ELEC2221 D1 Report Form 2023/24

Yike Zhang	yz50u22@soton.ac.uk

Abstract Summarise what you achieved. This is a summary of the report, not an introduction. 100 word limit. (2 marks)

The D1 design exercise focus on developing a 16-bit FIR function work as a low-pass filter with a cutoff frequency set at 11025Hz. It encompasses both digital design and signal processing components. A completed FIR module can process input audio signals and generate desired outcomes. Then the design leverages parameterization techniques to augment the bit depth of the FIR filter. Subsequently, the extended bit-depth FIR filter is instantiated on the DE1-soc hardware platform. Which facilitates a close examination of the practical characteristics of the output signal.

Design and Verification How did you implement the basic design? What modifications did you make to the code that you were given? How did you use the design software (ModelSim, Quartus and the online checker)? (3 marks)

In the existing SystemVerilog code FIR.sv, the initial filter coefficients, shift register, and accumulator register are already defined. The first task is to implement an address counter logic, which is a synchronous counter. According to the ASM chart provided in the D1 instruction, when the count signal is true, it begins counting up to the 16th bit. Following this, a posedge-triggered reset logic is implemented. Finally, the state machine is completed based on the ASM chart.

Subsequently, Modelsim can be applied. After resolving compilation errors in ModelSim, such as common syntax and symbol issues, the next step involves validating the waveforms generated. The initial attention should be given to ensuring that the clock cycles correspond appropriately to the states of the state machine. Ideally, states “saving”, “loading” and “waiting” each require a single clock cycle. However, the “processing” state, involving waiting for the address to count up from the 1st to the 16th position (i.e., 0-15), necessitates 16 clock cycles, so the total clock circles for the whole system is 19. It’s noteworthy that a slight delay between FIR_in and FIR_out is observed, attributed to a phase difference between input and output signals. Once all these tasks are completed, an online code checker can be employed to scrutinize the code. This tool is beneficial as it highlights warnings that might be overlooked. Next, in the D1.sv file, the FIR module is connected to the FPGA interface. It is preferable to filter only one channel of the audio to ensure a clear comparison between the audio before and after filtering. After completing this step, the entire code is compiled using Quartus, and the synthesized FIR module is inspected using the Netlist Viewer. Finally, the file is downloaded onto the FPGA development board for testing.

It’s noteworthy that when compiling with Quartus, an error may arise, indicating multiple values assigned to the same interface, such as “wiritting”. In such cases, a solution involves creating a new wire assignment, for instance, “wiritting_FIR”. These assignments can then be connected using the AND logic operator (&), and the newly created assignment should be instantiated. This approach effectively resolves the issue.

Extended Design What additional functionality did you implement? How did you verify and test this extended work? (3 marks)

The process of parameterizing Finite Impulse Response (FIR) filters is a pivotal element in their design, facilitating the incorporation of a variable parameter N within the code. This approach enables customization based on specific requirements, such as setting N=16 for a 16-bit FIR filter or N=24 for a 24-bit variant. The Verilog operator $clog2(N) is utilized to determine the upper boundary for adress.

For the extension part, MATLAB is employed to regenerate the filter coefficients for the implementation of a notch filter based on the FIR. By applying the plotting function of MATLAB, a frequency response diagram can be generated with the new coefficients. A significant dip in the frequency response at stop frequency can be observed, indicating strong attenuation at this frequency. Frequencies outside the notch ill be passed with minimal attenuation.

ModelSim is utilized to monitor the internal states and clock cycles of the system. The parameterization influences the output waveforms in ModelSim, transforming the FIR output from a stepped pattern into a uniform square wave.

When deploying the code onto FPGA, assessing the filtering effects through auditory means alone can be challenging. To overcome this, connecting the FPGA's audio output interface to an oscilloscope is recommended. This connection provides a visual representation of the alterations in the output signal, so a under-modulated output signal can be observed inside the notch filter. By checking the waveform displayed on oscilloscope screen, it is relatively easier to determine the effectiveness of the notch filter compared to listening the audio through headphones.

Reflection What did you learn from this exercise? What would you do differently if you were to do the exercise again, and why? (2 marks)

In this design exercise, I delved into the workings of FIR, enhancing my understanding of SystemVerilog code writing, instantiation, and debugging. However, it also brought to light some challenges. My grasp of SystemVerilog remains incomplete; for instance, the use of "unique case" triggered a warning in the online code checker ('No conditions are true in case statement'). Though such warnings have minimal impact on system output, they revealed gaps in my comprehension. Additionally, my proficiency in MATLAB proved lacking; neglecting the "low" parameter in "fir1(N, normalizedFc, "low")", whcich should be change to "stop", resulted in generating coefficients consistently for a low-pass filter, leading to time wastage.

Reflecting on this experience, if faced with a similar design exercise again, I would allocate more time for preparation, familiarize myself with tools in advance, read documentation thoroughly, and acquire additional relevant knowledge to complete each exercise more confidently.