Team C

Design Parameters

**AXI SPI REQUIREMENTS+**

**AXI Global System Signals**

**GENERIC PARAMETERS**

* AXI Base Address (C\_BASEADDR) and AXI High Address (C\_HIGHADDR) which must contain a valid address
* AXI Address Bus Width (C\_S\_AXI\_ADDR\_WIDTH) set to 32 bits
* AXI Data Bus Width (C\_S\_AXI\_DATA\_WIDTH) optionally 32 or 64 bits (default 32)
* Include receive and transmit FIFOs (C\_FIFO\_EXIST) optional flag to include fifo when set high (default high)
* SPI clock frequency ratio (C\_SCK\_RATIO) to take fraction of system clock for Baud Rate Generator of a value 2^x (default 32)
* Total number of slave select bits (C\_NUM\_SS\_BITS) integer value 1-32 (default 1)
* Select number of transfer bits (C\_NUM\_TRANSFER\_BITS) multiples of 8 no greater than 32 (default 8)

**RESET REQUIREMENTS**

* The system shall have an active low reset input (S\_AXI\_ARESETN) when active, set AXI outputs logic low except for read address ready bit (S\_AXI\_ARREADY) set to high
* When system reset active SPI interface should set (MOSI\_O, MISO\_O, SCK\_O, SS\_O) outputs to high impedance (high Z)
* When value of 0x0000\_000A is written to the SRR register enable software reset, any other write access generates an error condition with undefined results

**INTERRUPTS**

* If a value is logic high in any of the interrupt registers ( we set the interrupt output high

**AXI Write Address Channel Signals**

* AXI write address input (S\_AXI\_AWADDR) of 32 bit size
* AXI module shall have a write address ready output (S\_AXI\_AWREADY) initially set low, that is enabled the clock cycle after write response ready (S\_AXI\_BREADY) and write address valid (S\_AXI\_AWVALID) are set high and to be inverted in the next clock cycle

**AXI Write Channel Signals**

* The AXI SPI module shall have a 32 bit write data input (S\_AXI\_AWADDR).
* The AXI SPI module shall have a 4 bit write strobe (S\_AXI\_WSTB[3:0]) that indicates which byte lanes to update in memory
* AXI module shall have a write ready output (S\_AXI\_WREADY) initially set low, that is enabled the clock cycle after write response ready (S\_AXI\_BREADY) and write valid (S\_AXI\_WVALID) are set high and to be inverted in the next clock cycle

**AXI Write Response Channel Signals**

* AXI module shall have a write response valid output (S\_AXI\_BVALID) initially set low, that is enabled the clock cycle after write response ready (S\_AXI\_BREADY) and write address ready (S\_AXI\_AWREADY) and write ready (S\_AXI\_WREADY) are de-asserted one clock cycle after they are asserted. (1) assertion of readys -> (2) subsequent de-assertions -> (3) S\_AXI\_BVALID set high
* Write response (S\_AXI\_BRESP) outputs 2 bit signal based on status of write transaction
  + 00 – OKAY (normal response)
  + 10 – SLVERR (write address present, cannot complete transaction)
  + 11 – DECERR (not issued by core)

**AXI Read Address Channel Signals**

* AXI read address input (S\_AXI\_ARADDR) of 32 bit size
* AXI module shall have a address read ready output (S\_AXI\_ARREADY) initially set low, that is enabled the clock cycle after write response ready (S\_AXI\_ARVALID) and write valid (S\_AXI\_RREADY) are set high and to be inverted in the next clock cycle

**AXI Read Data Channel Signals**

* AXI module shall have a read response valid output (S\_AXI\_RVALID) initially set low, that is enabled the clock cycle after read address ready (S\_AXI\_ARREADY) are de-asserted one clock cycle after they are asserted and when read ready (S\_AXI\_RREADY) is set high. (1) assertion of ready -> (2) subsequent de-assertions -> (3) S\_AXI\_RVALID set high
* Read response (S\_AXI\_RRESP) outputs 2 bit signal based on status of write transaction
  + 00 – OKAY (normal response)
  + 10 – SLVERR (read address present, cannot complete transaction)
  + 11 – DECERR (not issued by core)

**SPI Signals**

**SPI CLOCK**

* AXI SPI module shall take an input clock signal into SCK\_I when SPI is configured as slave.
* AXI SPI module shall have a tri state output buffer for the SPI clock (SCK\_O) that is enabled when SCK\_T is active low and disabled when SCK\_T is active high.

