**I2S\_in.v Document – Zachary Nelson**

**Interfaces:**

|  |  |  |  |
| --- | --- | --- | --- |
| Signal Name | Direction | Bits | Comment |
| clk | in | 1 | Master Clock |
| rst | in | 1 | Reset |
| sck | in | 1 | Digital Audio Bit Clock (max of 48kHz) |
| ws | in | 1 | Word Select (Left/Right Audio Channel) |
| sd | in | 1 | Digital Audio Serial Data |
| din\_rtr | in | 1 | Ready to Receive |
| rf\_bist\_en | in | 1 | Built in Self-Test (BIST) |
| din\_lft | out | 16 | Left Parallel Digital Audio |
| din\_rgt | out | 16 | Right Parallel Digital Audio |
| din\_rts | out | 1 | Ready to Send |
| ro\_fifo\_overrrun | out | 1 | Input Audio FIFO Overrun |

**Functional Requirements:**

* Data Plane Requirements:
  + If the BIST control bit is true, the output of the block will be pre-defined such as a fixed sawtooth wave. If not, the input audio data will be pushed into the FIFO buffer and popped once it is needed. The block will also need to convert audio serial data into 16-bit digital audio.
* Control Plane Requirements:
  + Serial data is transmitted in two’s complement with the Most Significant Bit (MSB) first. Serial data must be latched into the receiver on the leading edge of the serial clock signal. See Section 3.1 of I2S Specification Sheet for more details.
  + Support audio input sample rates of 8 kilosamples/sec – 48 kilosamples/sec.
  + The buffer sizing will have 2 or 3 storage elements which means we need a pointer that is 4 bits.
  + Overflow a possibility because the FIFO buffer could be full when trying to input audio. In this case, the input will be ignored and the buffer will not change.
* Control and Status Interface Bit Descriptions:
  + Control Bits
    - ws: selects w­hat audio channel is being read. 0 = left channel, 1= right channel.
    - rst: clears the buffer, output, and read/write pointers
    - clk: master clock
    - sck: serial clock
    - bist: built in self-test
  + Status Bits
    - fifo\_overrun: The FIFO buffer is full and no more can be added to the buffer
    - din\_rtr: Output FIFO asserts ready to receive
    - din\_rts: Write Client asserts ready to send

**Micro-Architecture:**

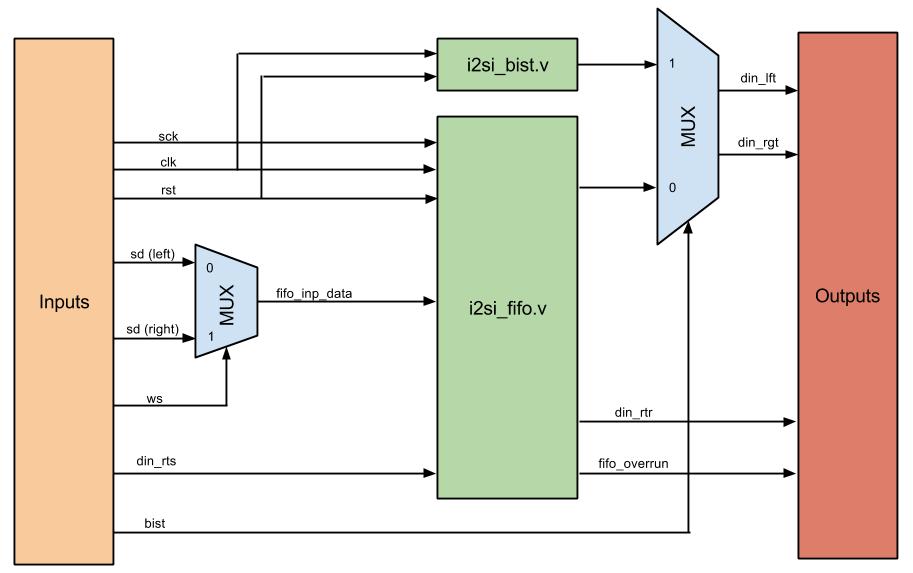
* Sub-blocks
  + i2si\_fifo.v

|  |  |  |  |
| --- | --- | --- | --- |
| Signal Name | Direction | Bits | Comment |
| clk | in | 1 | Master Clock |
| rst | in | 1 | Reset |
| fifo\_inp\_data | in | 16 | Input Data |
| fifo\_inp\_rts | in | 1 | Write Client Asserts Ready to Send |
| fifo\_inp\_rtr | out | 1 | Output FIFO Asserts Read to Receive |
| fifo\_out\_data | out | 16 | Output FIFO Asserts Ready to Send |
| fifo\_out\_rts | out | 1 | Output FIFO Asserts Ready to Send |
| fifo\_out\_rtr | in | 1 | Read Client Asserts Read to Receive |

* + i2si\_bist.v

|  |  |  |  |
| --- | --- | --- | --- |
| Signal Name | Direction | Bits | Comment |
| clk | in | 1 | Master Clock |
| rst | in | 1 | Reset |
| bist\_out\_data | out | 16 | Output BIST Data |

* Block Diagram



**Design:**

**Verification:**