*Project: RMC75E FPGA TEST BENCH*

*Module:* SSITop*.vhd*

*Author: Satchel Hamilton*

*Company: Delta Motion*

*Date: 7/13/2023*

*Last updated: July 26, 2023*

Overview:  
  
the "SSITop" module represents an interface for communicating with linear and rotary transducers using a synchronous serial communication protocol known as SSI (Synchronous Serial Interface). It takes in data and system clock signals and provides an SSI clock signal and output data for external systems.

Here's how it works:

1. System Clocks: The module operates with two system clocks: "H1\_CLKWR" running at 60MHz and "SysClk" running at 30MHz.
2. Data and Control Signals: The module receives various control signals such as "Enable," "SlowEnable," "SynchedTick," "PositionRead," and "StatusRead," which control its behavior.
3. Data Transfer: It receives 32-bit data through the "intDATA" input signal, which is then processed and made available for external systems through the "ssiDataOut" output signal.
4. SSI Clock Generation: The module generates an internal SSI clock signal called "intSSI\_CLK" that controls the timing of data transfer. The SSI clock frequency is determined by the "ClockRate" signal, which is set externally.
5. Data Latching: It latches the received data into a holding register on the rising edge of "SynchedTick."
6. Shift Register: The module contains a shift register that is used to convert incoming serial data into parallel data for processing.
7. Wire Break Detection: It checks for wire break conditions by monitoring the data line before and after data transfer.
8. Valid Data Indicator: The "DataValid" signal is used to indicate whether valid data is available for external access.

Overall, the "SSITop" module acts as an interface between the system's clocks, data signals, and the external SSI communication protocol, ensuring proper data transfer and synchronization with linear and rotary transducers.

Initialization guide:

Here's an outline initialization procedure for the SSITop module:

1. Setting Parameters 1: ParamWrite1 should be set to send a 60 MHz activation pulse. This will enable the writing of parameters into the intDATA signal on a rising edge of the H1\_CLKWR clock signal. TransducerSelect is set to SSIBinaryAnalog, which equals 6 (110). DataLength is set to 8.

2. Setting Parameters 2: ParamWrite2 should be set to send a second 60 MHz activation pulse, which will be slightly offset from the first pulse (e.g. by 1 μs). This will enable the writing of the next set of parameters. DelayTerminalCount is set to 0, and HalfPeriod is set to 2.

3. Beginning SSI Read Cycle: The StartRead signal can be asserted right away because DelayTerminalCount is set to 0.

4. Reading Data: During the SSI read cycle, the SSI\_DATA signal is sampled at the falling edge of the internally generated SSI clock (intSSI\_CLK) if Shift signal is active. This process continues until the ShiftCounter matches DataLength, indicating the required number of bits that have been read.

5. SynchedTick Pulses: SynchedTick will need to send two 30 MHz pulses. The first pulse is sent at 2 μs and the second pulse at 10 μs. This process leads to the latching of Serial2ParallelData contents into SSIDataLatch, and Serial2ParallelData is cleared.

6. Ending the Read Cycle: The read cycle is terminated when ShiftCounter matches DataLength, and the internal data valid (intDataValid) status is cleared.

7. Data Output: After the read cycle, if SynchedTick signal is asserted, the contents of the Serial2ParallelData register are latched into SSIDataLatch and Serial2ParallelData is cleared.

Correct initialization of the system will be visible through the functioning of the SSITop module. Here are some indicators that can confirm the correct initialization of the system:

1. \*\*SSI\_DATA Stream\*\*: After setting all the parameters and initiating the read cycle, we should see meaningful data being read into `Serial2ParallelData` from `SSI\_DATA` at the falling edge of the internally generated SSI clock (`intSSI\_CLK`). The data should be in the format we expect based on the `TransducerSelect` and `Datalength` parameters.

2. \*\*SSI Clock\*\*: The SSI clock (`SSI\_CLK`) should generate nine negative pulses as mentioned in the notes. The frequency of the SSI clock is defined by `HalfPeriod` parameter and it should be as per the set configuration.

3. \*\*Data Output\*\*: The data out signal (`ssiDataOut`) should hold the read data when `PositionRead` is '1' and SSI Select (`SSISelect`) is '1'. This implies that the data from the SSI transducer is being correctly latched into `SSIDataLatch` and is ready to be output.

4. \*\*Status\*\*: When `StatusRead` and `SSISelect` are both '1', the `ssiDataOut` port should show the status information. The status information should reflect the configuration we have set (ClockRate, DataLength, TransducerSelect, etc) as well as the `NoXducer` and `DataValid` flags.

5. \*\*DataValid\*\*: The `DataValid` signal should be set to '1' after successful data read and remain so until the `PositionRead` signal falls. This indicates that the data read cycle was successful and that the read data is valid.

6. \*\*SynchedTick pulses\*\*: The `SynchedTick` should generate two pulses at 30 MHz as per the provided intervals. The first pulse should occur at 2 microseconds and the second pulse should occur at 10 microseconds. These pulses are essential for controlling the timing of various operations.

7. \*\*Wire Break Detection\*\*: The Wire Break detection functionality should function correctly. If the `SSI\_DATA` line is not high at the start of the read cycle or not low at the end of the read cycle, a wire break should be detected and the corresponding signals `DataLineHi` and `DataLineLo` should reflect this.