**SPI MOSI**

* AXI SPI module shall take an input MOSI signal into MOSI\_I when SPI is configured as slave.
* AXI SPI module shall have a tri state output buffer for the MOSI signal (MOSI\_O) that is enabled when MOSI\_T is active low and disabled when SCK\_T is active high.

**SPI MISO**

* AXI SPI module shall take an input MISO signal into MISO\_I when SPI is configured as master.
* AXI SPI module shall have a tri state output buffer for the MISO signal (MISO\_O) that is enabled when MISO\_T is active low and disabled when MISO\_T is active high.

**SPISEL**

* AXI SPI module shall have a slave select signal (SPISEL) that allows for the receiving and transferring of data as a slave.
* The slave select signal line shall cause the interrupt signal (IP2INTC\_Irpt) to be asserted if the input is active low when the AXI SPI module is configured as a master.

**SPI SLAVE SELECT**

* AXI SPI module shall take (SS\_I[(C\_NUM\_SS\_BITS - 1):0]) signal when SPI is configured as slave.
* AXI SPI module shall have (SS\_O[(C\_NUM\_SS\_BITS - 1):0]) signal that initially is set high when SPI is configured as master and enabled when (SS\_O[(C\_NUM\_SS\_BITS - 1):0]) is set low.
* AXI SPI module shall have tri state output buffer for the (SS\_O[(C\_NUM\_SS\_BITS - 1):0]) signal that is enabled when (SS\_T) is active low and disable when (SS\_T) is active high.

**SPI Interface Requirements**

**SPI SRR Signal**

* SPI module shall be reset when receiving “0x0000000A” of (SRR) signal.

**SPI SPICR Signal**

* SPI module shall enable “LOOPBACK” mode when (loop\_en) is high and enable normal operation when (loop\_en) is low.
* SPI module shall be disabled and both master and slave outputs, inputs is ignored when (spi\_en) is low and enabled when (spi\_en) is high.
* SPI module shall be configured as slave when (master\_en) is low or master when (master\_en) is high.
* SPI module shall active high clock when (CPOL) is low and active low clock when (CPOL) is high.
* SPI module shall sample data at rising edge of clock when (CPHA) is low and sample data at falling edge of clock when (CPHA) is high.
* SPI module shall reset TX FIFO when (tx\_fifo\_rst) is high and TX FIFO is under normal operation when (tx\_fifo\_rst) is low.
* SPI module shall reset RX FIFO when (rx\_fifo\_rst) is high and RX FIFO is under normal operation when (rx\_fifo\_rst) is low.
* SPI module shall enable manual slave select mode when (mss) is high and enable automatic slave select mode when (mss) is low.
* SPI module shall not outputs data to slave when (mas\_tran\_dis) is high.
* SPI module shall outputs data to slave when (mas\_tran\_dis) is low.
* SPI module shall enable MSB first format when (LSB\_EN) is low and enable LSB first format when (LSB\_EN) is high.

**SPI SPISR Signal**

* SPI module shall set (rx\_empty) is high when (rx\_empty\_flag) is high and set (rx\_empty) is low when (rx\_empty\_flag) is low.
* SPI module shall set (rx\_full) is high when (rx\_full\_flag) is high and set (rx\_empty) is low when (rx\_full\_flag) is low.
* SPI module shall set (tx\_empty) is high when (tx\_empty\_flag) is high and set (tx\_empty) is low when (tx\_empty\_flag) is low.
* SPI module shall set (tx\_full) is high when (tx\_full\_flag) is high and set (tx\_empty) is low when (tx\_full\_flag) is low.
* SPI module shall set (MODF) high when (SSEL) is active while SPI module is configured as mater and set (MODF) low when no error is detected.
* SPI module shall set (slave\_mode\_select) high when (SSEL) is high and set (slave\_mode\_select) low when (SSEL) is low.

