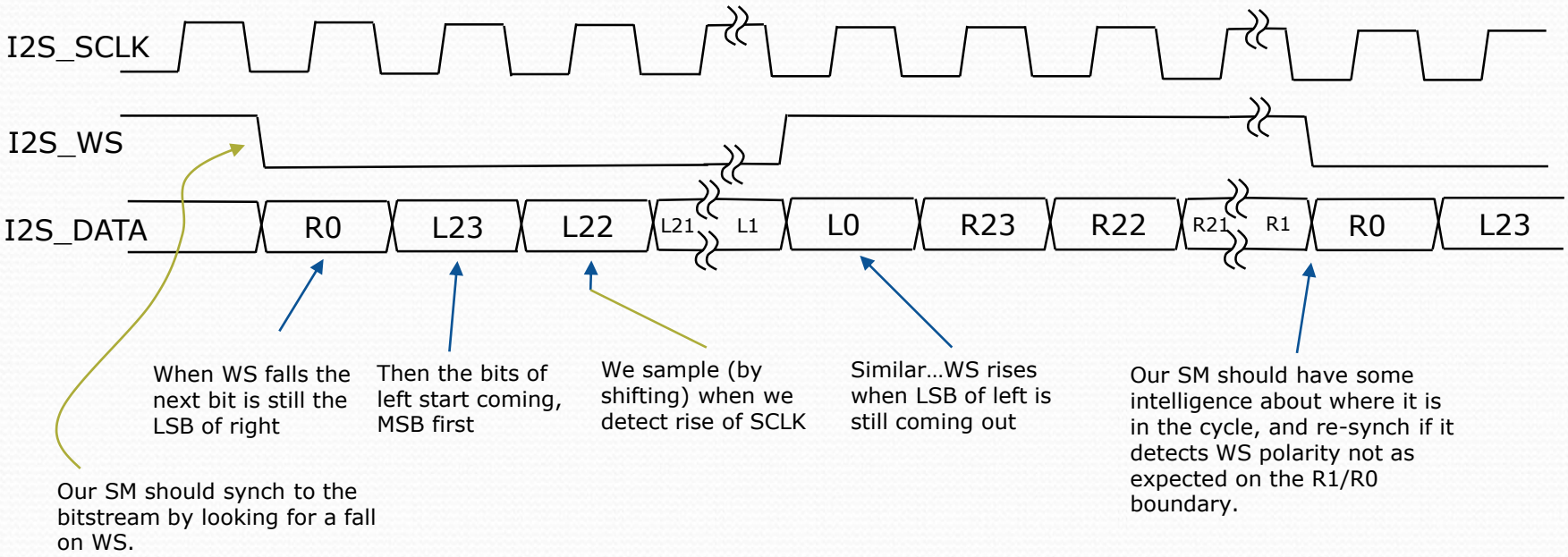
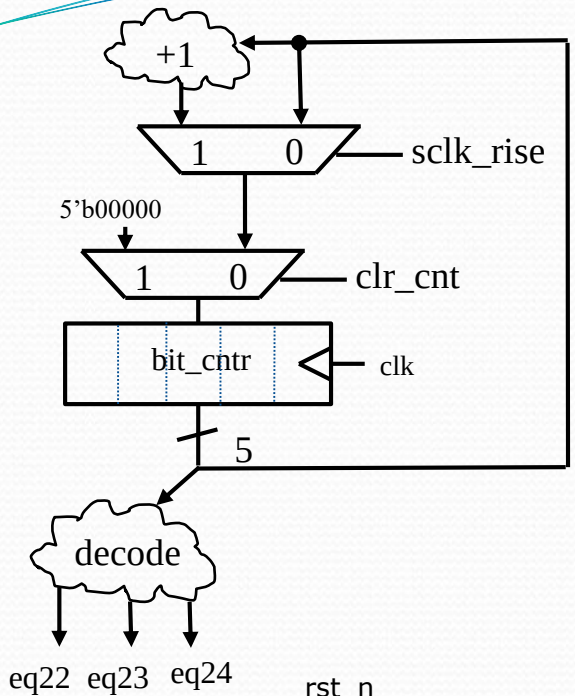
Exercise 13: I2S_Slave (HW4 Prob 5).

