

Laboratory 3

(Due date : **002**: March 4th, **003**: March 5th)

OBJECTIVES

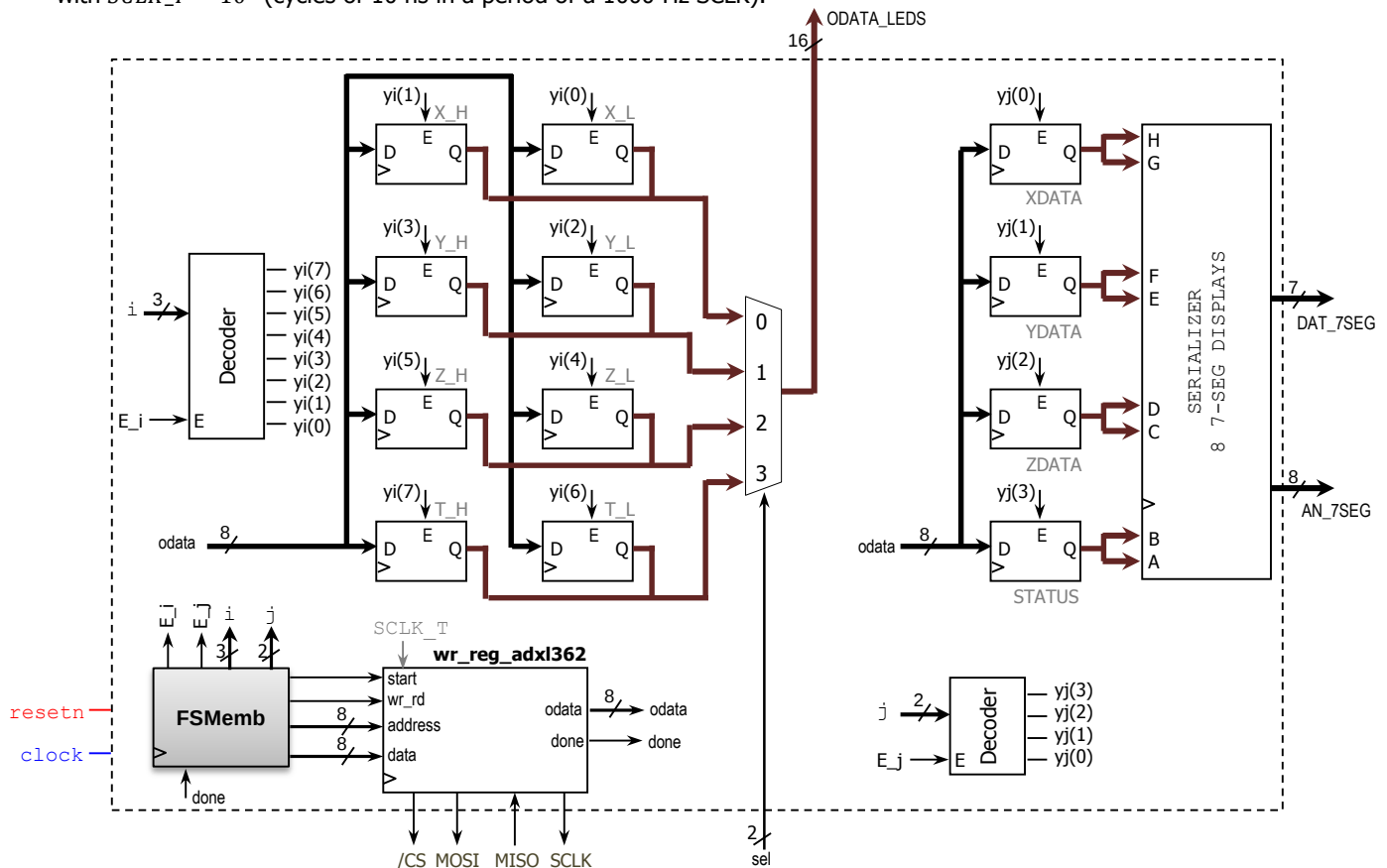
- ✓ Implement a Digital System: Control Unit and Datapath Unit
- ✓ Describe Algorithmic State Machine (ASM) charts in VHDL.
- ✓ Learn interfacing with SPI devices.

