EECS 443 Final Project
Audio Recorder

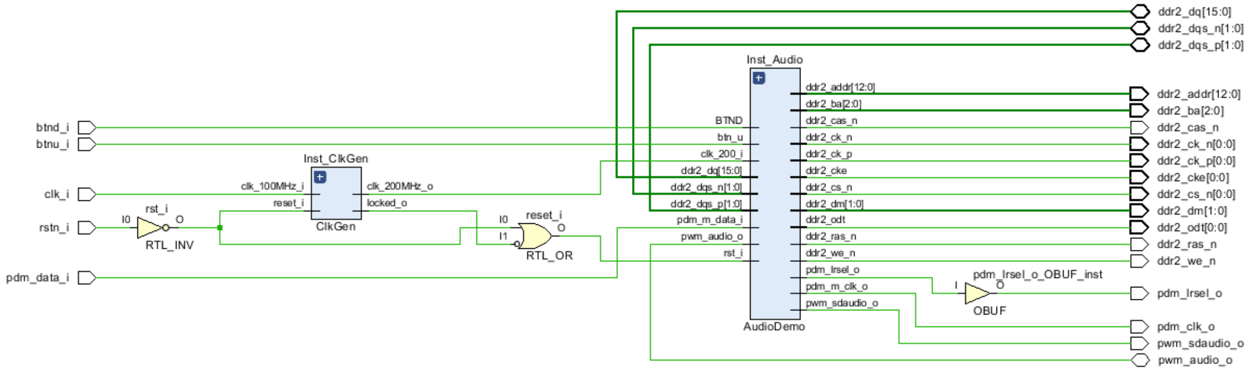
Ronald Heminway and Preston Wehrman

May 4th, 2023

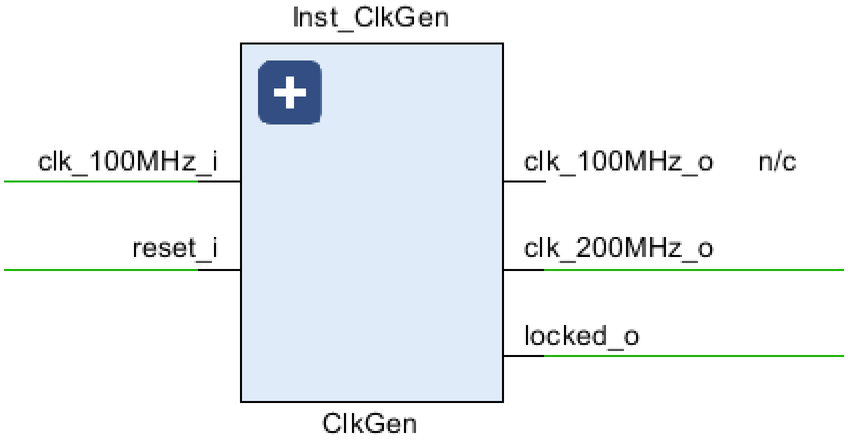
Detailed description of the project:

Our audio recorder is implemented on our NEXYS A7 board and consists of a few key components: a microphone input, a button input, a memory buffer, and a headphone output. When the btnu button is pressed, the recording process begins. Here the microphone input will begin to capture the audio data, which is then stored in the DDR memory. The recording process will continue for approximately 5 seconds. Once the recording is complete, the user can press the btnd button to trigger the playback process. Here the audio data we stored in the memory is then sent to the headphone output, where the user can listen to the recorded audio.

