1. AXI Interface (Tag prefix: AXI)
   1. General functionality(Tag : AXI\_GEN\_)
      1. [AXI\_GEN\_01] The device should function only if provided either read or write address in range [C\_BASEADDR , C\_HIGHADDR].
   2. Address Decoder and Register Select(Chip select for registers) (Tag : AXI\_CS\_)  
       --Software Reset Register Chip Select generation (Write only)
      1. [AXI\_CS\_01] Srr\_cs signal shall be set when s\_axi\_awaddr is equal to 0x[C\_BASEADDR + 40] and s\_axi\_awvalid is set.  
           
         --SPI Control Register Chip Select generation (R/W)
      2. [AXI\_CS\_02] Spicr\_cs signal shall be set when s\_axi\_awaddr is equal to 0x[C\_BASEADDR + 60] and s\_axi\_awvalid is set.
      3. [AXI\_CS\_03] Spicr\_cs signal shall be set when s\_axi\_araddr is equal to 0x[C\_BASEADDR + 60] and s\_axi\_arvalid is set.  
           
         --SPI Status Register Chip Select generation (Read only)
      4. [AXI\_CS\_04] Spisr\_cs signal shall be set when s\_axi\_araddr is equal to 0x[C\_BASEADDR + 64] and s\_axi\_arvalid is set.  
           
         --SPI Data Transmit Register Chip Select generation (Write only)
      5. [AXI\_CS\_05] Spidtr\_cs signal shall be set when s\_axi\_awaddr is equal to 0x[C\_BASEADDR + 68] and s\_axi\_awvalid is set.  
           
         --SPI Data Receive Register Chip Select generation (Read only)
      6. [AXI\_CS\_06] Spidrr\_cs signal shall be set when s\_axi\_araddr is equal to 0x[C\_BASEADDR + 6C] and s\_axi\_arvalid is set.  
           
         --SPI Slave Select Register Chip Select generation (R/W)
      7. [AXI\_CS\_07] Spissr\_cs signal shall be set when s\_axi\_awaddr is equal to 0x[C\_BASEADDR + 70] and s\_axi\_awvalid is set.
      8. [AXI\_CS\_08] Spissr\_cs signal shall be set when s\_axi\_araddr is equal to 0x[C\_BASEADDR + 70] and s\_axi\_arvalid is set.  
           
         --SPI Transmit FIFO Occupancy Register Chip Select generation (Read only)
      9. [AXI\_CS\_09] Tx\_fifo\_ocy\_cs signal shall be set when s\_axi\_araddr is equal to 0x[C\_BASEADDR + 74] and s\_axi\_arvalid is set.  
           
         --SPI Receive FIFO Occupancy Register Chip Select generation (Read only)
      10. [AXI\_CS\_10] Rx\_fifo\_ocy\_cs signal shall be set when s\_axi\_araddr is equal to 0x[C\_BASEADDR + 78] and s\_axi\_arvalid is set.  
            
          --Device Global Interrupt Enable Register Chip Select generation (R/W)
      11. [AXI\_CS\_11] Dgier\_cs signal shall be set when s\_axi\_awaddr is equal to 0x[C\_BASEADDR + 1C] and s\_axi\_awvalid is set.
      12. [AXI\_CS\_12] Dgier\_cs signal shall be set when s\_axi\_araddr is equal to 0x[C\_BASEADDR + 1C] and s\_axi\_arvalid is set.  
            
          --IP Interrupt Status Register Chip Select generation (R/TOW)
      13. [AXI\_CS\_13] Ipisr\_cs signal shall be set when s\_axi\_awaddr is equal to 0x[C\_BASEADDR + 20] and s\_axi\_awvalid is set.
      14. [AXI\_CS\_14] Ipisr\_cs signal shall be set when s\_axi\_araddr is equal to 0x[C\_BASEADDR + 20] and s\_axi\_arvalid is set.  
            
