**i2si.v Document – Zachary Nelson**

**Interfaces:**

|  |  |  |  |
| --- | --- | --- | --- |
| Signal Name | Direction | Bits | Comment |
| clk | in | 1 | Master Clock |
| rst | in | 1 | Reset |
| i2si\_sck | in | 1 | Digital Audio Bit Clock (max of 48kHz) |
| i2si\_ws | in | 1 | Word Select (Left/Right Audio Channel) |
| i2si\_sd | in | 1 | Digital Audio Serial Data |
| i2si\_rtr | in | 1 | Ready to Receive |
| i2si\_rts | out | 1 | Ready to Send |
| rf\_i2si\_en | in | 1 | i2s input is enabled |
| ro\_fifo\_overrrun | out | 1 | Input Audio FIFO Overrun |
| rf\_bist\_en | in | 1 | Built in Self-Test (BIST) |
| rf\_bist\_start\_val | in | 16 | Start Value |
| rf\_bist\_inc | in | 8 | Increment |
| rf\_bist\_up\_limit | in | 16 | Upper Limit |
| i2si\_data | out | 32 | Output Digital Audio |

**Functional Requirements:**

* Data Plane Requirements:
  + If the rf\_bist\_en control bit is true, the output of the block will be pre-defined such as a fixed sawtooth wave. This wave will be generated by the i2si\_bist\_gen.v module. If not, the input audio data will be pushed into the FIFO buffer and later popped. 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 2 bits.
  + Overflow is 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
    - i2si\_ws: selects w­hat audio channel is being read. 0 = left channel, 1= right channel.
    - rf\_i2si\_en: i2s input is enabled. Is idle when rf\_i2si\_en = 0 and active upon the first high-to-low transition of word select (ws) and rf\_i2si\_en = 1.
    - rf\_bist\_en: built in self-test
  + Status Bits
    - ro\_fifo\_overrun: The FIFO buffer is full and no more can be added to the buffer
    - i2si\_rtr: Output FIFO asserts ready to receive
    - i2si\_rts: Write Client asserts ready to send

**Micro-Architecture:**

* Sub-blocks
  + i2si\_bist\_gen.v

|  |  |  |  |
| --- | --- | --- | --- |
| Signal Name | Direction | Bits | Comment |
| clk | in | 1 | Master Clock |
| rst | in | 1 | Reset |
| i2si\_sck | in | 1 | Serial Clock |
| rf\_bist\_start\_val | in | 12 | Start Value |
| rf\_bist\_inc | in | 8 | Increment |
| rf\_bist\_up\_limit | in | 12 | Upper Limit |
| i2si\_xfc | out | 1 | Transfer Complete |
| i2si\_bist\_out\_data | out | 32 | Output Data |

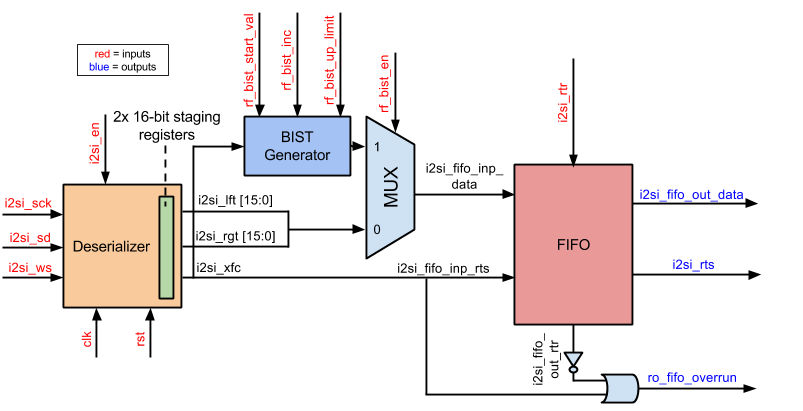
* + i2si\_fifo.v

|  |  |  |  |
| --- | --- | --- | --- |
| Signal Name | Direction | Bits | Comment |
| clk | in | 1 | Master Clock |
| rst | in | 1 | Reset |
| fifo\_inp\_data | in | 32 | 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 | 32 | Output Data |
| fifo\_out\_rts | out | 1 | Output FIFO Asserts Ready to Send |
| fifo\_out\_rtr | in | 1 | Read Client Asserts Read to Receive |

* + i2si\_deserializer.v

|  |  |  |  |
| --- | --- | --- | --- |
| Signal Name | Direction | Bits | Comment |
| clk | in | 1 | Master Clock |
| rst | in | 1 | Reset |
| i2si\_sck | in | 1 | Digital Audio Bit Clock (max of 48kHz) |
| i2si\_sd | in | 1 | Digital Audio Serial Data |
| i2si\_ws | in | 1 | Word Select (Left/Right Audio Channel) |
| rf\_i2si\_en | in | 1 | i2s input is enabled |
| i2si\_lft | out | 16 | Left Audio |
| i2si\_rgt | out | 16 | Right Audio |
| i2si\_xfc | out | 1 | Read Data Transfer Complete |

* Block Diagram



**Design:**

* As-Built Design Features
* Number of Gates and Flip-Flops
* How fast can it run?

**Verification:**

* Testbench
* Test Plan
  + The first thing that needs to be created for a test is a stream of bits that represents an I2S audio signal. More specifically, data for the 3 I2S inputs will need to be created.
  + The block needs to be tested when the i2si\_en bit is true and false. If the bit is false the I2S block should not input any data.
  + The block needs to be tested when the BIST is enabled. In this case only the predefined waveform will be sent to the filter module.
  + The block needs to be tested when the FIFO overflow and underflows. The block should be able to handle these situations without any problems.