VHDL CODING

- ✓ Refer to the [Tutorial: VHDL for FPGAs](#) for parametric code for: Registers, busmux.

FIRST ACTIVITY: DESIGN OF AN ACCELEROMETER DATA RETRIEVER (100/100)

- **ACCELEROMETER ADXL362:** This 3-axis MEMS device communicates via a 4-wire SPI and operates as a SPI slave device. We read/write 8-bit data via a register-based interface. ADXL362 parameters (range, resolution, ODR are selectable):
 - ✓ Range: $\pm 2g$ (default at reset), $\pm 4g$, $\pm 8g$.
 - ✓ Resolution: 1mg/LSB (default at reset), 2 mg/LSB, 4 mg/LSB
 - ✓ Output data rate (ODR): 12.5 – 400 Hz. Default at reset: 100 Hz.
 - ✓ Output resolution: 12 bits. Representation: signed.
- **wr_reg_adxl362:** This circuit handles basic SPI communication with the ADXL362. The user provides address, data, and read/write. Then, a read/write transaction is executed via the SPI bus (data structure specified in the ADXL362 datasheet). Use the VHDL code wr_reg_axl362.vhd (use all the design .vhd files in [accelerom.zip](#) except accelerom.vhd) with $SCLK_T = 10^5$ (cycles of 10 ns in a period of a 1000 Hz SCLK).

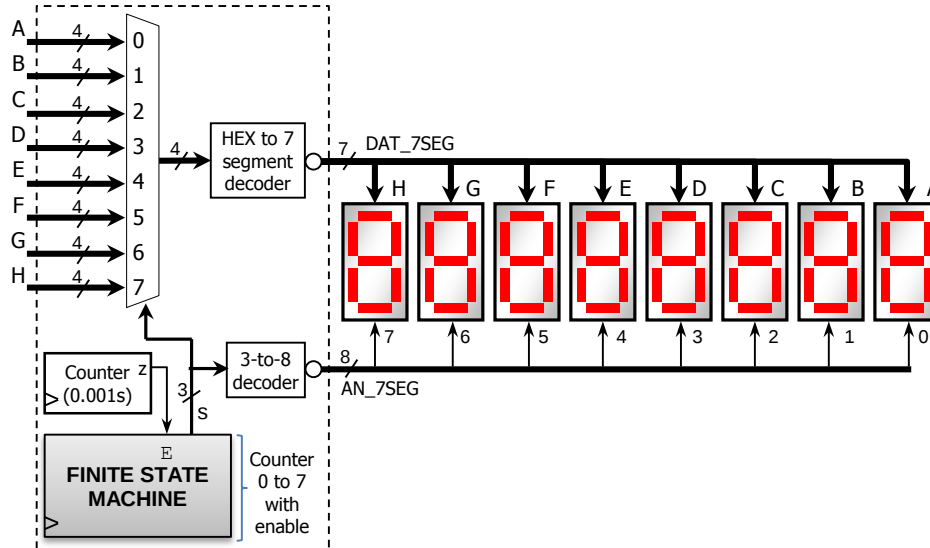


- **FSMemb:** FSM embedding counters i and j . It configures 2 ADXL362 registers ($0 \times 1F$, $0 \times 2D$), and then cyclically requests read from 12 8-bit ADXL362 registers containing accelerometer data and places retrieved data on 12 8-bit output registers.
- Data is organized into:
 - ✓ 4 16-bit measurements (X, Y, Z, Temperature). We display this on 16 LEDs (selectable by sel). Note that since data is 12-bit wide, the 4 MSBs are sign-extended.
 - ✓ 3 8-bit measurements (low precision X, Y, Z) and 8-bit Status: Shown (as hex) on 8 7-segment displays: $|X|Y|Z|ST|$

- This is the list of registers we deal with in this experiment, which is a basic operation mode. Refer to the ADXL362 datasheet for a full list of registers and operation modes.