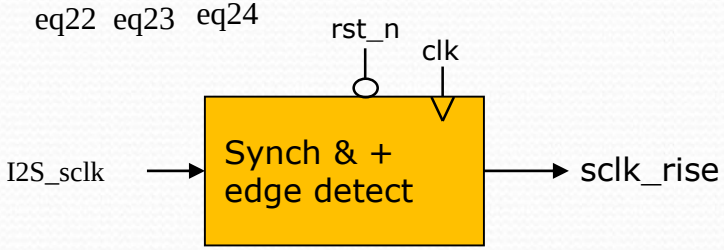
- I2S is the protocol the RN-52 BT Module will use to output audio. It is quite similar to SPI in many ways, except this time we are the slave and the RN-52 is the master.

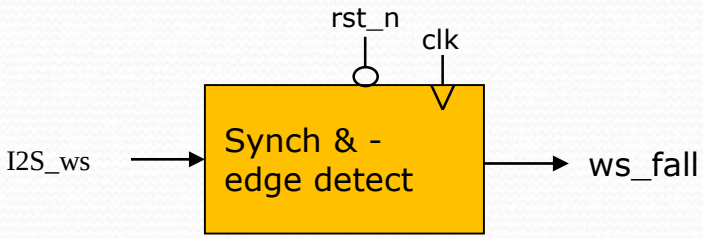


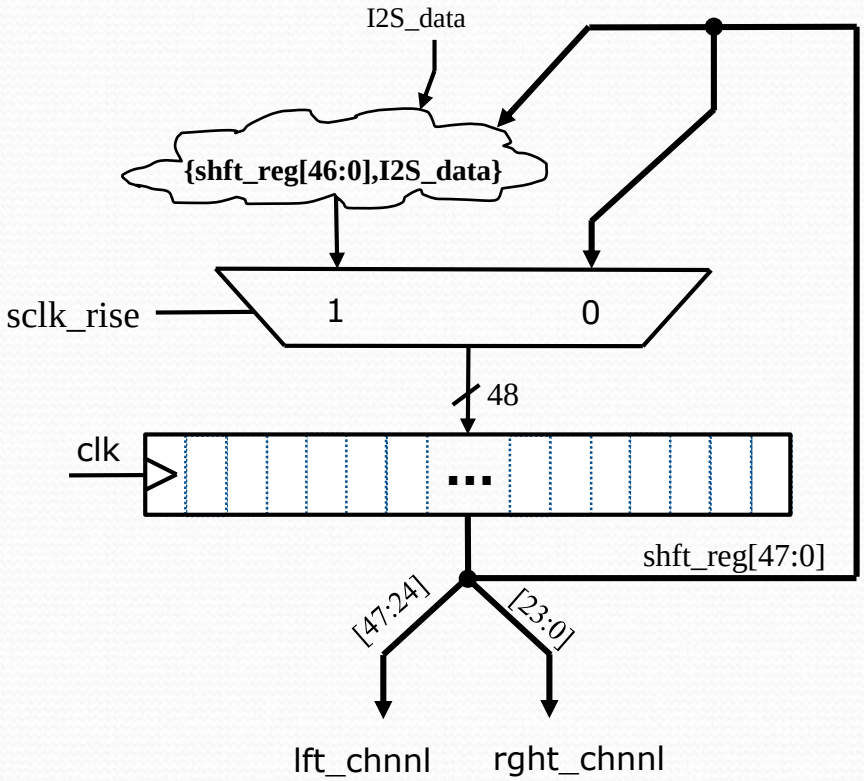
- The protocol is 24-bit left/right audio data at 44.1kHz…**SCLK** freq is: 24*2*44.1k = 2.11MHz
- The **WS** line is generally low during left data, and high during right…except it leads by 1 bit.
- We would sample the data by shifting our shift register(s) when we detect **SCLK** rise.
- We initially synched to the stream by looking for the fall of **WS** then ignoring the next **SCLK** rise, and then start shifting in left data on **SCLK** rise.
- When we know we are at ***R1*** we should check that **WS** is still high. When we know we are at ***R0*** we should check that **SCLK** is low. If this is not true we should re-synch. Otherwise we would go right back to sampling left data after ***R0***.

Exercise 13: Possible Architecture of I2S Slave









Also need a SM to control it, and assert when the data {**lft_chnnl**,**rght_chnnl}** is **vld**. **vld** can be a single system clock cycle wide. **lft_chnnl** and **rght_chnnl** will get buffered (captured) in registers) elsewhere.