          --IP Interrupt Enable Register Chip Select generation (R/W)
      15. [AXI\_CS\_15] Ipier\_cs signal shall be set when s\_axi\_awaddr is equal to 0x[C\_BASEADDR + 28] and s\_axi\_awvalid is set.
      16. [AXI\_CS\_16] Ipier\_cs signal shall be set when s\_axi\_araddr is equal to 0x[C\_BASEADDR + 28] and s\_axi\_arvalid is set.
   3. Read Cycle Signaling (Tag : AXI\_RD\_)  
       --State S1: Setting up address for read
      1. [AXI\_RD\_01] Axi\_state shall be set to S1 when axi\_state is S0 and s\_axi\_arvalid is set.   
           
         --State S2: Broadcast
      2. [AXI\_RD\_02] Axi\_state shall be set to S2 when axi\_state is S1 and reg\_rack is set.  
           
         --End of broadcast: return to S0(waiting)
      3. [AXI\_RD\_03] Axi\_state shall be set to S0 when axi\_state is S2 and reg\_rack is reset.  
           
         --Register Read enable signal generation
      4. [AXI\_RD\_04] Reg\_read\_enable shall be set when s\_axi\_arvalid is set.  
           
         --Address read ready signal generation
      5. [AXI\_RD\_05] S\_axi\_arready shall be set when reg\_rack is set.  
           
         --Read data output latching
      6. [AXI\_RD\_06] S\_axi\_rdata shall be set equal to reg\_rdata\_latch when s\_axi\_rready is set.
      7. [AXI\_RD\_07] Reg\_rdata\_latch shall be set equal to reg\_rdata when reg\_rack is set.  
           
         --Read response bus error signal generation
      8. [AXI\_RD\_08] S\_axi\_rresp shall be set to 01 when reg\_rerror signal is set.  
           
         --Read valid signal generation
      9. [AXI\_RD\_09] S\_axi\_rvalid shall be set after one s\_axi\_aclk clock cycle when reg\_rack is set.
   4. Write Cycle Signaling (Tag : AXI\_WR\_)  
       --State S3: Writing to a register address
      1. [AXI\_WR\_01] Axi\_state shall be set to S3 when axi\_state is S0 and s\_axi\_awvalid is set and s\_axi\_wvalid is set and reg\_wack is reset.  
           
         --End of write sequence: return to S0(waiting)
      2. [AXI\_WR\_02] Axi\_state shall be set to S0 when axi\_state is S3 and reg\_wack is set.  
           
         --Write data passthrough
      3. [AXI\_WR\_03] Reg\_wdata shall be set to s\_axi\_wdata when s\_axi\_wvalid is set.  
           
         --Write data strobe passthrough
      4. [AXI\_WR\_04] Reg\_wstb shall be set to s\_axi\_wstb when s\_axi\_wvalid is set.  
           
         --Register Write enable signal generation
      5. [AXI\_WR\_05] Reg\_write\_enable shall be set when s\_axi\_wvalid is set.  
           
         --Address write ready signal generation
      6. [AXI\_WR\_06] S\_axi\_awready shall be set when reg\_wack is set.  
           
         --Write ready signal generation
      7. [AXI\_WR\_07] S\_axi\_wready shall be set when reg\_wack is set.  
           
         --Write response bus error signal generation
      8. [AXI\_WR\_08] S\_axi\_bresp shall be set to 01 when reg\_werror signal is set.  
           
