

# AXI-Lite Quad-SPI v1.0

---

*IP User Guide(Beta Release)*



November 26, 2023

# Contents

|                                       |           |
|---------------------------------------|-----------|
| <b>IP Specifications</b>              | <b>2</b>  |
| <b>Overview</b>                       | <b>3</b>  |
| AXI-Lite QuadSPI .....                | 3         |
| Licensing .....                       | 4         |
| <b>IP Specification</b>               | <b>5</b>  |
| Overview .....                        | 5         |
| Standards .....                       | 5         |
| IP Support Details .....              | 6         |
| Resource Utilization .....            | 6         |
| Port List .....                       | 7         |
| Parameters .....                      | 8         |
| <b>Design Flow</b>                    | <b>9</b>  |
| IP Customization and Generation ..... | 9         |
| Registers Address Space .....         | 11        |
| CSRs Description .....                | 11        |
| <b>Example Design</b>                 | <b>12</b> |
| Simulating the Example Design .....   | 12        |
| Synthesis and PnR .....               | 13        |
| <b>Release</b>                        | <b>14</b> |
| Revision History .....                | 14        |

# IP Summary

## Introduction

The AXIL QUADSPI core is a configurable FPGA core that provides communication with external flash memory devices.

External flash memory is commonly used in FPGA designs to store the FPGA bitstream and CPU firmware. AXIL QUADSPI can be useful in any FPGA design that requires access to external flash memory for storing and accessing the FPGA bitstream and CPU firmware.

## Features:

### PHY

- Portable/Generic.
- Single/Dual/Quad/Octal SPI Bus support.
- Dynamic Clk frequency configuration and auto-calibration.

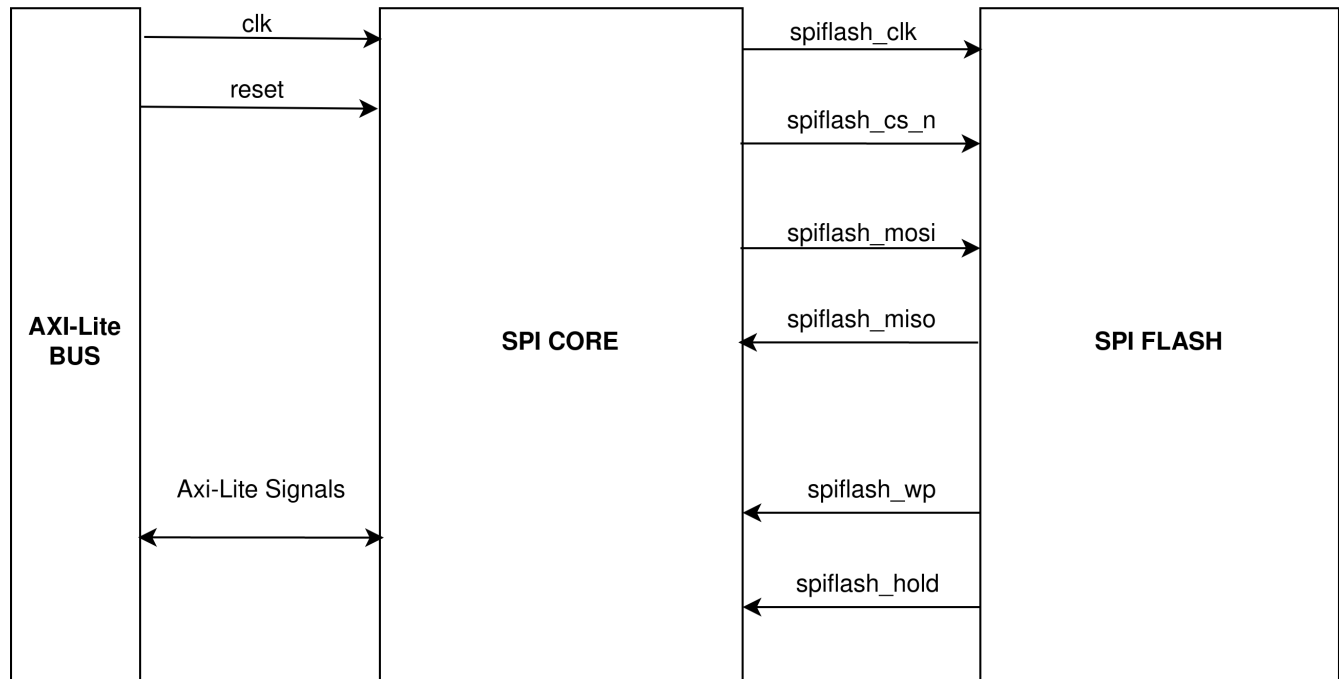
### CORE

- Dynamic Crossbar.
- MMAP read accesses.
- CSR-based read/write accesses.