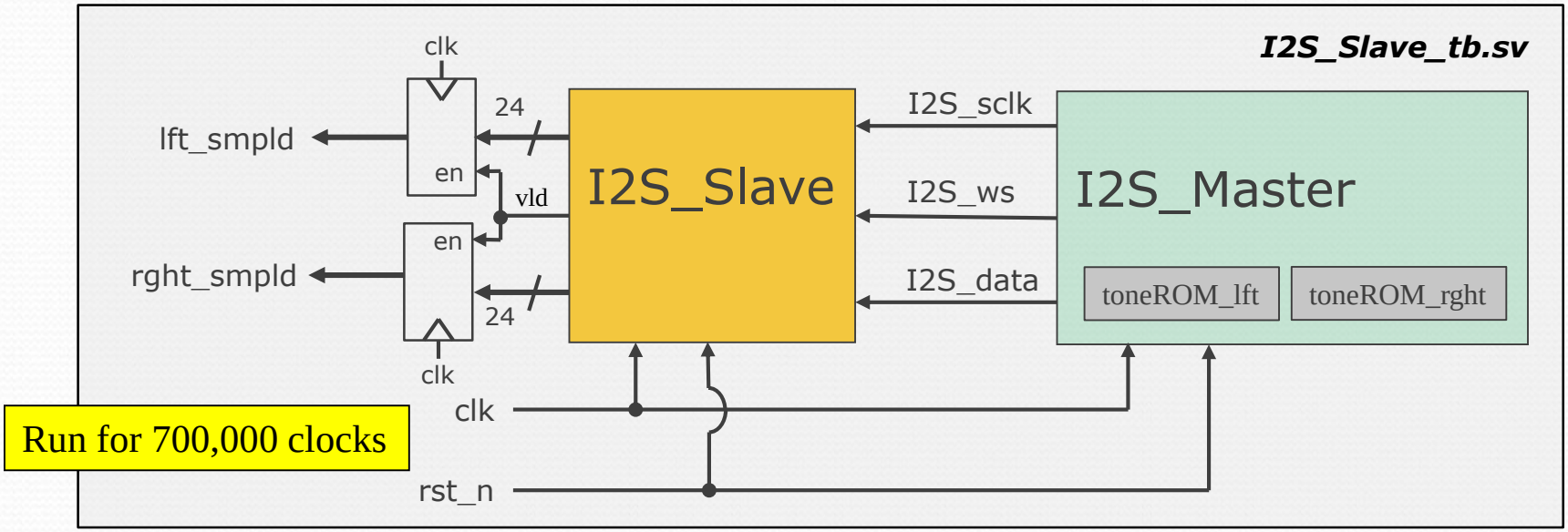
Exercise 13: I2S_Slave (HW4 Prob 5).

- Implement a module called **I2S_Slave.sv** with the following interface:

Signal:	Dir:	Description:
clk,rst_n	In	You know what these are
I2S_sclk	In	This is the SCLK signal of the I2S master. You should shift your shift register when you detect a rising edge on this signal.
I2S_ws	In	Word Select signal from I2S master. Leads bitstream by one bit. Low means left, high means right data
I2S_data	In	Serial data from I2S master. Should be sampled (shifted) into the LSB of a 48-bit shift register
lft_chnnl[23:0]	Out	Parallel 24-bit representation of left audio channel data
rght_chnnl[23:0]	Out	Parallel 24-bit representation of right audio channel data
vld	Out	Only has to be valid for 1 system clock cycle. **lft_chnnl** and **rght_chnnl** will be sampled (buffered) in the EQ engine based on this **vld** signal.

Exercise 13: Testing I2S_Slave

- A module called **I2S_master.sv** is provided on the Canvas page. Use this to help you build a test bench for **I2S_Slave.sv**

