. The size of the Output Shift Register is 24, 48, 72 or 96 bits, depending on whether 1, 2, 3 or all 4 channels are enabled, respectively. The frame sequence in **Figure 2** must occur at the rate of the fastest active channel. Other channels may not have new data to submit on every frame, so they submit the data from the prior frame with the New Data bit cleared.

Design Note

Be aware of what is sent to the QF4A512 in Run mode. A reduced command set is still active. For example, the PGA gain may be controlled on-the-fly in Run mode. Unless intentionally controlling the device, always send zeroes. The 0 command is benign (it writes to the Scratch register, GLBL_SW).

Calculating SCLK

The QF4A512 SPI interface is always in the slave configuration, with the host CPU providing SCLK. To avoid data loss, the Total Data Transfer Time (see **Figure 3**) must occur in less time than it takes the channel to produce consecutive samples. Expressing that concept as an equation yields;

Equation 2 – Maximum Time to Remove SPI Data without Loss

$$\text{Data Transfer Time} \leq \text{Data Production Period}$$

The Data Transfer Time is the sum of the three Phases shown in **Figure 3**.

Equation 3 – Restatement of Equation 2

$$❶ + ❷ + ❸ \leq \text{Data Production Period}$$

Replacing ❷ with the right side of **Equation 1** and solving for SCLK Period produces the following.

Equation 4 – Minimum SCLK Period to Remove SPI Data without Loss

$$\text{SCLK Period} \leq \frac{\text{Data Production Period} - ❶ - ❸}{\text{Number of Active Channels} \times 24}$$

SCLK Frequency, not *SCLK Period* is what is ultimately needed. Likewise, *Data Production Period* should be expressed as the *Data Production Rate*. Inverting *SCLK Period* and *Data Production Rate* converts them to the frequency/rate form. The equation changes to

Equation 5 – Minimum SCLK Frequency to Remove SPI Data without Loss

$$\text{SCLK Frequency} \geq \frac{\text{Number of Active Channels} \times 24}{(1 / \text{Data Production Rate}) - ❶ - ❸}$$

Notes -
1. If SCLK Frequency < SYS_CLK, be sure /CS will be low for at least four SYS_CLK cycles (per DRDY timing discussion above)

Note that the fastest channel Sampling Rate (f_s) is used, not the raw ADC sampling rate. Also note that the QuickFilter software provides a minimum SCLK calculation for each design.

Debug Note
DRDY is asserted whenever data is ready, regardless of the state of /CS. If a low-to-high transition is observed on DRDY while /CS is low, it is likely that the bus is not running fast enough to keep up with the data output rate.

3) Example

Now a multiple-channel example can be considered. In this example an FPGA is connected to the QF4A512 SPI interface. Channels 1, 2 and 4 are emitting processed samples at a rate of 10 Ksps, 25 Ksps and 50 Ksps, respectively. Channel 3 is not used. The FPGA is tasked with reading the samples without any data loss. What follows is a description of the FPGA structure, SPI initialization and operation sequences that will fulfill that task.

FPGA Structure

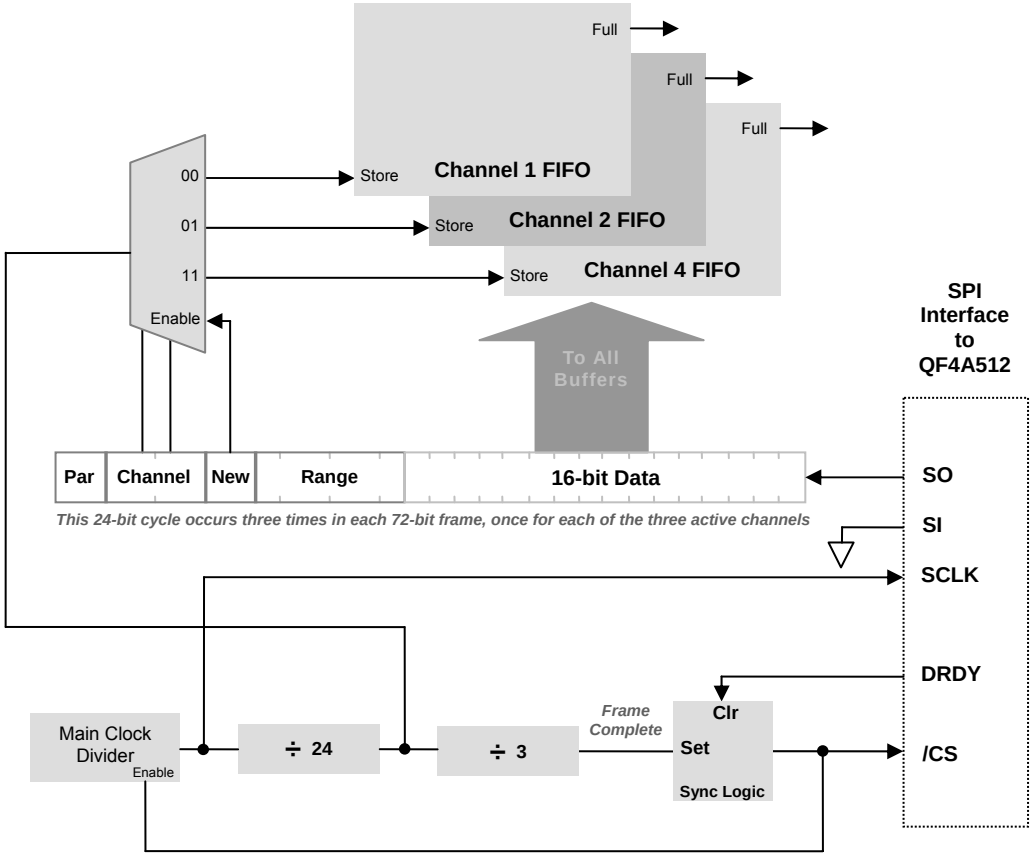
Figure 4 shows the design structure for this example. It is designed to interface to the QF4A512 SPI port, as shown in **Figure 1**. With three channels enabled, the QF4A512 will emit 72-bit data frames. The first 24-bits will contain the Channel 1 data, the next 24-bits will be Channel 2 data and the final 24-bits will be Channel 4 data.