a. Hardware architecture with block diagrams

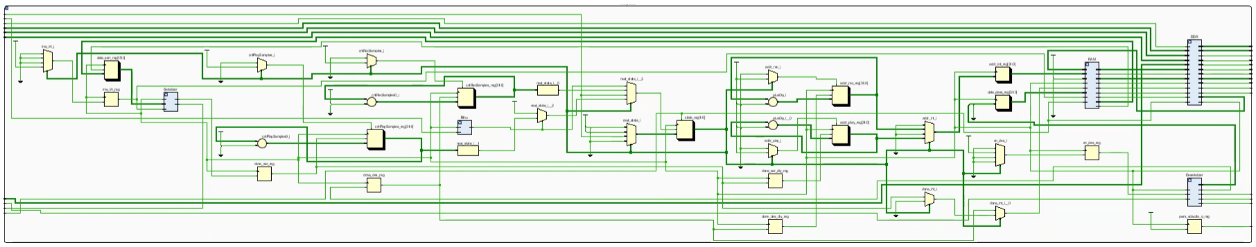


Full Architecture

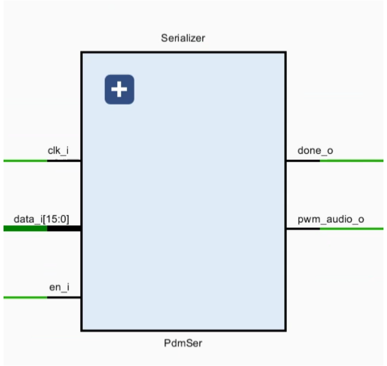


ClkGen: Generates a 200MHz signal

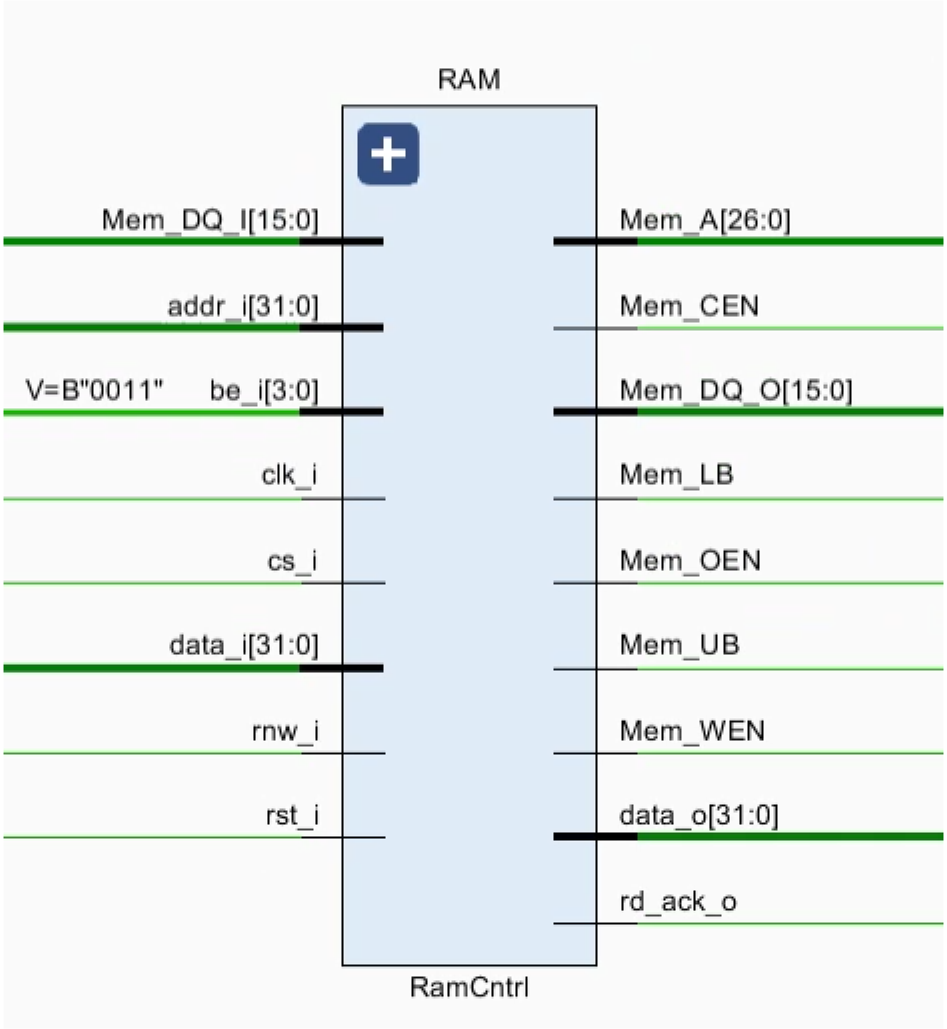
b. Behavior of individual components



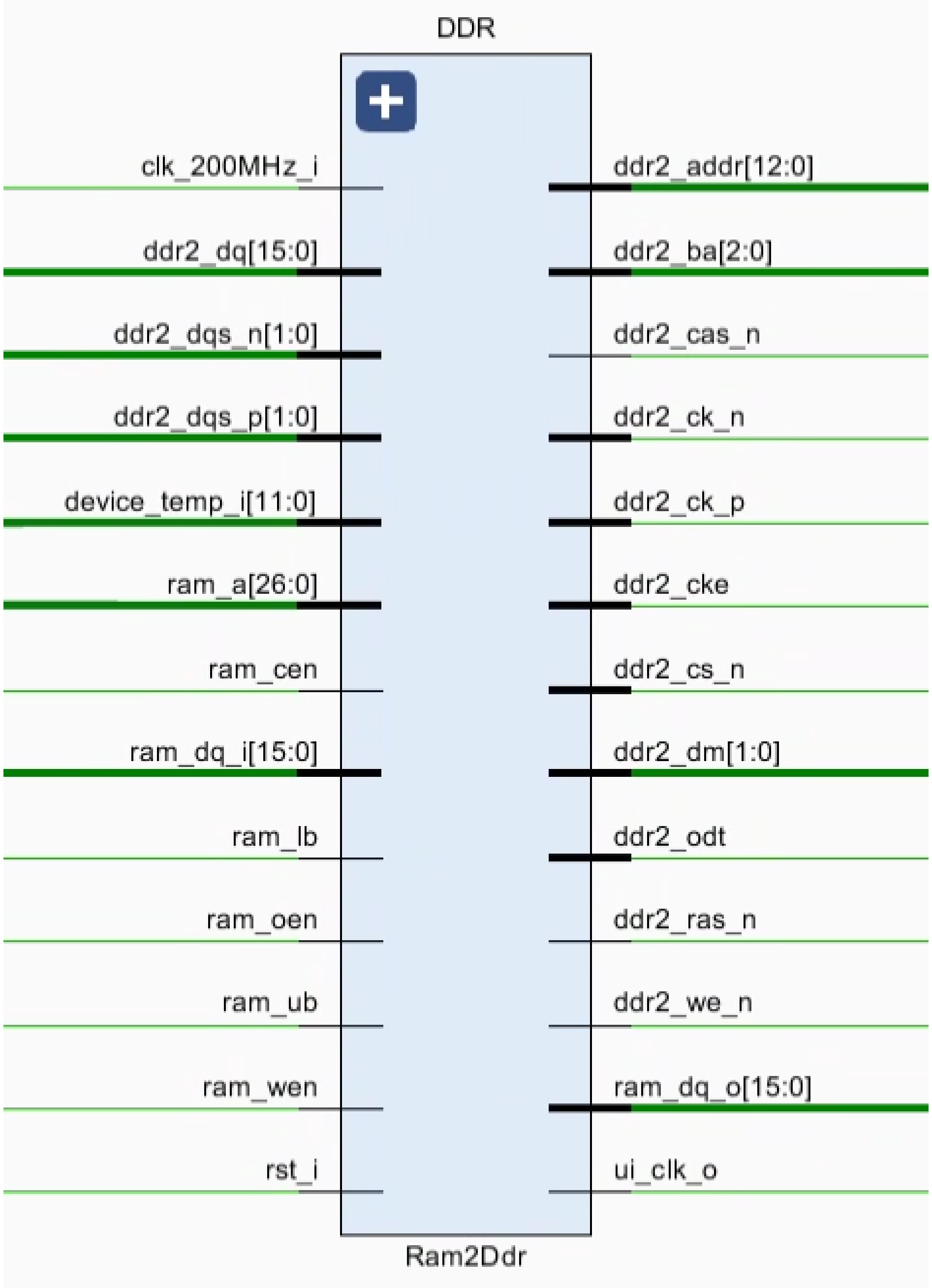
Audio Handler



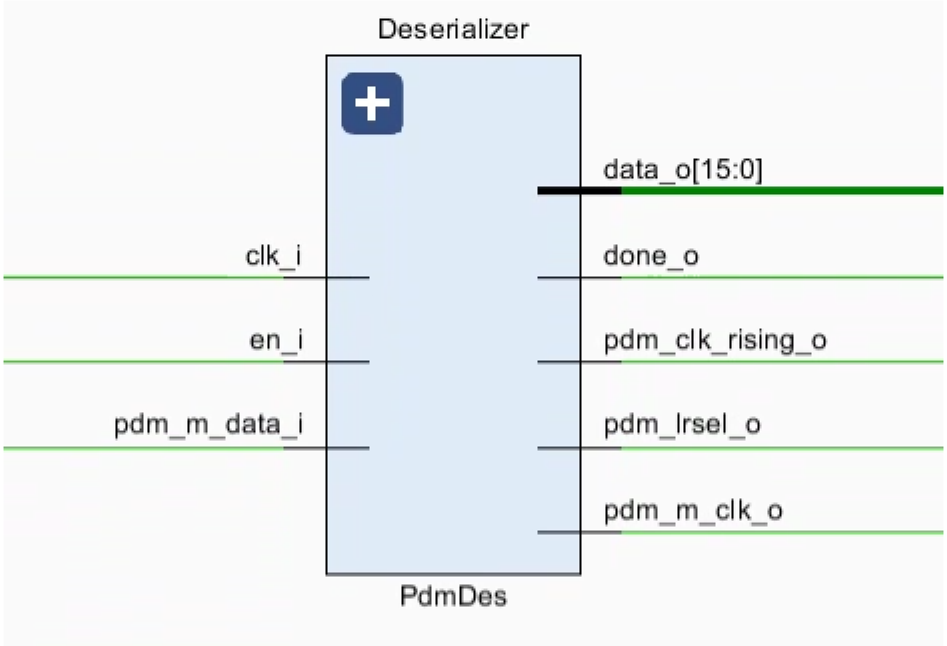
serializer: This module takes in input data and serializes it into a PDM data stream. It uses a clock divider to generate an internal PDM clock signal and count the number of sampled bits. The serializer is implemented using a shift register that temporarily stores the input data and shifts it out bit-by-bit on each rising edge of the PDM clock. The module also includes an enable signal to start and stop the serialization process and a done signal to indicate when the serialization is complete. The serialized PDM data stream is output through an inout port that can be connected to an external PWM (Pulse Width Modulation) controller.



RamCntr: implements the state machine to control the basic read and write procedures of a RAM memory.