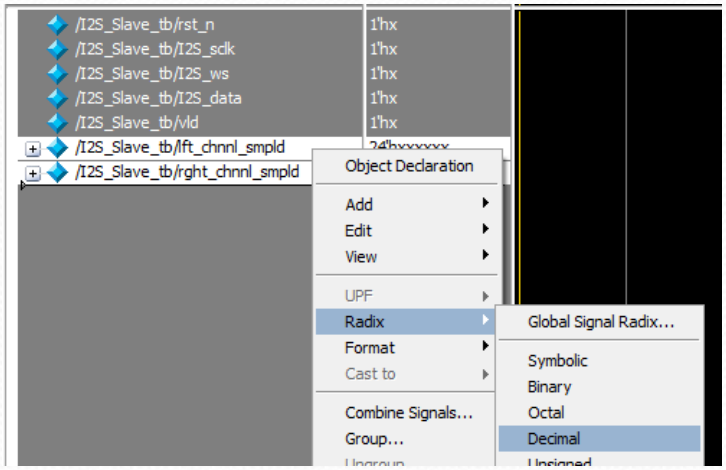


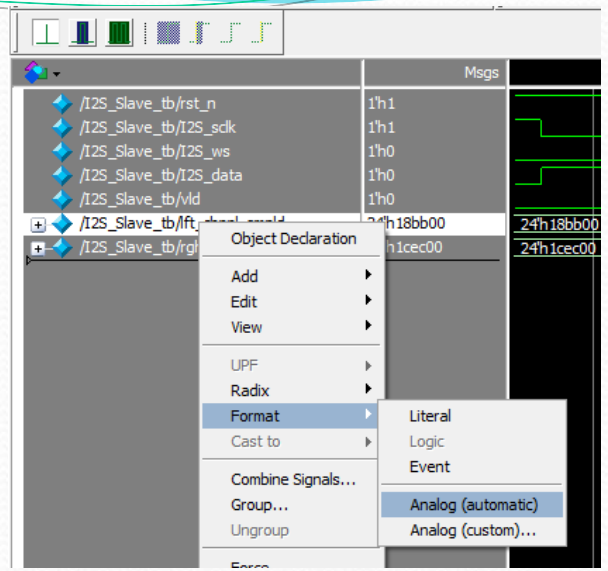
- Build a testbench as you see above. You should have a couple of 24-bit registers sampling the left and right outputs of your slave based on **vld**. One can plot signal in the modelSim viewer as analog. If you have a couple of sinusoids for the smpld values you have it right.

- **Turn in**: **I2S_Slave.sv**, **I2S_Slave_tb.sv**, and a **screen shot** showing it worked.

Exercise 13: Testing I2S_Slave

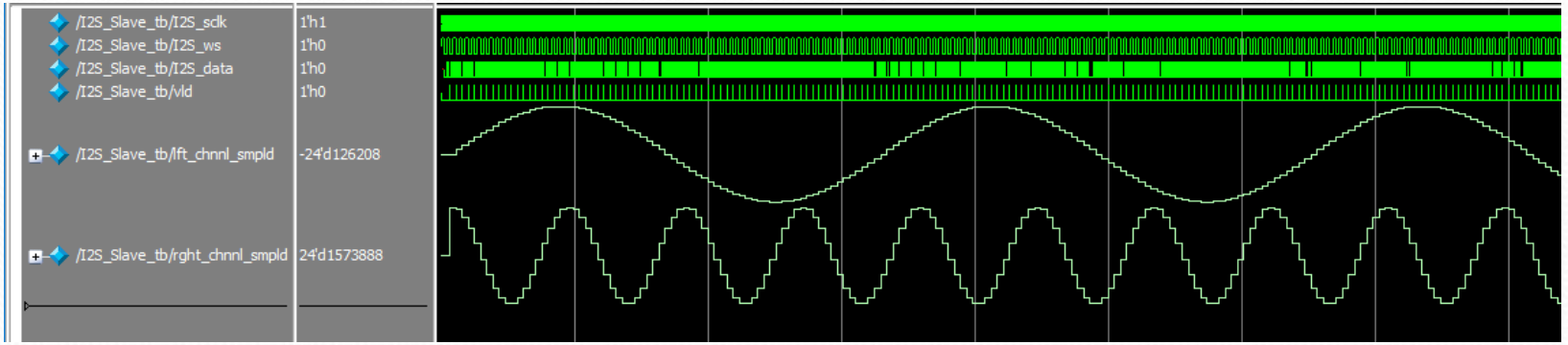
Displaying Signal As Analog





Change Radix to Decimal

Change Format to Analog with Auto Scaling



If your I2S_Slave is working properly then left and right channels should be sinusoidal right being a higher freq than left.

Submit whatever you have done to dropbox. Rest is due as part of HW4.