The purpose of this assignment is to familiarize you with the APB and the hardware/software interface. You will design a hardware module that performs the scaling task in your audio decoder pipeline. This hardware module will be accessed by a software subsystem that is responsible for interfacing to the hardware. The hardware module will be physically interfaced to the processor via the APB. The resulting hardware/software scaling subsystem will be exercised by your software from assignment #1.

The hardware/software scaling subsystem needs to support the following functions:

- `void set_scale_factor(uint8 scale_factor); /* set the scale factor */`
- `int16 scale(int8 sample); /* scale input sample, returning scaled sample */`
- `void scale_vector(int8 samples[], int16 scaled_samples[], uint16 sample_count);`
- `uint8 read_scale_factor(void); /* return the current scale factor */`

The `set_scale_factor()` software routine will write the value of `scale_factor` to a fixed address on the APB. The `scale()` routine will write the value of `sample` to a distinct (different) address on the APB and then read the scaled value from an APB address. The `scale_vector()` routine performs scaling on the input vector `samples`, returning `scaled_samples`, with `sample_count` specifying the vector length. Finally, the `read_scale_factor()` routine will read the current scale factor from an APB address. Since all APB transactions are 32 bits, set all unused bits to zero.

Test your scaling subsystem using the test data from `input01.h` and `input02.h`.

For the hardware version, you are required to design a Verilog testbench that mimics the processor side of the APB. It should exercise your hardware with the same functional requirements as the software test, although you may decrease the number of tests performed if you wish (i.e., smaller number of samples). Be sure and exercise your hardware module: 1) using the testbench, and 2) on the actual hardware.

Time the operation of both the software and hardware versions with text output disabled.

Turn in commented code (C and Verilog), timing results (including a description of how you performed the timing), output data, and simulation screenshots. Include enough description so we can tell what is going on.