# Overview

## AXI-Lite QuadSPI

AXIL\_QuadSPI is configured via its AXI lite slave interface. It configures the SPI core to communicate with the SPI Flash. The figure 1 shows the AXIL QuadSPI IP core connection with a SPI Flash on an FPGA.



**Figure 1.** AXIL QuadSPI BLock with an SPI Flash

## Licensing

Copyright (c) 2022 RapidSilicon

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

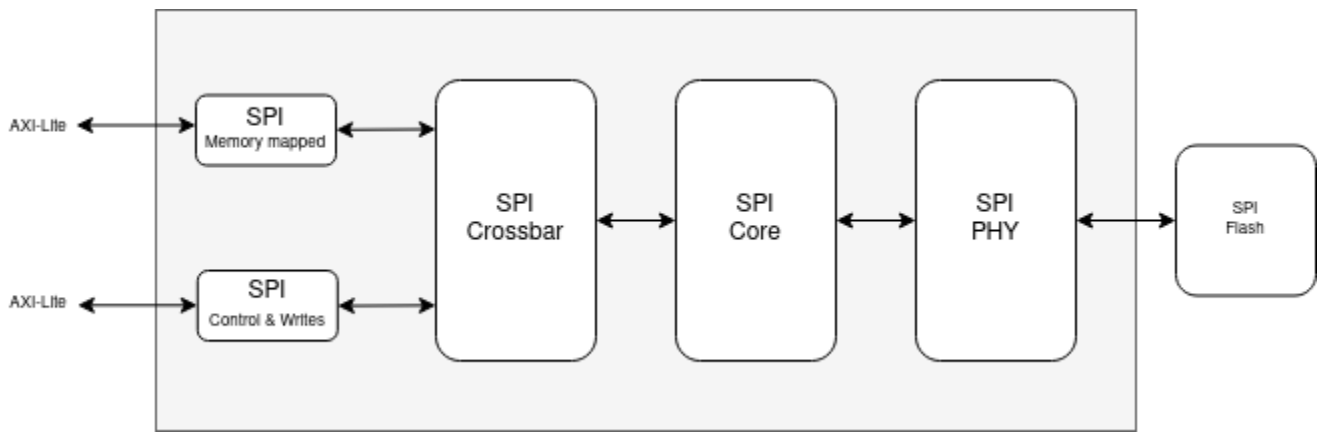
---

# IP Specification

## Overview

AXIL QUADSPI supports three types of SPI flash access: single, dual, and quad. Single access uses a single data line to transfer data between the host system and the flash memory, while dual access uses two data lines and quad access uses four data lines. Using more data lines allows for higher data transfer rates, but may not be supported by all flash memory devices.

AXIL QUADSPI allows the user to select the type of SPI flash that is connected to the core, and configure the type of access that will be used. This allows the user to optimize the performance of the core for the specific flash memory device that is being used. For example, if the flash memory device supports quad access, the user can configure AXIL QUADSPI to use quad access for maximum data transfer performance.



**Figure 2.** IP Core

- LiteSPIPHY: This component generates the clock signals and provides a physical interface for communicating with the external flash memory device.
- LiteSPICore: This component manages the data transfer between the external flash memory and the frontend modules. It includes a crossbar component, which routes data between the various frontend modules.
- LiteSPIMMAP: This component provides an interface for mapping the external flash memory into the host system's memory space.
- LiteSPIMaster: This component provides an interface for executing control commands on the external flash memory and do writes.

## Standards

The AXI4-Lite Slave interface is compliant with the AMBA® AXI Protocol Specification.

