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

**Interfaces:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Signal Name** | **Direction** | **Bits** | **Comment** |
| clk | in | 1 | Master Clock |
| rst\_n | in | 1 | Reset |
| sck\_transition | in | 1 | Serial clock transition |
| filt\_i2so\_lft | in | 16 | Left Parallel Digital Audio |
| filt\_i2so\_rgt | in | 16 | Right Parallel Digital Audio |
| filt\_i2so\_rtr | out | 1 | Ready to Receive |
| filt\_i2so\_rts | in | 1 | Ready to Send |
| i2so\_sck | out | 1 | Digital Audio Bit Clock (max of 48kHz) |
| i2so\_ws | out | 1 | Word Select (Left/Right Audio Channel) |
| i2so\_sd | out | 1 | Digital Audio Serial Data |
| ro\_fifo\_underrun | out | 1 | Output Audio FIFO Underrun |
| ro\_fifo\_overrun | out | 1 | Output Audio FIFO Overrun |

**Functional Requirements:**

* Data Plane Requirements:
  + Uses FIFO. Audio from filter is going to inputted into the block and stored in FIFO. The data will then be popped from the FIFO to convert audio serial data into 16-bit digital audio to serial data.
* 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.
  + The buffer sizing will have 16 storage elements which means we need a pointer that is 4 bits.
  + Overflow/Underflow is a possibility because of the FIFO buffer. In either case, the input or output will be ignored and the buffer will not change.
* Control and Status Interface Bit Descriptions:
  + Control Bits
    - i2so\_ws: selects what audio channel is being read. 0 = left channel, 1 = right channel
  + Status Bits
    - ro\_fifo\_overrun: The FIFO buffer is full and no more audio can be added
    - ro\_fifo\_underrun: The FIFO buffer is empty so there is no audio to output
    - i2so\_rtr: Read client asserts ready to receive
    - i2so\_rts: Write client asserts ready to send

**Micro-Architecture:**

* Sub-blocks
  + i2so\_fifo.v

|  |  |  |  |
| --- | --- | --- | --- |
| **Signal Name** | **Direction** | **Bits** | **Comment** |
| clk | in | 1 | Master Clock |
| rst | in | 1 | Reset |
| i2so\_fifo\_inp\_data | out | 16 | Output Data |
| i2so\_fifo\_inp\_rts | out | 1 | Write Client Asserts Ready to Send |
| i2so\_fifo\_inp\_rtr | in | 1 | Input FIFO Asserts Read to Receive |
| i2so\_fifo\_out\_data | in | 16 | Input Data |
| i2so\_fifo\_out\_rts | in | 1 | Input FIFO Asserts Ready to Send |
| i2so\_fifo\_out\_rtr | out | 1 | Read Client Asserts Read to Receive |

* + i2so\_serializer.v

|  |  |  |  |
| --- | --- | --- | --- |
| **Signal Name** | **Direction** | **Bits** | **Comment** |
| clk | in | 1 | Master Clock |
| rst\_n | in | 1 | Reset |
| i2so\_sck | out | 1 | Digital Audio Bit Clock (max of 48kHz) |
| i2so\_sd | out | 1 | Digital Audio Serial Data |
| i2so\_ws | out | 1 | Word Select (Left/Right Audio Channel) |
| i2so\_en | in | 1 | i2s output is enabled |
| filt\_i2so\_lft | in | 16 | Left Audio |
| filt\_i2so\_rgt | in | 16 | Right Audio |
| filt\_i2so\_rts | in | 1 | Ready to send |
| filt\_i2so\_rtr | out | 1 | Ready to read |

* Block Diagram



**Design:**

**Verification:**

* Testbench:
* Test Plan:
  + Create sample 16-bit audio as a stream of bits to see if the output is serialized as i2s data format.
  + The block needs to be tested when the FIFO overflow and underflows. The block should be able to handle these situations without any problems.