**IP SPECIFICATION**

1.About IP:

IP(Intellectual Property) is a reusable IP system that consists of the peripherals APB and SPI. It supports both the RISC-V RI5CY and zero-riscy core.

The SPI master IP is a low frequency device that uses an APB bus of 32 bits wide as a bridge to connect to the SPI master. For data-storage, the IP uses a memory-map which will be controlled by the assigned addresses. It includes a FIFO to store the data that is received and needs to be transmitted via the TX and RX registers. It contains an event unit that can put the core to sleep and wake it up when there is an event or interrupt from a peripheral.

2.Key Features:

1. Standard SPI mode
2. Full Duplex System
3. Single master SPI and single slave SPI
4. Single master APB and single slave APB
5. Serial transfer in SPI
6. Clock divider
7. Asynchronous active low hard reset
8. Software reset
9. SPI transfer length
10. Appending Dummy Data
11. Separate TXFIFO and RXFIFO
12. Interrupt Configuration

3.Basic Architecture:

        The APB master initiates the transfer by selecting the APB slave peripheral present in the SPI master rtl later asserts the penable signal for the transfer based on the ready signal sent by the master via the APB interface as shown in fig 3.1. The received data is sent to the FIFO via the TX registers. The SPI master initiates its transfer and uses the RX register to get the data from the FIFO and send it to the SPI slave based on the SPILEN declared in the registers via the SPI interface.

Diagram

Description automatically generated

**Fig 3.1:** IP Basic Architecture

SPI Rtl has the APB slave signals and it will store the data into the FIFO. SPI rtl has the system clock and system reset connected to it and uses the clock divider to modulate the frequency and to generate the spi clk. The SPI master rtl has a CS signal which is of 4 bit so that it can select upto 4 slaves.  It has the mosi and miso signals which will be used based on the modes used i.e., only one mosi and miso signal is used for Standard SPI mode whereas four mosi and miso signals are used for Quad SPI mode as shown in fig 3.2.

Chart

Description automatically generated

**Fig 3.2:**IP RTL Block

4.APB, SPI Master Signals:

        The SPI master rtl uses the apb slave signals as well as the SPI master signals in the SPI master rtl i.e., as shown below :

Text

Description automatically generated

**Fig 4.1:**APB signals

As the spi master rtl has the apb slave, so that it will sample the addr, data and all the signals sent by the master and hence it has to be declared as input in the spi master rtl as shown in fig 4.1

Table

Description automatically generated

**Fig 4.2:**SPI signals

As the spi master rtl has to drive the sclk, cs and mosi data on the SPI interface, so those signals will be assigned as output as shown in fig 4.2.

5.Registers:

The registers used in the spi master rtl are as follows :

1. Status Register
2. CLKDIV
3. SPICMD Register
4. SPIADDR Register
5. SPILEN Register
6. SPIDUM Register
7. TXFIFO Register
8. RXFIFO Register
9. INTCFG Register

5.1 Status Register (WO):

        Status register is used to pass the configurations to the slave. These configurations consist of chip select, software reset, quad\_write, quad\_read, standard write and read modes as shown in fig 5.1.

The status register is 32-bit wide. The bits are allocated to the respective configurations as follows :

* Bit 0                  - Performs read in standard SPI mode
* Bit 1                - Performs write in standard SPI mode
* Bit 2                 - Performs read in quad SPI mode
* Bit 3                - Performs write in quad SPI mode
* Bit 4                 - Software reset
* Bit 5 : 7         - 0
* Bit 11: 8        - Chip select
* Bit 12: 31         - Unused
* Address for the SPI status register starts at 0x1A10\_2000 and the reset value is 0x000\_0000.
* When software reset is asserted, then the address will be 0x000\_0000.

Table

Description automatically generated

**Fig 5.1:**bit allocation for status register

5.1.1 Standard SPI mode :

        In standard spi mode, mosi signal acts as master-out-slave-in signal to send the data serially to the spi slave and miso acts as master-in-slave-out signal to receive the data sent by slave spi. Bit 0 and bit 1 are used to read and write modes for standard spi respectively.

5.1.2 Quad SPI mode :

        In quad spi mode, four mosi signals acts as master-out-slave-in to send the data serially to the spi slave and four miso signals acts as master-in-slave-out to receive the data sent by slave spi. Bit 2 and bit 3 are used to read and write modes for quad spi, respectively.