## IP Support Details

| Compliance |           | IP Resources |                 |           |                  |                 | Tool Flow               |            |           |
|------------|-----------|--------------|-----------------|-----------|------------------|-----------------|-------------------------|------------|-----------|
| Device     | Interface | Source Files | Constraint File | Testbench | Simulation Model | Software Driver | Analyze and Elaboration | Simulation | Synthesis |
| GEMINI     | AXI4-lite | Verilog      | SDC             | Verilog   | -                | -               | Raptor                  | Raptor     | Raptor    |

## Resource Utilization

| Tool             |                     | Raptor Design Suite |  |  |               |                      |          |  |  |
|------------------|---------------------|---------------------|--|--|---------------|----------------------|----------|--|--|
| FPGA Device      |                     | GEMINI              |  |  |               |                      |          |  |  |
| Configuration    |                     |                     |  |  |               | Resource Utilization |          |  |  |
| Minimum Resource | Options             |                     |  |  | Configuration | Resources            | Utilized |  |  |
|                  | CORE MODULE         |                     |  |  | S25FL128L     | LUT                  | 778      |  |  |
|                  | CORE MODE           |                     |  |  | x1            | Registers            | 492      |  |  |
|                  | CORE RATE           |                     |  |  | 1:1           | BRAM                 | 0        |  |  |
|                  | CORE BUS ENDIANNESS |                     |  |  | big           | DSP                  | 0        |  |  |
| Maximum Resource | Options             |                     |  |  | Configuration | Resources            | Utilized |  |  |
|                  | CORE MODULE         |                     |  |  | S25FL128L     | LUT                  | 778      |  |  |
|                  | CORE MODE           |                     |  |  | x4            | Registers            | 492      |  |  |
|                  | CORE RATE           |                     |  |  | 1:2           | BRAM                 | 0        |  |  |
|                  | CORE BUS ENDIANNESS |                     |  |  | big           | DSP                  | 0        |  |  |

## Ports

Table 2 lists the top interface ports of the AXIL QuadSPI.

| Signal Name                       | I/O | Description                    |
|-----------------------------------|-----|--------------------------------|
| <b>AXI Clock and Reset</b>        |     |                                |
| clk                               | I   | AXI4-Lite Clock                |
| rst                               | I   | AXI4-Lite RESET                |
| <b>SPI AXI SLAVE INTERFACE</b>    |     |                                |
| <b>AXI WRITE ADDRESS CHANNEL</b>  |     |                                |
| s_axil_awvalid                    | I   | AXI4-Lite Write address valid  |
| s_axil_awready                    | O   | AXI4-Lite Write address ready  |
| s_axil_awaddr                     | I   | AXI4-Lite Write address        |
| s_axil_awprot                     | I   | AXI4-Lite Protection type      |
| <b>AXI WRITE DATA CHANNEL</b>     |     |                                |
| s_axil_wvalid                     | I   | AXI4-Lite Write valid          |
| s_axil_wready                     | O   | AXI4-Lite Write ready.         |
| s_axil_wdata                      | I   | AXI4-Lite Write data           |
| s_axil_wstrb                      | I   | AXI4-Lite Write strobes        |
| <b>AXI WRITE RESPONSE CHANNEL</b> |     |                                |
| s_axil_bvalid                     | O   | AXI4-Lite Write response valid |
| s_axil_bready                     | I   | AXI4-Lite Response ready       |
| s_axil_bresp                      | O   | AXI4-Lite Write response       |
| <b>AXI READ ADDRESS CHANNEL</b>   |     |                                |
| s_axil_arvalid                    | I   | AXI4-Lite Read address valid   |
| s_axil_arready                    | O   | AXI4-Lite Read address ready   |
| s_axil_araddr                     | I   | AXI4-Lite Read address         |
| s_axil_arprot                     | I   | AXI4-Lite Protection type      |
| <b>AXI READ DATA CHANNEL</b>      |     |                                |
| s_axil_rvalid                     | I   | AXI4-Lite Read valid           |
| s_axil_rready                     | O   | AXI4-Lite Read ready           |
| s_axil_rresp                      | I   | AXI4-Lite Read data            |
| s_axil_rdata                      | O   | AXI4-Lite Read response        |
| <b>SPI AXI MASTER INTERFACE</b>   |     |                                |
| <b>AXI WRITE ADDRESS CHANNEL</b>  |     |                                |
| m_axil_awvalid                    | I   | AXI4-Lite Write address valid  |
| m_axil_awready                    | O   | AXI4-Lite Write address ready  |
| m_axil_awaddr                     | I   | AXI4-Lite Write address        |
| m_axil_awprot                     | I   | AXI4-Lite Protection type      |
| <b>AXI WRITE DATA CHANNEL</b>     |     |                                |
| m_axil_wvalid                     | I   | AXI4-Lite Write valid          |
| m_axil_wready                     | O   | AXI4-Lite Write ready.         |
| m_axil_wdata                      | I   | AXI4-Lite Write data           |
| m_axil_wstrb                      | I   | AXI4-Lite Write strobes        |
| <b>AXI WRITE RESPONSE CHANNEL</b> |     |                                |
| m_axil_bvalid                     | O   | AXI4-Lite Write response valid |
| m_axil_bready                     | I   | AXI4-Lite Response ready       |
| m_axil_bresp                      | O   | AXI4-Lite Write response       |