Data from each channel is present in every frame, regardless of whether the channel has any new data to offer. For this example, with a frame rate of 50 KHz, every frame will contain new data for Channel 4, every other frame for Channel 2 and every fifth frame for Channel 1. The New Data bit in the Flags area indicates whether the data is, in fact, new.

Design Note
Note that the FPGA does not need to calculate which samples will have new data in a given frame. Observing the state of the New bit is all that is required.

The FPGA receives samples individually and captures them into a channel-appropriate FIFO. Storage only occurs if the New bit is set in the data. The FIFO generates an interrupt when full. Channel 1 data arrives first, then Channel 2 data and then Channel 4 data.

Figure 4 – FPGA Design Structure for Example 1



The logic of **Figure 4** is used to implement the following sequence in Run mode.

Figure 5 – Logic Sequence for Data Reception

1. Detect DRDY and synchronize it to the FPGA clock
2. Take /CS low and start SCLK, storing the state of SO on the rising edge. Send 0 on SI.
3. After 24 SCLK cycles, store data specified by the Channel bits if the corresponding New bit is set. Generate a Full interrupt if adding the data fills the FIFO.
4. Repeat step 3 two more times.
5. Stop SCLK, take CS high, clear divide-by logic

Clocking

The start of data transfer begins when the QF4A512 takes DRDY high. But the actual transfer time is dependent on SCLK, which is generated by the SPI master (e.g. the host CPU). The FPGA synchronizes the DRDY transition with its internal clock, then it activates /CS and generates 72 SCLK cycles to transfer data. Data must be stored after 24, 48 and 72 SCLK cycles, so a divide-by-24 functionality is provided. Finally, a divide-by-3 counter counts the number of 24-bit cycles to detect the end of the frame.

Once DRDY goes high, it takes the FPGA 1 µs to detect the transition, take /CS low and start the first SCLK edge. This represents phase ❶ in **Figure 3**. At the end of the transfer, it takes the FPGA another 1 µs to de-assert /CS after the last rising edge of SCLK. This represents phase ❸ in **Figure 3**. Using this data and the Data Production Rate of 50,000 samples/s in **Equation 5**, SCLK must be greater than or equal to 4 MHz for this example.

Initializing the QF4A512

This Section describes how to initialize the QF4A512 SPI-related features. Configuration of other QF4A512 features, such as clocks, sampling rates and FIR filters are described in other Application notes.

Figure 6 - Host CPU Initialization Sequence Description

1. Enter QF4A512 Configure mode and setup the SPI features

RUN_MODE[run_mode]	=	0	Enter Configure mode
SPI_CTRL[fast_ch]	=	3	Set Channel 4 as the source of DRDY
ENABLE_2[sif_ch_en1]	=	1	Enable Channels 1, 2 and 4
ENABLE_2[sif_ch_en2]	=	1	
ENABLE_2[sif_ch_en4]	=	1	
RUN_MODE[run_mode]	=	1	Enable Run mode

2. Synchronize with data stream by Enabling Run mode then throwing away the first sample. To do this, activate /CS (don't shift any data), wait for DRDY to clear, and then enable DRDY interrupts to start on the beginning of the next cycle.

Notes -
1. The Quickfilter PC software generates an appropriate initialization sequence like the one shown in 1. above.

4) Conclusion

This Application Note describes the Run mode operation of the SPI interface. For single-channel applications, refer to *QFAN0002-Single-Channel High-Speed Operation*. Also, examples of both interrupt-based and polling-based operation were presented.

Contact Information:

Quickfilter Technologies, Inc.
1024 S. Greenville Avenue, Suite 100
Allen, TX 75002-3324

General: info@quickfilter.net
Applications: apps@quickfilter.net
Sales: sales@quickfilter.net
Phone: 214-547-0460
Fax: 214-547-0481
Web: www.quickfiltertech.com