Laboratory 4

(Due date: **005**: March 23rd, **006**: March 24th)

OBJECTIVES

- ✓ Implement a Digital System: Control Unit and Datapath.
- ✓ Learn about reading PDM-coded input audio and playback PDM-coded output audio.
- ✓ Learn interfacing with MEMS microphones (that generate PDM signals) and using BlockRAMs in FPGAs.

VHDL CODING

 \checkmark Refer to the <u>Tutorial: VHDL for FPGAs</u> for a tutorial and a list of examples.

FIRST ACTIVITY: AUDIO RETRIEVAL AND PLAYBACK CIRCUIT (100/100)

 MICROPHONE AND MONO AUDIO OUTPUT: Implement the following circuit that reads data from the ADMP421 MEMS microphone, stores data in memory and plays data back on a mono audio output.



MEMS MICROPHONE (ADMP421):

- ✓ SCLK: 1 3 MHz. We use 1.2 MHz.
- ✓ PDM_IN: PDM signal generated by the ADMP421.
- \checkmark LR: Left right control for stereo mode. We use LR = 0 (Data is captured on CLK rising edge).

• MONO AUDIO OUTPUT:

- ✓ The on-board audio jack of the board (Nexys-A7-50T/A7-100T, Nexys 4-DDR) does not support stereo output. Thus, we only retrieve a mono audio input from the ADMP42 microphone (e.g. L/R = 0). Keep this in mind when testing (as only one speaker or earphone side will work).
- The on-board audio jack is driven by a low-pass filter with 12 KHz cut-off frequency. We need to feed a PDM signal (AUD_PWM: open-drain output) into the low-pass filter. Note that AUD_PWM is the output of a tri-state buffer:
 - If we want to transmit a '0', $AUD_PWM = '0'$.
 - If we want to transmit a '1', $AUD_PWM = 'Z'$.
- FSM_SCLK:
 - ✓ This circuit generates a free running clock (SCLK). It also generates a pulse on *z*R when a rising edge on SCLK occurs, and a pulse on *z*F when a falling edge on SCLK occurs. We want to use a frequency of 1.2 MHz. This requires SCLK *T*=84.
 - You can use the VHDL code fsm_sclk.vhd (along with my_genpulse_sclr.vhd) with the parameter COUNT_SCKLHP = SCLK_T/2=42.

- **MEMORY:** Implemented with the on-chip memory (BlockRAMs) inside the Artix-7 FPGA.
 - Data requested or to be written is available on the next clock cycle. Use the given VHDL code in_RAMgen.vhd. This is a 2D memory of nrowsxncols. Word length: 16 bits. We can use it as a 1D memory by making ncols=1. Use the following parameters nrows=(this depends on the board), ncols=1, INIT_VALUES="NO", FILE_IMG ="myinival.txt".
 For Nexys A7-100T/Nexys-4 DDR, use: nrows=512*512
 - For Nexys A7-50T, use nrows=512*256 (there is less memory in the FPGA inside this board)
- **FSM_MEM:** This is the main controller. This FSM embeds the counter RAM address. Make sure to use the correct ND:
 - \checkmark ND = 512 × 512 = 2¹⁸ for Nexys A7-100T/Nexys-4 DDR. Sequence duration: 2¹⁸ × 16 × 84 × 10ns = 3.52 seconds.
 - ✓ $ND = 512 \times 256 = 2^{17}$ for Nexys A7-50T.



PROCEDURE

- Vivado: Complete the following steps:
 - ✓ Create a new Vivado Project. Select the corresponding Artix-7 FPGA device (e.g.: the XC7A100T-1CSG324 FPGA device for the Nexys A7-100T board).
 - \checkmark Write the VHDL for the given circuit. Synthesize your circuit. (Run Synthesis).
 - Use the Structural description: Create (or re-use) a separate .vhd file for i) parallel access shift register, ii) inRAM_gen, iii) FSM_MEM, iv) FSM_SCLK, and v) top file (where you will interconnect them all).
 - Use the proper parameters and I/O connections. Note that FSM_SCLK is made out of two .vhd files.
 - ✓ Write the VHDL testbench (generate a 100 MHz input clock for your simulations).
 - To avoid long simulation times, use nrows=4x4, ncols=1 and COUNT_SCLKHP = SCLK_T/2 = 2 for simulation purposes (resynthesize your circuit).
 - PDM_IN: Generate the following serial input stream (4x4 16-bit words), where each bit is to be captured at the rising edge of SCLK: 1110 1100 1110 1011 0100 0111 0001 0000 1010 1100 1110 1101 1110 1010 0101 0001 and then just 1's. These bits are stored in memory (inRAM_gen) as 16-bit values: ECEB 4710 ACED EA51 FFFF ... FFFF.

- ✓ Perform <u>Behavioral Simulation</u> (Run Simulation \rightarrow Run Behavioral Simulation). **Demonstrate this to your TA.**
 - To help debug your circuit, add internal signals to the waveform (e.g.: FSM_MEM/state, RAM_address, SCLK, inRAM gen/in, inRAM gen/out, etc).
 - Run the simulation for 21 us (remember to use nrows=4x4, ncols=1, COUNT_SCLKHP = SCLK_T/2 = 2).
 - Verify that the hexadecimal values (see testbench directions) appear on inRAM_gen/out (memory output) during state S5 in FSM MEM.
 - Verify that the serial stream (see testbench directions) appears on dout (parallel access left shift register output)
- $\checkmark~$ I/O Assignment: Create the XDC file associated with your board.

Suggestion (Nexys A7-50T/A7-100T, Nexys 4/DDR):									
Board pin names	CLK100MHZ	CPU_RESET	BTNC	M_DATA	M_CLK	M_LRSEL	AUD_PWM	AUD_SD	LED0
Signal names in code	clock	resetn	start	PDM_IN	SCLK	LR	AUD_PWM	AUD_SD	ready_out

- ✓ Implement your design (Run Implementation).
- ✓ Generate and download the bitstream on the FPGA. Test the circuit. Demonstrate this to your TA.
 - Press start and record an audio sequence for 3.52 seconds (this is for the Nexys A7-100T/Nexys-4 DDR). After that, the audio sequence is played back (use a headphone/speaker).
- Submit (<u>as a .zip file</u>) the generated files: VHDL code, and VHDL testbench, and XDC file to Moodle (an assignment will be created). DO NOT submit the whole Vivado Project.

TA signature: _____

Date: _____