| AXI READ ADDRESS CHANNEL |   |                                     |
|--------------------------|---|-------------------------------------|
| m_axil_arvalid           | I | AXI4-Lite Read address valid        |
| m_axil_arready           | O | AXI4-Lite Read address ready        |
| m_axil_araddr            | I | AXI4-Lite Read address              |
| m_axil_arprot            | I | AXI4-Lite Protection type           |
| AXI READ DATA CHANNEL    |   |                                     |
| m_axil_rvalid            | I | AXI4-Lite Read valid                |
| m_axil_rready            | O | AXI4-Lite Read ready                |
| m_axil_rresp             | I | AXI4-Lite Read data                 |
| m_axil_rdata             | O | AXI4-Lite Read response             |
| SPI PORTS                |   |                                     |
| spiflash_cs_n            | O | SPI Flash chip select               |
| spiflash_clk             | O | SPI Flash clock                     |
| spiflash_mosi            | O | SPI Flash Master output slave input |
| spiflash_miso            | I | SPI Flash Master input slave output |
| spiflash_wp              | I | SPI Flash write protect             |

## SPI Interface

## Parameters

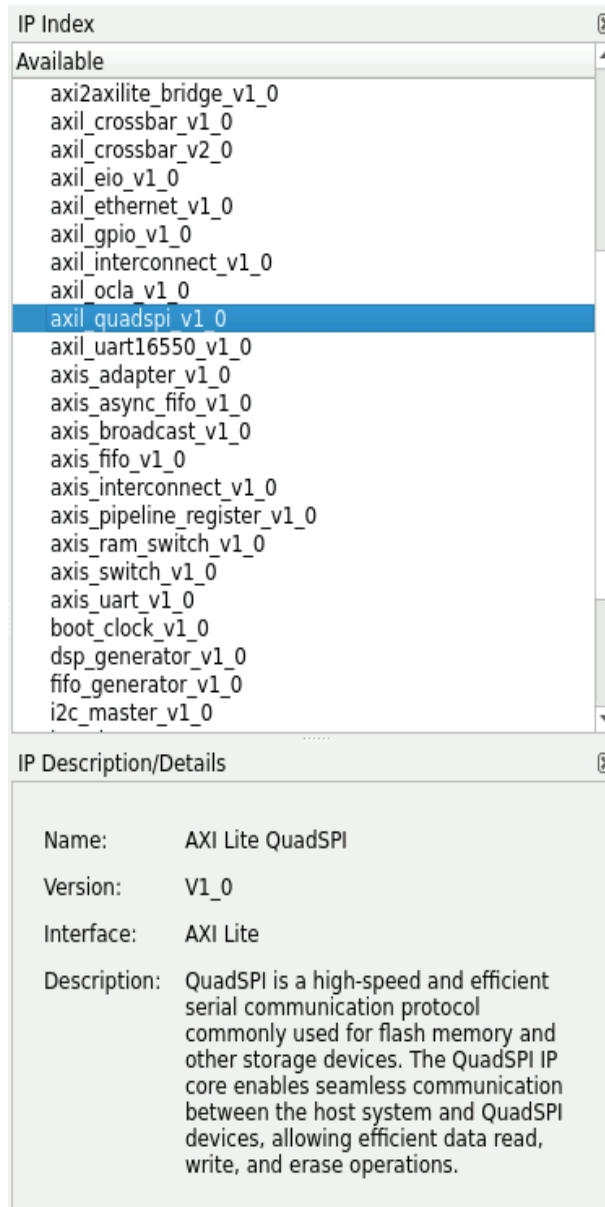
Table 2 lists the parameters of the AXIL QuadSPI.

