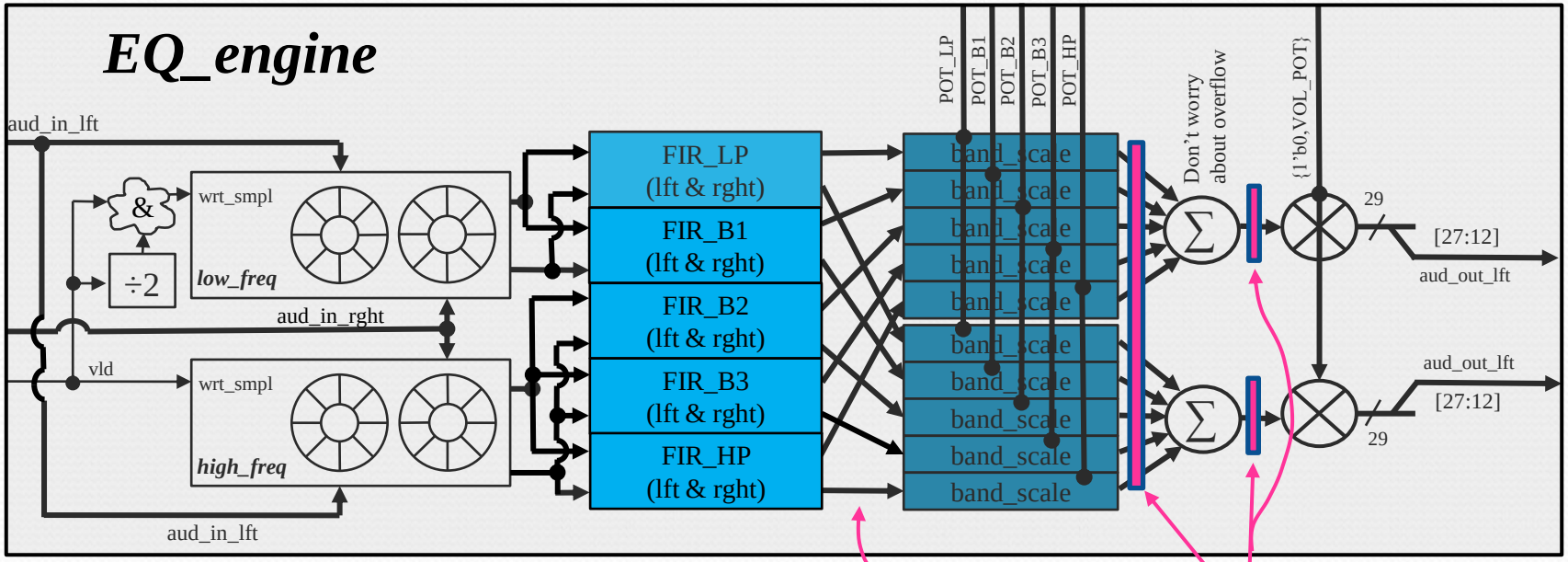
Exercise 23: Remaining Tasks

- **Remaining Design Tasks:**
  - **EQ_Engine**

- **Full Chip Integration and Testing:**
  - Download all provided files and get **Equalizer_tb.sv** & all children compiling.
  - Flush out **Equalizer_tb.sv** with tests.
    - Test all functions from toplevel TB *(objective is to ensure you are ready to pass our tests)*
    - Use good self-checking practices
    - Use tasks, make code readable and extensible

- **Full Chip Synthesis & Post Synth Sim:**
  - Develop & test synthesis script
  - Meet synthesis constraints
  - Pass a simple fullchip test using post synthesis simulation

Exercise 23: Remaining Tasks (EQ_engine)



Recommended flop insert points

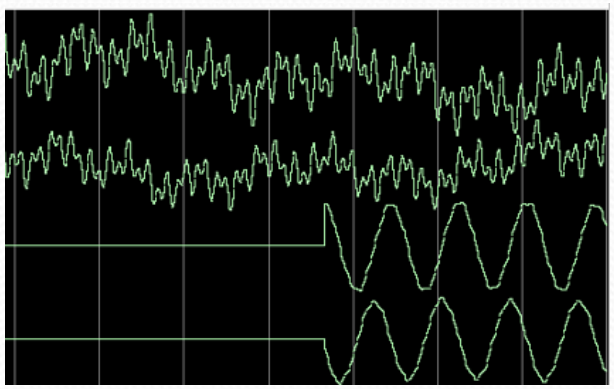
Why not here?

In the final summation of the bands it is possible there would be overflow. Don't sweat it, just keep everything at 16-bits. If there is overflow it just means you young whipper snappers are playing your rock-n-roll music too loud and should turn it down. However, I ask that you always keep overflow in mind in case you are working on a mission critical application in the future.

There is a heck of a lot of math going on here kids. Is Synopsys going to be able to fit all this math in a single cycle at a 3ns clock period? You need to insert some flops in this path to pipeline it and ease the timing constraints. Start without flops first and get it functionally working. Add the pipelining flops later, the latency of the flops affects nothing *(they can be free running flops).*

After scaling by volume you should have a signed 29-bit product. Just use bits 27:12 as the result for each channel. Don't worry about any saturation. Again, you are playing your music too loud if that happens.

Exercise 23: Remaining Tasks (Fullchip testing)



- The idea presented in Ex22 of filtering a superimposed set of sinusoids is a good method of testing full functionality
- Test each band. Can you make your own input files *(tone_hex_lft/right.txt)?*
- How can you self check these results?
- Think about zero crossings and amplitudes
- Can you test the full path? Can you make an “inverse PDM”?
- What other functionality do you need to check?

Exercise 23: Remaining Tasks (Synthesis)

- You need to develop a synthesis script and synthesize everything from **Equalizer.sv** on down.

- The synthesis constraints to use are given near the end of the ProjectSpec

- You need to make max & min delay timings
  - How are you going to solve that speed path?
  - Add a set of pipelining flops in the offending block.

- You need to produce an **Equalizer.vg** output and be able to perform a simple post synthesis simulation of it.
  - It should be an abbreviated test. Does not have to show full functionality of the FIR filters (this would take too long).
  - Show I2S working and amp enabled will be enough.