         --Write valid signal generation
      9. [AXI\_WR\_09] S\_axi\_bvalid shall be set when reg\_wack is set.
2. SPI Interface (Tag Prefix : SPI)
   1. General requirements: (Tag : SPI\_GEN\_)
      1. [SPI\_GEN\_01] SCK\_T, MOSI\_T, MISO\_T shall all be initialized as low.
      2. [SPI\_GEN\_02] MOSI\_O, MISO\_O, and SCLK\_O shall all be initialized as high
   2. Input tristate signal: (Tag: SPI\_TRI\_)
      1. [SPI\_TRI\_01] The MOSI port shall be equal to MOSI\_O when MOSI\_T is high.
      2. [SPI\_TRI\_02] The MOSI\_I shall be equal to value at MOSI port when MOSI\_T is low.
      3. [SPI\_TRI\_03] The MISO port shall be equal to MISO\_O when MISO\_T is high.
      4. [SPI\_TRI\_04] The MISO\_I shall be equal to value at MISO port when MISO\_T is low.
      5. [SPI\_TRI\_05] The SCK port shall be equal to SCK\_O when SCK\_T is high.
      6. [SPI\_TRI\_06] The SCK\_I shall be equal to value at SCK port when SCK\_T is low.
   3. Baud Rate Generator (Tag : SPI\_BRG\_)
      1. [SPI\_BRG\_01] SCK\_O shall be set to the clock signal of frequency equal to that of SCK\_I divided by C\_SCK\_RATIO when the device is in the master mode.
   4. Serializer / Deserializer (Tag: SPI\_SR\_)
      1. [SPI\_SR\_01] The shift register shall perform a bit shift right on the rising edge of SCK\_O when lsb\_signal is high and cpha is low.
      2. [SPI\_SR\_02] The shift register shall perform a bit shift left on the rising edge of SCK\_O when lsb\_signal is low and cpha is low.
      3. [SPI\_SR\_03] The shift register shall perform a bit shift right on the falling edge of SCK\_O when lsb\_signal is high and cpha is high.
      4. [SPI\_SR\_04] The shift register shall perform a bit shift left on the falling edge of SCK\_O when lsb\_signal is low and cpha is high.
      5. [SPI\_SR\_05] Internal counter that indicates current bit position shall advance in range 0 to [C\_NUM\_TRANSFER\_BITS – 1] with value of [C\_NUM\_TRANSFER\_BITS – 1] corresponding to the complete message sent.
      6. [SPI\_SR\_06] Shift register shall perform a write into the Rx FIFO followed by read from Tx FIFO on internal counter value of [C\_NUM\_TRANSFER\_BITS – 1].
      7. [SPI\_SR\_07] The carry-in for shift register shall originate from MOSI\_I in slave mode.
      8. [SPI\_SR\_08] The carry-in for shift register shall originate from MISO\_I in master mode.
      9. [SPI\_SR\_09] The carry-out for shift register shall be connected to MISO\_O in slave mode.
      10. [SPI\_SR\_10] The carry-out for shift register shall be connected to MOSI\_O in master mode when master\_inhibit is low.
   5. Slave Selector (Tag : SPI\_SS\_)
      1. [SPI\_SS\_01] The SS(N) bus shall output the value written into the SPISSR register when manual\_ss\_en is high.
      2. [SPI\_SS\_02] The SS(N) bus shall advance through the SS bits starting with SS(0) in zero\_hot encoding manner at internal counter value of [C\_NUM\_TRANSFER\_BITS – 1] when manual\_ss\_en is low.
   6. Control Unit (Tag : SPI\_CU\_)
      1. [SPI\_CU\_01] The slave mode shall initiate for device when SPISEL\_I is reset.
      2. [SPI\_CU\_02] The slave mode shall initiate for device when spi\_master\_en is reset.
      3. [SPI\_CU\_03] The SCK\_O shall be inverted when cpol is set.
      4. [SPI\_CU\_04] The module shall be disabled when spi\_system\_en is reset.
      5. [SPI\_CU\_05] The MOSI and MISO lines shall be shorted when loopback\_en is set.