| Parameter           | Values      | Default Value | Description                              |
|---------------------|-------------|---------------|--|
| CORE MODULE         | S25FL128L   | S25FL128L     | SPI Flash Module.                        |
| CORE MODE           | x1, x4      | x4            | SPI Modes, quad or simple.               |
| CORE RATE           | 1:1, 1:2    | 1:1           | SPI Flash Core rate SDR, DDR.            |
| CORE BUS ENDIANNESS | littel, big | big           | Bus Endianness (big, little)             |
| CORE PHY            | real, model | real          | Type or PHY (Real or Model (Simulation)) |
| CORE DIVISOR        | 1-256       | 1             | SPI Clock Dividor                        |

# Design Flow

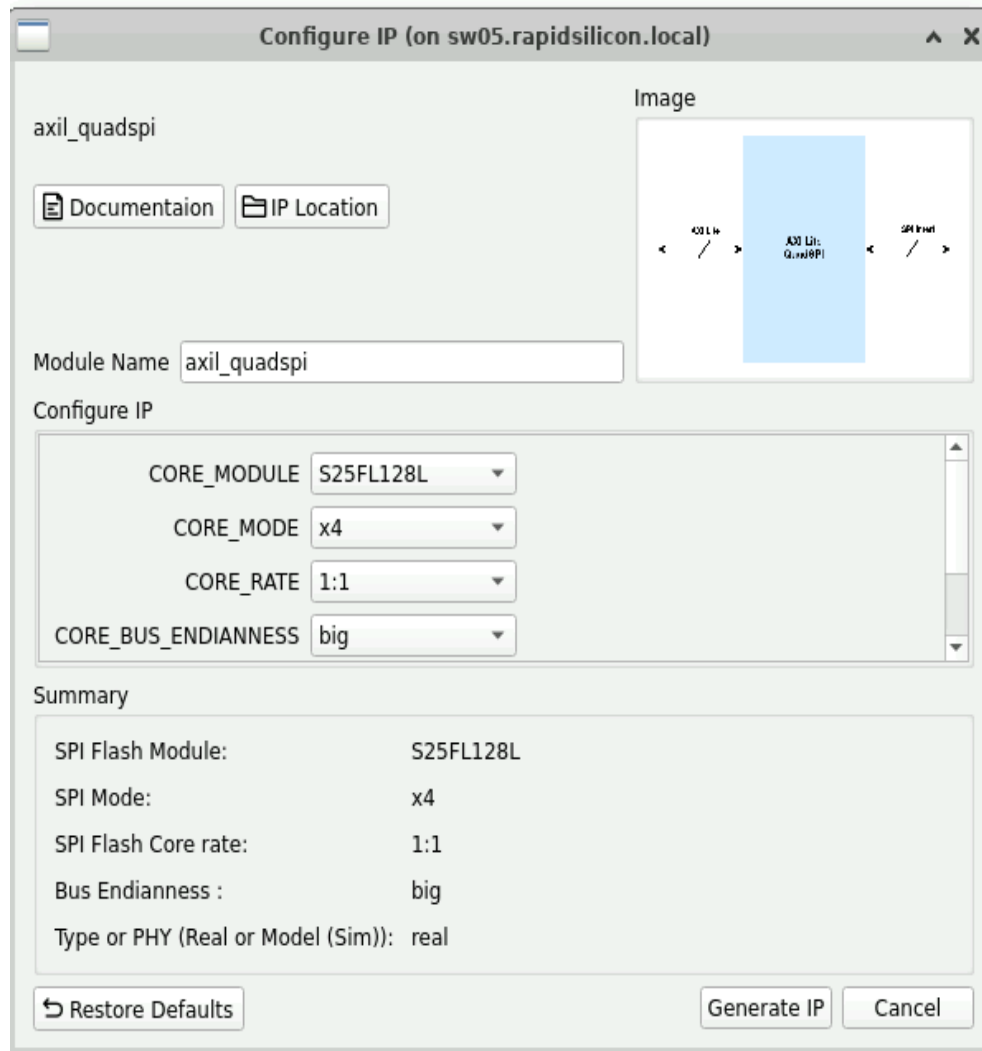
## IP Customization and Generation

AXIL QuadSPI IP core is a part of the Raptor Design Suite Software. A customized SPI can be generated from the Raptor’s IP configurator window.



Raptor IP list

**Parameters Customization:** From the IP configuration window, the parameters of the AXIL QuadSPI can be configured and the modes can be enabled for generating a customized SPI IP core that suits the user application requirement as shown in Figure. After IP Customization, all the source files are made available to the user with a top wrapper that instantiates a parameterized instance of the AXIL QUADSPI.



Configure IP (on sw05.rapidsilicon.local)

axil\_quadspi

Documentation IP Location

Module Name axil\_quadspi

Image

Configure IP

CORE\_MODULE S25FL128L

CORE\_MODE x4

CORE\_RATE 1:1

CORE\_BUS\_ENDIANNESNESS big

Summary

SPI Flash Module: S25FL128L

SPI Mode: x4

SPI Flash Core rate: 1:1

Bus Endianness : big

Type or PHY (Real or Model (Sim)): real

Restore Defaults Generate IP Cancel

IP Configuration

## Registers Address Space

Configuration Registers

| Name                    | Register ID | Bits | Type | Off sets | Default Value | Description                                     |