5.1.3 Software Reset :

        Software reset clears the fifo and aborts the current transfers. When bit 4 is asserted, then the software reset is asserted.

5.1.4 Reserved Bits:

        These are reserved bits i.e., from bits 5 to 7

5.1.5 Chip Select :

        It is of 4 bits which is used to select slaves which means it can support upto four slaves which will one hot encoding so that one slave only is selected at a time. It is configured in bits 8 to 11.

5.2 CLKDIV(RW) :

        Clock divider is used to divide the system clock for SPI transfers. Consider APB has frequency of 80Mhz but the spi supports only 8Mhz then with the help of clock divider the required frequency is generated for the spi. This clock divider value should not change during a transfer. The bit allocation for the clock divider is as shown in fig 5.2. Clock divider register is 32 bits wide. 31 to 8 bits are unused bits whereas 7 to 0 bits are allocated for clock divider value.

Address starts at 0x1A10\_2004 and the reset value is 0x0000\_0000.

Graphical user interface, application

Description automatically generated

**Fig 5.2:**bit allocation for clock divider

5.3 SPICMD Register (SPI Command) (RW):

SPI Command Register is used when performing a read or write transfer. The SPI command is sent first before any data is read or written. This register is 32-bit wide. The length of the SPI command can be controlled with the SPIELEN register.

Address starts at 0x1A10\_2008 and the reset value is 0x0000\_0000.

Shape

Description automatically generated

**Fig 5.3**. : SPI Command Register

5.4 SPIADR Register (SPI Address)(RW)

SPI Address Register is used when performing a read or write transfer. The SPI command is sent first before any data is read or written, after this the SPI address is sent. This register is 32-bit wide. The length of the SPI address can be controlled with the SPILEN register.

Address starts at 0x1A10\_200C and the reset value is 0x0000\_0000.

Shape

Description automatically generated with low confidence

**Fig. 5.4:** SPI Address Register

5.5 SPILEN Register (SPI Transfer Length) (RW):

        SPI transfer length register is used to control the length of SPI command register, SPI address register and data register.

Address starts at 0x1A10\_2010 and the reset value is 0x0000\_0000.

The bit allocations are as shown in the fig. 5.5:

Table

Description automatically generated with medium confidence

**Fig. 5.5:** SPI Transfer Length

* Bit 31:16 – DATALEN

SPI Data Length bits to be controlled are assigned using DATALEN bits. The number of bits read or written is given by this DATALEN value. Note that first the SPI command and address are written to an SPI slave device.

* Bit 13:8 – ADDRLEN

SPI Address Length bits to be controlled are assigned using ADDRLEN bits. The number of bits of the SPI address that should be sent.

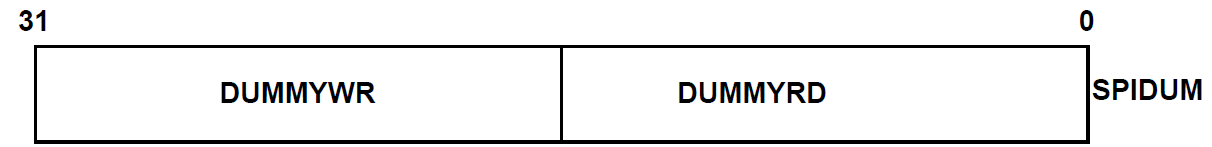
* Even Though command register is 32 bit wide, ADDRLEN is 6 bit wide only to accommodate the 32nd bit.
* Bit 5:0 CMDLEN

The number of bits of the SPI command that should be sent is assigned using SPI Command Length.

* Even Though command register is 32 bit wide, CMDLEN is 6 bit wide only to accommodate the 32nd bit.

5.6 SPIDUM (SPI Dummy Cycles)(RW):

* SPI Dummy Cycles is where nothing is written or read.
* Address starts at 0x1A10\_2014 and the reset value is 0x0000\_0000.
* The bit allocations are as shown in the fig 5.6:



**Fig 5.6:** spi dummy cycle bits allocation

* Bit 31:16 - **DUMMYWR**: Write Dummy Cycles.

Dummy cycles (nothing being written or read) between sending the SPI command

SPI address and writing the data as shown in the below fig 5.7

* Bit 15:0 -**DUMMYRD**: Read Dummy Cycles.