      6. [SPI\_CU\_06] Slave\_mode\_select shall be set when the SPISEL\_I is set.
      7. [SPI\_CU\_07] Slave\_mode\_select shall be set when the spi\_master\_en is set.
3. Registers (Tag prefix : REG\_)
   1. General/Shared register functionality (Tag prefix: REG\_GEN\_)
      1. Register Size (Tag : REG\_GEN\_SZ\_)
         1. [REG\_GEN\_SZ\_01] The register shall be instantiated to a size of 32 bits.
            1. [REG\_GEN\_SZ\_01\_EX] Exceptions are SPIDTR and SPIDRR registers, which initialize to a size of C\_NUM\_TRANSFER\_BITS bits.
      2. Data Write (Tag : REG\_GEN\_WR\_)
         1. [REG\_GEN\_WR\_01] The register shall only modify contents of the bits flagged by reg\_wstb by equivalent bits from reg\_wdata when reg\_write\_enable is set.
         2. [REG\_GEN\_WR\_02] The flagged bits shall set their value equal to the value of the corresponding bit from reg\_wdata.
            1. [REG\_GEN\_WR\_02\_EX] Exception is IPISR register, which follows toggle-on-write procedure. (refer to IPISR requirements)
         3. [REG\_GEN\_WR\_03] The signal reg\_wack shall be set when the register write has been completed for a duration of 1 clock cycle.
      3. Data Read (Tag : REG\_GEN\_RD\_)
         1. [REG\_GEN\_RD\_01] Reg\_rdata shall be set to the contents of the register when both reg\_read\_enable is set and corresponding chip select signal (e.g. srr\_cr for SRR) is set.
         2. [REG\_GEN\_RD\_02] The signal reg\_wack shall be set when the register write has been completed for a duration of 1 clock cycle.
      4. Chip select error correction (Tag : REG\_GEN\_ERR\_)
         1. [REG\_GEN\_ERR\_01] The register shall set reg\_werror and reg\_rerror when reg\_read\_enable is set and reg\_write\_enable is set.
   2. Software Reset Register (SRR) (Tag : REG\_SRR\_)
      1. [REG\_SRR\_01] Register shall initialize to 0x0000\_0000 upon system bootup.
      2. [REG\_SRR\_02] Register shall initialize to 0x0000\_0000 upon system reset.
      3. [REG\_SRR\_03] Soft\_reset shall be reset when the register content is equal to 0x0000\_0000.
      4. [REG\_SRR\_04] Soft\_reset shall be set when the register content is equal to 0x0000\_000A.
      5. [REG\_SRR\_05] Reg\_werror shall be set when srr\_cs is set and reg\_wdata is not equal to 0x0000\_000A.
   3. SPI Control Register (SPICR) (Tag: REG\_SPICR\_)
      1. [REG\_SPICR\_01] Register shall initialize to 0x0000\_0180 upon system bootup.
      2. [REG\_SPICR\_02] Register shall initialize to 0x0000\_0180 upon system reset.
      3. [REG\_SPICR\_03] Lsb\_first signal shall be set when bit 9 of the register is set.
      4. [REG\_SPICR\_04] Master\_inhibit signal shall be set when bit 8 of the register is set.
      5. [REG\_SPICR\_05] Manual\_ss\_en signal shall be set when bit 7 of the register is set.
      6. [REG\_SPICR\_06] Rx\_fifo\_reset signal shall be set when bit 6 of the register is set.
      7. [REG\_SPICR\_07] Tx\_fifo\_reset signal shall be set when bit 5 of the register is set.
      8. [REG\_SPICR\_08] Chpa signal shall be set when bit 4 of the register is set.
      9. [REG\_SPICR\_09] Cpol signal shall be set when bit 3 of the register is set.
      10. [REG\_SPICR\_10] Spi\_master\_en signal shall be set when bit 2 of the register is set.
      11. [REG\_SPICR\_11] Spi\_system\_en signal shall be set when bit 1 of the register is set.