**AXI INTERFACE REQUIREMENTS**

**AXI Interface Write Address Channel Signals**

* AXI interface write address input (S\_AXI\_AWADDR) of 32 bit size
* AXI interface module shall have a write address ready output (S\_AXI\_AWREADY) initially set low, that is enabled the clock cycle after write response ready (S\_AXI\_BREADY) and write address valid (S\_AXI\_AWVALID) are set high and to be inverted in the next clock cycle.

**AXI Interface Write Channel Signals**

* The AXI interface module shall have a 32 bit write data input (S\_AXI\_AWADDR).
* The AXI interface module shall have a 4 bit write strobe (S\_AXI\_WSTB[3:0]) that indicates which byte lanes to update in memory
* AXI inerface module shall have a write ready output (S\_AXI\_WREADY) initially set low, that is enabled the clock cycle after write response ready (S\_AXI\_BREADY) and write valid (S\_AXI\_WVALID) are set high and to be inverted in the next clock cycle.

**AXI Interface Write Response Channel Signals**

* AXI interface module shall have a write response valid output (S\_AXI\_BVALID) initially set low, that is enabled the clock cycle after write response ready (S\_AXI\_BREADY) and write address ready (S\_AXI\_AWREADY) and write ready (S\_AXI\_WREADY) are de-asserted one clock cycle after they are asserted. (1) assertion of readys -> (2) subsequent de-assertions -> (3) S\_AXI\_BVALID set high
* Write response (S\_AXI\_BRESP) outputs 2 bit signal based on status of write transaction
  + 00 – OKAY (normal response)
  + 10 – SLVERR (write address present, cannot complete transaction)
  + 11 – DECERR (not issued by core)

**AXI Interface Write Transaction**

* AXI interface module shall output the value of the input S\_AXI\_WDATA, at the time of when both S\_AXI\_WREADY and S\_AXI\_WVALID are active high on the rising edge of the clock, to WriteToReg.
* WriteToReg shall continuously hold the data from S\_AXI\_WDATA at the time of when both S\_AXI\_WREADY and S\_AXI\_WVALID are active high on the rising edge of the clock until the next handshake.
* AXI interface module shall output the value of the input S\_AXI\_WSTB, at the time of when both S\_AXI\_WREADY and S\_AXI\_WVALID are active high on the rising edge of the clock, to Strobe.
* Strobe shall continuously hold the data from S\_AXI\_WSTB at the time of when both S\_AXI\_WREADY and S\_AXI\_WVALID are active high on the rising edge of the clock until the next handshake.
* AXI Interface module shall decode the S\_AXI\_AWADDR at the time of when both S\_AXI\_AWREADY and S\_AXI\_AWVALID are active high on the rising edge of the clock and output the value to one of the register enable outputs (SRR\_En, SPICR\_En, SPIDTR\_En, SPISSR\_En, DGIER\_En, IPISR\_En, and IPIER\_En) when S\_AXI\_BVALID and S\_AXI\_BREADY is high.
* The register enable outputs (SRR\_En, SPICR\_En, SPIDTR\_En, SPISSR\_En, DGIER\_En, IPISR\_En, and IPIER\_En) shall be decoded to the specifications below when S\_AXI\_BREADY is high and S\_AXI\_BVALID is high:
  + SRR\_En = ‘1’ when stored awaddr = C\_BASE\_ADDR + 0x40
  + SPICR\_En = ‘1’ when stored awaddr = C\_BASE\_ADDR + 0x60
  + SPIDTR\_En = ‘1’ when stored awaddr = C\_BASE\_ADDR + 0x68
  + SPISSR\_En = ‘1’ when stored awaddr = C\_BASE\_ADDR + 0x70
  + DGIER\_En = ‘1’ when stored awaddr = C\_BASE\_ADDR + 0x1C
  + IPISR\_En = ‘1’ when stored awaddr = C\_BASE\_ADDR + 0x20
  + IPIER\_En = ‘1’ when stored awaddr = C\_BASE\_ADDR + 0x28
* S\_AXI\_BRESP shall output “00” when S\_AXI\_BREADY is high, S\_AXI\_BVALID is high, and the stored awaddr is a decodable address
* S\_AXI\_BRESP shall output “01” when an address is a decodable read only address, S\_AXI\_BVALID is high, and S\_AXI\_BREADY is high
* S\_AXI\_BRESP shall output “11” when an address is not decodable, S\_AXI\_BVALID is high, and S\_AXI\_BREADY is high.