Dummy cycles (nothing being written or read) between sending the SPI command +

SPI address and reading the data as shown in the below fig 5.7

Diagram

Description automatically generated

**Fig 5.7:** Adding Dummy cycle during Transfer

5.7 TXFIFO (SPI Transmit FIFO) :WO

* SPI Transmit fifo is used to store write data into the fifo.
* Address starts at 0x1A10\_2018 and the reset value is 0x0000\_0000.
* It is a 32bit wide register as shown in the below fig 5.8:

Table

Description automatically generated with medium confidence

**Fig 5.8:**SPI Transmit FIFO

TX-FIFO is not a register ,but TX-regs is connected to fifo. Where APB writes the values into the TX-regs and TX-regs values go into the TX-FIFO as shown in the below fig 5.9. Since the register is 32bit wide ,we can transfer only 4 bytes of data to TX-regs .TX\_FIFO has the capacity to store a large number of transfers.

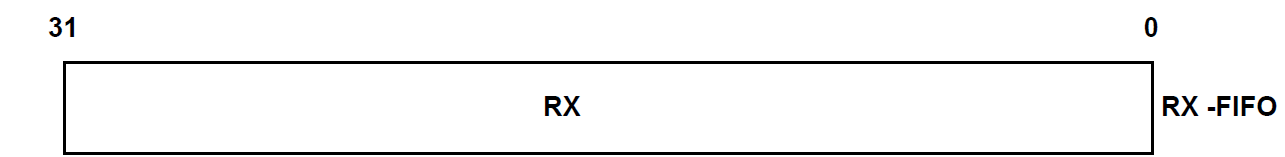
Chart

Description automatically generated with medium confidence

**Fig 5.9**: Internal connection of TX-FIFO

5.8 RXFIFO (SPI Receive FIFO) :RO

* SPI Receive fifo is used to read data from the fifo.
* Address starts at 0x1A10\_2020 and the reset value is 0x0000\_0000.
* It is a 32bit wide register as shown in the below fig 5.10:



**Fig 5.10:** SPI Receive FIFO

It is similar to the TX-FIFO ,but in the receive fifo we are reading data from the fifo .RX-regs values read from RX-FIFO to APB read interface as shown in fig 5.11:

Diagram, box and whisker chart

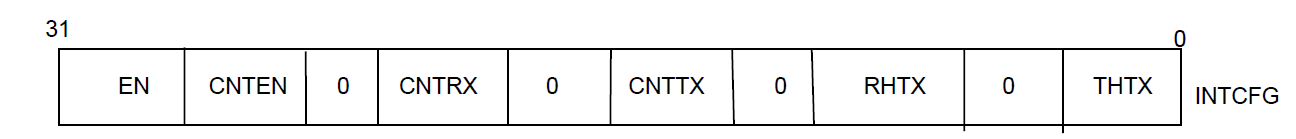
Description automatically generated with medium confidence

**Fig 5.11:** Internal connection of RX-FIFO

5.9 INTCFG (Interrupt Configuration) (RW

):

* Interrupt Configurations are used to enable the events.
* Address starts at 0x1A10\_2024 and the reset value is 0x0000\_0000.
* The bit allocations are as shown in the fig.5.12:



**Fig 5.12**: Interrupt Configuration

* Bit 31- **EN**: Interrupt Enable.
* Bit 30-**CNTEN:**count enable

When EN is zero event(0) is also zero.

* **THTX[4:0]**: Threshold of TX

In TX-FIFO ,once FIFO reaches the threshold value of TX  and  the interrupt is enabled(EN =1), Since, the event(0) is set to 1.

* **RHTX[12:8]**: Threshold of RX

In RX-FIFO ,once FIFO reaches the threshold value of RX  and the interrupt is enabled(EN =1), Since, the event(0) is set to 1.

Diagram

Description automatically generated

**Fig 5.13:** THTX and RHTX fifo

The above figure refers to THTX and RHTX fifo. When TX threshold is set 2 ,the data transfer to the TX FIFO is done continuously until it reaches the threshold value . Once the fifo pointer reaches the threshold value and enable is also 1, then event(0) is set to 1.Similarly for RHTX, once RX fifo reaches the threshold value and enable is 1 , then event(0) is set to 1.

* **CNTTX** [20:16]:Count fot TX

CNTTX is used to count the number of elements in the TX-FIFO.

* **CNTRX[28:24]:** Count for RX

CNTTX is used to count the number of elements in the RX-FIFO.