      12. [REG\_SPICR\_12] Loopback\_en signal shall be set when bit 0 of the register is set.
   4. SPI Status Register (SPISR) (Tag : REG\_SPISR\_)
      1. [REG\_SPISR\_01] Bit 5 of the register shall be set when slave\_mode\_select is set.
      2. [REG\_SPISR\_02] Bit 4 of the register shall be set when mode\_fault\_error is set.
      3. [REG\_SPISR\_03] Bit 3 of the register shall be set when tx\_full is set.
      4. [REG\_SPISR\_04] Bit 2 of the register shall be set when tx\_empty is set.
      5. [REG\_SPISR\_05] Bit 1 of the register shall be set when rx\_full is set.
      6. [REG\_SPISR\_06] Bit 0 of the register shall be set when rx\_empty is set.
   5. Data Registers (Tag prefix unchanged)
      1. AXI to SPI Data Transmit Register (SPIDTR) (Tag: REG\_SPIDTR\_)
         1. [REG\_SPIDTR\_01] Register shall initialize to empty upon system bootup.
         2. [REG\_SPIDTR\_02] Register shall initialize to empty upon system reset.
         3. [REG\_SPIDTR\_03] Tx\_fifo\_data shall be of size of C\_NUM\_TRANSFER bits.
         4. [REG\_SPIDTR\_04] Tx\_fifo\_data shall be equal to the contents of the register.
      2. SPI to AXI Data Receive Register (SPIDRR) (Tag: REG\_SPIDRR\_)
         1. [REG\_SPIDRR\_01] Register shall initialize to empty upon system bootup.
         2. [REG\_SPIDRR\_02] Register shall initialize to empty upon system reset.
         3. [REG\_SPIDRR\_03] Contents of the register shall be equal to Rx\_fifo\_data.
   6. Slave Select Register (SPISSR) (Tag: REG\_SPISSR\_)
      1. [REG\_SPISSR\_01] Register shall initialize to 0x0000\_0001 upon system bootup.
      2. [REG\_SPISSR\_02] Register shall initialize to 0x0000\_0001 upon system reset.
      3. [REG\_SPISSR\_03] Bits [ [C\_NUM\_SS\_BITS -1] : 0] of the register shall be a one-hot encoded representation of the selected slave to transfer data to.
         1. [REG\_SPISSR\_03\_HOT] At most one bit of the register is allowed to be set.(one-hot encoding schema)
      4. [REG\_SPISSR\_04] Slave\_select shall be of size of C\_NUM\_SS\_BITS bits.
      5. [REG\_SPISSR\_05] Slave\_select shall be equal to the contents of the register.
   7. Transmit FIFO Occupancy Register (Tx\_FIFO\_OCY) (Tag : REG\_TXOCY\_)
      1. [REG\_TXOCY\_01] Register shall initialize to empty upon system bootup.
      2. [REG\_TXOCY\_02] Register shall initialize to empty upon system reset.
      3. [REG\_TXOCY\_03] Bits [3:0] of the register shall be equal to the tx\_fifo\_occupancy.
   8. Receive FIFO Occupancy Register (Rx\_FIFO\_OCY) (Tag : REG\_RXOCY\_)
      1. [REG\_RXOCY\_01] Register shall initialize to empty upon system bootup.
      2. [REG\_RXOCY\_02] Register shall initialize to empty upon system reset.
      3. [REG\_RXOCY\_03] Bits [3:0] of the register shall be equal to the rx\_fifo\_occupancy.
   9. Interrupt generation interface (Tag prefix unchanged)
      1. Global Interrupt Enable Register (DGIER) (Tag : REG\_DGIER\_)
         1. [REG\_DGIER\_01] Register shall initialize to empty upon system bootup.
         2. [REG\_DGIER\_02] Register shall initialize to empty upon system reset.
         3. [REG\_DGIER\_03] Bit 31 of the register shall be equal to the gi\_en.
      2. IP Interrupt Status Register (IPISR) (Tag : REG\_IPISR\_)
         1. [REG\_IPISR\_01] Register shall initialize to empty upon system bootup.