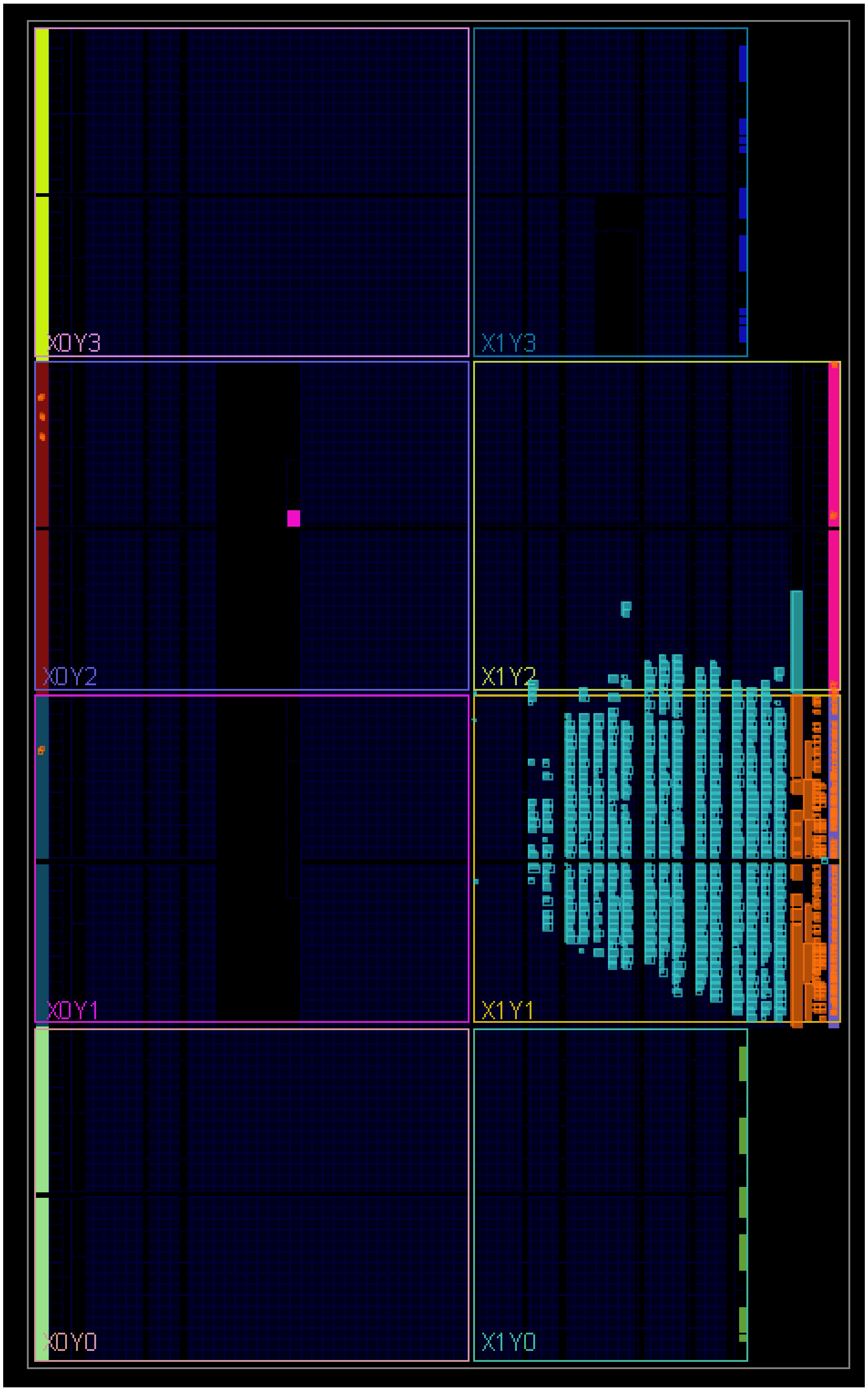


DDR: implements a simple Static RAM to DDR2 interface converter designed to be used with Digilent Nexys4-DDR board



Deserializer: takes in PDM (pulse-density modulation) microphone data and generates a clock signal for the microphone. The module deserializes the PDM data into 16-bit audio samples when recording is enabled. Additionally, the module generates a pdm_clk_rising_o signal to output audio data on the screen.

Implementation results



5. Discussion of results and comparison with expected behavior

After we completed the implementation of the audio recorder in VHDL, we conducted thorough testing to ensure that the device was functioning as expected. We are pleased to report that the audio recorder is working as intended. The recording process begins when the btnu button is pressed and ends after approximately 5 seconds. Following this we are able to successfully play back the recorded audio through the headphone jack by pressing the btnd button.

Work distribution among members

Ron and Preston were both instrumental in the development of the audio recorder. Ron focused on the audio interface module, working to ensure that the device was properly capturing and processing the audio data from the microphone input. He also worked on the headphone output, ensuring that the recorded audio could be played back in high quality. Meanwhile, Preston focused on the memory module, specifically figuring out how to utilize the DDR memory to store and retrieve the recorded audio data. Overall, both Ron and Preston played key roles in developing this audio recorder, and their contributions were essential to the project's success.