**Register Module:**

* The Acknowledgment bit going to AXI module is a one clock cycle **pulse** that can start when read enable is set to ‘1’.
* Whenever data changes in a register it sends a pulse to top entity signaling change.

**SPI Module:**

* There shall be an Ack in port for the register module. SPI module will only update its settings based on what is coming from the register module after it sees a pulse on the Ack port.
* SPI\_CLK and AXI\_CLK are assumed to be synchronized with each other during slave operation.
* In automatic mode slave select becomes all ones in between transfers and the spissr register contents is used to direct the destination of the next element transfer.
* In manual mode slave select register contents are put out directly to the slave select register contents. Slave select must still be set to all 1’s in between change of slave destinations but not in between transfers to the same destination. (User will know this must be done by software to prevent errors on SPI bus).

**FIFO:**

* Should accept read and write as agreed.
* Should have empty flag sent directly to master and slave (for TX)
* Should discard data if full (for RX)
* Depth is set at 16, width = C\_NUM\_TRANSFER\_BITS

**SPI MASTER / SLAVE:**

* (master and slave will receive the empty flag directly from the TX FIFO)
* Slave is responsible for sending zeros when TX FIFO is empty
* Master will be disable when TXFIFO is empty

Write cycle

-out data on the line, we =1

-we recognized, data accepted into fifo, we set back to 0

-we is recognized as zero, process repeats if new write must be initiated

Read cycle

-set RE=1

-RE recognized, pointer increments, RE set back to 0

-read data from the line, re recognized as 0, read cycle will restart next time a read is needed

Control Unit

-produces latched output to run master or slave from spissr, spicr and other registers

-registers sends contents to control unit any time their contents change. (using an ack pulse).