|-------------------------|-------------|------|------|----------|---------------|---|
| Control Stream Register | CS          | 1    | RW   | 0x00     | 0x0           | SPI flash core CS read/write                    |
| PHY Config Register     | phyconfig   | 32   | RW   | 0x04     | 0x00000000    | SPI PHY settings                                |
| RXTX FIFO Register      | rxtx        | 2    | RW   | 0x08     | 0x00000000    | Reciever and Transmitter FIFO full/empty status |

Table 3 lists the configuration registers of the AXIL\_QuadSPI.

## CSRs Description

**Control Stream Register:** Control stream register is a single bit register at 0x0 address that sets control stream to write or read.

**PHY Config Register:** PHY configuration register is used to configure PHY for the type of transfers to make.

| Bits     | Description  | Bitfield | Values   | Configuration              |
|----------|--|----------|----------|----------------------------|
| Length   | SPI transfer length (in bits) (1/2/4/8)  | [7:0]    | 8        | Send Command Configuration |
| Width    | SPI transfer width (1/2/4/8)   | [11:8]   | 1        | Send Command Configuration |
| RESERVED | RESERVED   | [15:12]  | RESERVED | RESERVED                   |
| Mask     | SPI DQ output enable mask (set bits to “1” to enable output drivers on DQ lines) | [23:16]  | 11       | Send Command Configuration |
| RESERVED | RESERVED   | [31:24]  | RESERVED | RESERVED                   |

**RXTX Fifo Register:** RXTX Fifo Register updates status of TX/RX fifo empty or full.

| Bits     | Description          | Bitfield | Values  | Configuration     |
|----------|----------------------|----------|---------|-------------------|
| TX_ready | TX FIFO is not full  | [0]      | tx_data | Transmitting data |
| RX_Ready | RX FIFO is not empty | [1]      | rx_data | Reading RX data   |