**AXI Interface Read Transaction**

* AXI interface module shall output the value within the read signals (SPICR\_Read, SPISR\_Read, SPIDRR\_Read, SPISSR\_Read, Tx\_FIFO\_COY\_Read, RX\_FIFO\_OCY\_Read, DGIER\_Read, IPISR\_Read,IPIER\_Read) depending on S\_AXI\_ARADDR at the rising edge time S\_AXI\_ARVALID = 1 and S\_AXI\_ARREADY = 1 to S\_AXI\_RDATA when S\_AXI\_RVALID and S\_AXI\_RREADY are high.
* S\_AXI\_RDATA shall output the following based on the address entered into S\_AXI\_ARADDR when both S\_AXI\_ARVALID and S\_AXI\_ARREADY are high at the time of the rising edge of a clock:
  + S\_AXI\_RDATA = SPICR\_Read when stored araddr = C\_BASE\_ADDR + 0x60
  + S\_AXI\_RDATA = SPISR\_Read when stored araddr = C\_BASE\_ADDR + 0x64
  + S\_AXI\_RDATA = SPIDRR\_Read when stored araddr = C\_BASE\_ADDR + 0x6C
  + S\_AXI\_RDATA = SPISSR\_Read when stored araddr = C\_BASE\_ADDR + 0x70
  + S\_AXI\_RDATA = Tx\_FIFO\_OCY\_read when stored araddr = C\_BASE\_ADDR + 0x74
  + S\_AXI\_RDATA = Rx\_FIFO\_OCY\_read when stored araddr = C\_BASE\_ADDR + 0x78
  + S\_AXI\_RDATA = DGIER\_Read when when stored araddr = C\_BASE\_ADDR + 0x1C
  + S\_AXI\_RDATA = IPISR\_Read when stored araddr = C\_BASE\_ADDR + 0x20
  + S\_AXI\_RDATA = IPIER\_Read when stored araddr = C\_BASE\_ADDR + 0x28
* S\_AXI\_RRESP shall equal “00” when the read address entered into S\_AXI\_ARADDR when both S\_AXI\_ARVALID and S\_AXI\_ARREADY are high at the time of the rising edge of a clock is a decodable read register address, S\_AXI\_RREADY is high, and S\_AXI\_RVALID is high
* S\_AXI\_RRESP shall equal “01” when the read address entered into S\_AXI\_ARADDR when both S\_AXI\_ARVALID and S\_AXI\_ARREADY are high at the time of the rising edge of a clock is a decodable write register address, S\_AXI\_RREADY is high, and S\_AXI\_RVALID is high
* S\_AXI\_RRESP shall equal “11” when the read address entered into S\_AXI\_ARADDR when both S\_AXI\_ARVALID and S\_AXI\_ARREADY are high at the time of the rising edge of a clock is a non-decodable register address, S\_AXI\_RREADY is high, and S\_AXI\_RVALID is high

**AXI Interface Read Address Channel Signals**

* AXI interface read address input (S\_AXI\_ARADDR) of 32 bit size
* AXI interface module shall have a address read ready output (S\_AXI\_ARREADY) initially set low, that is enabled the clock cycle after write response ready (S\_AXI\_ARVALID) and write valid (S\_AXI\_RREADY) are set high and to be inverted in the next clock cycle

**AXI Interface Read Data Channel Signals**

* AXI interface module shall have a read response valid output (S\_AXI\_RVALID) initially set low, that is enabled the clock cycle after read address ready (S\_AXI\_ARREADY) are de-asserted one clock cycle after they are asserted and when read ready (S\_AXI\_RREADY) is set high. (1) assertion of ready -> (2) subsequent de-assertions -> (3) S\_AXI\_RVALID set high
* Read response (S\_AXI\_RRESP) outputs 2 bit signal based on status of write transaction
  + 00 – OKAY (normal response)
  + 10 – SLVERR (read address present, cannot complete transaction)
  + 11 – DECERR (not issued by core)