         2. [REG\_IPISR\_02] Register shall initialize to empty upon system reset.
         3. [REG\_IPISR\_03] Register bit 8 shall have its value inverted when bit 8 of reg\_wdata is set and bit 8 of reg\_wstrb is set.
         4. [REG\_IPISR\_04] Register bit 8 shall be set when register bit 8 is currently reset and drr\_not\_empty is set.
         5. [REG\_IPISR\_05] Register bit 7 shall have its value inverted when bit 7 of reg\_wdata is set and bit 7 of reg\_wstrb is set.
         6. [REG\_IPISR\_06] Register bit 7 shall be set when register bit 7 is currently reset and slave\_select\_mode is set.
         7. [REG\_IPISR\_07] Register bit 6 shall have its value inverted when bit 6 of reg\_wdata is set and bit 6 of reg\_wstrb is set.
         8. [REG\_IPISR\_08] Register bit 6 shall be set when register bit 6 is currently reset and tx\_fifo\_half\_empty is set.
         9. [REG\_IPISR\_09] Register bit 5 shall have its value inverted when bit 5 of reg\_wdata is set and bit 5 of reg\_wstrb is set.
         10. [REG\_IPISR\_10] Register bit 5 shall be set when register bit 5 is currently reset and drr\_overrun is set.
         11. [REG\_IPISR\_11] Register bit 4 shall have its value inverted when bit 4 of reg\_wdata is set and bit 4 of reg\_wstrb is set.
         12. [REG\_IPISR\_12] Register bit 4 shall be set when register bit 4 is currently reset and drr\_full is set.
         13. [REG\_IPISR\_13] Register bit 3 shall have its value inverted when bit 3 of reg\_wdata is set and bit 3 of reg\_wstrb is set.
         14. [REG\_IPISR\_14] Register bit 3 shall be set when register bit 3 is currently reset and dtr\_underrun is set.
         15. [REG\_IPISR\_15] Register bit 2 shall have its value inverted when bit 2 of reg\_wdata is set and bit 2 of reg\_wstrb is set.
         16. [REG\_IPISR\_16] Register bit 2 shall be set when register bit 2 is currently reset and dtr\_empty is set.
         17. [REG\_IPISR\_17] Register bit 1 shall have its value inverted when bit 1 of reg\_wdata is set and bit 1 of reg\_wstrb is set.
         18. [REG\_IPISR\_18] Register bit 1 shall be set when register bit 1 is currently reset and slave\_mode\_fault\_error is set.
         19. [REG\_IPISR\_19] Register bit 0 shall have its value inverted when bit 0 of reg\_wdata is set and bit 0 of reg\_wstrb is set.
         20. [REG\_IPISR\_20] Register bit 0 shall be set when register bit 0 is currently reset and mode\_fault\_error is set.
      3. IP Interrupt Enable Register (IPIER) (Tag : REG\_IPIER\_)
         1. [REG\_IPIER\_01] Register shall initialize to empty upon system bootup.
         2. [REG\_IPIER\_02] Register shall initialize to empty upon system reset.
         3. [REG\_IPIER\_03] Drr\_not\_empty\_int\_en is set when bit 8 of the register is set.
         4. [REG\_IPIER\_04] Ss\_mode\_int\_en is set when bit 7 of the register is set.
         5. [REG\_IPIER\_05] Tx\_fifo\_half\_int\_en is set when bit 6 of the register is set.
         6. [REG\_IPIER\_06] Drr\_overrun\_int\_en is set when bit 5 of the register is set.
         7. [REG\_IPIER\_07] Drr\_full\_int\_en is set when bit 4 of the register is set.
         8. [REG\_IPIER\_08] Dtr\_underrun\_int\_en is set when bit 3 of the register is set.
         9. [REG\_IPIER\_09] Dtr\_empty\_int\_en is set when bit 2 of the register is set.
         10. [REG\_IPIER\_10] Slave\_mode\_fault\_int\_en is set when bit 1 of the register is set.