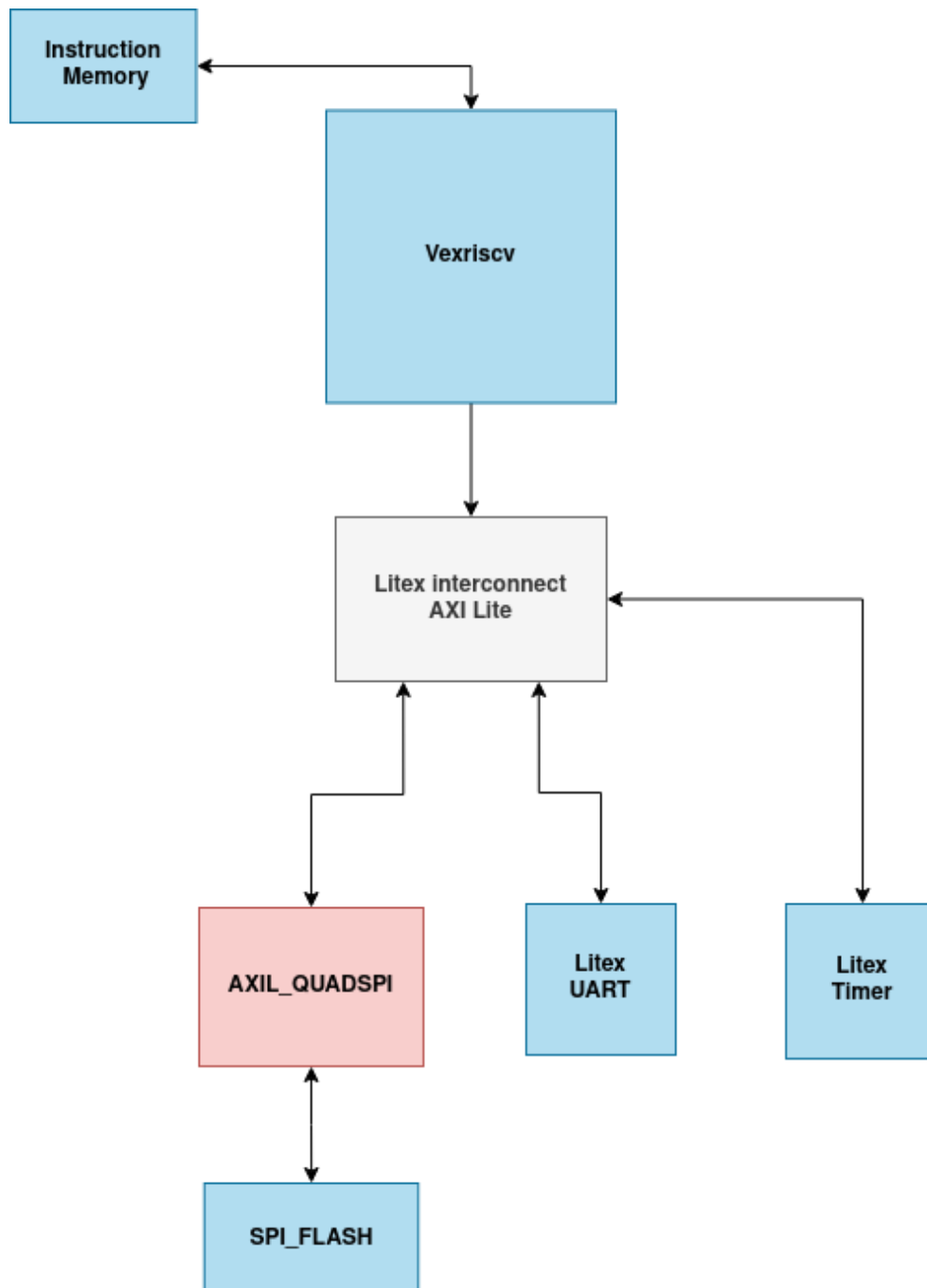
# Example Design

## Overview

### Simulating the Example Design

The IP being Verilog HDL, can be simulated via a bunch of industry standard stimulus. The bundled example design is simulated via a LiteX based SoC design where the IP is simulated by driving it from within the generated LiteX SoC. We read and write from the SPI flash attached with the controller.

The figure shows the AXIL QuadSPI IP integrated in a SoC environment for verification.



## Synthesis and PnR

Raptor Suite is armed with tools for Synthesis along with Post and Route capabilities and the generated post- synthesis and post-route and place netlists can be viewed and analyzed from within the Raptor. The generated bitstream can then be uploaded on an FPGA device to be utilized in hardware application

# Revision History

| Date              | Version | Revisions  |
|-------------------|---------|--|
| November 26, 2023 | 0.01    | Initial version AXIL QUADSPI User Guide Document |