         11. [REG\_IPIER\_11] Mode\_fault\_int\_en is set when bit 0 of the register is set.
4. Asynchronous FIFO (Tag prefix : FIFO\_)
   1. [FIFO\_01] FIFO structure is generated when C\_FIFO\_EXIST is high (instance generic)
      1. [FIFO\_01\_EX]Otherwise a basic double synchronizer is generated.
   2. [FIFO\_02] FIFO data width is tied to C\_NUM\_TRANSFER\_BITS (instance generic)
   3. [FIFO\_03] FIFO depth is always 16 elements.
   4. Double synchronizer for register signals (Tag: FIFO\_SYNC\_)
      1. [FIFO\_SYNC\_01] The AXI and SPI interfaces will double synchronize read and write data from/to the selected non-data registers.
      2. [FIFO\_SYNC\_01]Data register (SPIDRR and SPIDTR) synchronization is handled by FIFO modules.
   5. Tx FIFO (AXI to SPI) (Tag: FIFO\_TX\_)
      1. [FIFO\_TX\_01] Wdata tied to SPI Data Transmit Register (SPIDTR)
      2. [FIFO\_TX\_02] Rdata is tied to SPI Module data serializer input
      3. [FIFO\_TX\_03] w\_enable is set when transmit data register has been loaded
         1. Note: S\_AXI\_BVALID
      4. [FIFO\_TX\_04] r\_enable is set when SPI module requests data transfer
         1. Note: SPI serializer requests new byte (internal count)
      5. [FIFO\_TX\_05] Reset is tied to Tx FIFO Reset bit of SPI Control Register (SPICR[5])
      6. [FIFO\_TX\_06] wclk is tied to AXI clock (S\_AXI\_ACLK)
      7. [FIFO\_TX\_07] rclk is tied to SPI clock output (SCK\_O)
      8. [FIFO\_TX\_08] full\_flag is tied to Tx\_Full bit of SPI Status Register (SPISR[3])
      9. [FIFO\_TX\_09] empty\_flag is tied to Tx\_Empty bit of SPI Status Register (SPISR[2])
      10. [FIFO\_TX\_10] Occupancy value is tied to Tx\_FIFO\_OCY[3:0]
   6. Rx FIFO (SPI to AXI) (Tag : FIFO\_RX\_)
      1. [FIFO\_RX\_01] Wdata is tied to SPI module data output (shift register output)
      2. [FIFO\_RX\_02] Rdata tied to SPI Data Receive Register (SPIDRR)
      3. [FIFO\_RX\_03] w\_enable is set when SPI shift register is full (internal counter)
      4. [FIFO\_RX\_04] r\_enable is set when AXI register module requests data transfer
         1. Note: S\_AXI\_RRESP
      5. [FIFO\_RX\_05] Reset tied to Rx FIFO Reset bit of SPI Control Register (SPICR[6])
      6. [FIFO\_RX\_06] wclk is tied to SPI clock output (SCK\_O)
      7. [FIFO\_RX\_07] rclk is tied to AXI clock (S\_AXI\_ACLK)
      8. [FIFO\_RX\_08] full\_flag is tied to Rx\_Full bit of SPI Status Register (SPISR[1])
      9. [FIFO\_RX\_09] empty\_flag is tied to Rx\_Empty bit of SPI Status Register (SPISR[0])
      10. [FIFO\_RX\_10] Occupancy value is tied to Rx\_FIFO\_OCY [3:0]
   7. IP Interrupt Status Register Conditions (Tag: FIFO\_SR\_)
      1. [FIFO\_SR\_01] Bit 8 – DRR\_Not\_Empty is set when the Rx\_FIFO recieves the first data value during SPI transaction
      2. [FIFO\_SR\_02] Bit 6 – Tx FIFO Half Empty is set when the transmit FIFO occupancy decrements from “1000” to “0111”
      3. [FIFO\_SR\_03] Bit 5 – DRR Overrun is set when Rx FIFO full flag is 1 and a write operation is attempted by SPI bus.
      4. [FIFO\_SR\_04] Bit 4 – DRR Full is set with Rx FIFO full\_flag and one clock strobe to indicate end of SPI element transfer
      5. [FIFO\_SR\_05] Bit 3 – DTR Under run is set with Tx FIFO empty and SPI request. (slave only)
      6. [FIFO\_SR\_06] Bit 2 – DTR Empty is set with Tx FIFO empty to indicate end of SPI element transfer. (slave only)