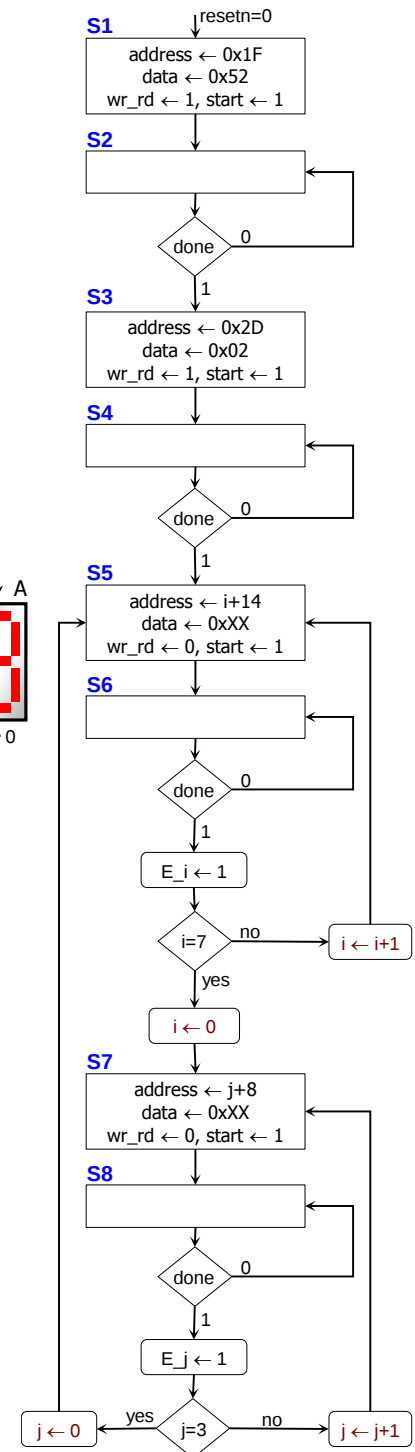
Reg. Address	Name	Reg. Address	Name
0x1F	SOFT_RESET	0x0E	XDATA_L
0x2D	POWER_CTL	0x0F	XDATA_H
		0x10	YDATA_L
		0x11	YDATA_H
0x08	XDATA	0x12	ZDATA_L
0x09	YDATA	0x13	ZDATA_H
0x0A	ZDATA	0x14	TEMP_L
0x0B	STATUS	0x15	TEMP_H

- Serializer:** Eight 7-segment displays. You can download the Serializer for four 7-segment displays available in [Tutorial: VHDL for FPGAs](#) and modify it for 8 displays.



VIVADO DESIGN FLOW FOR FPGAs – NEXYS A7-50T

- ✓ Create a new Vivado Project. Select: **XC7A50T-1CSG324 Artix-7 FPGA**.
- ✓ Write the VHDL code for the given circuit. Structural description is recommended.
- ✓ With a 100 MHz input clock, write the VHDL testbench.
 - Since SCLK=1000 Hz, it will take a long while to simulate. Only for simulation purposes, use SCLK_T=16. For simplicity, use MISO=1. Observe the SPI signals and verify that data on ODATA_LEDS and DAT_7SEG are correct. Use sel to select what to display on ODATA_LEDS.
- ✓ Perform Functional Simulation and Timing Simulation of your design. **Demonstrate this to your TA.**
- ✓ I/O Assignment: Create the XDC file.
 - Use SW1-SW0 for sel[1..0], CLK100MHZ for the input clock, CPU_RESET for resetn, LED15-LED0 for ODATA_LEDS[15..0], CA-CG for DAT_7SEG[6..0], AN7-AN0 for AN_7SEG[7..0], ACL_MISO for MISO, ACL_MOSI for MOSI, ACL_SCLK for SCLK, and ACL_CSN for /CS.
- ✓ Generate and download the bitstream on the FPGA (use SCLK_T=10⁵). Test the circuit. **Demonstrate this to your TA.**
 - 16 LEDs: Verify the 4 MSBs are effectively sign-extended bits.
 - Verify that the low precision data (XDATA, YDATA, ZDATA) on the 7-seg displays match the high precision X, Y, Z on ODATA_LEDS[11..4]; use sel to pick among different 12-bit measurements. Also, STATUS should be 0x41.
 - Feel free to tilt the axes of the FPGA Board to detect changes. For example: at rest, Z should be about 0xC28 (7-seg display: 0xC2). This corresponds to -984mg. When the board is face down, the sign of Z should be positive.
- Submit (as a .zip file) all the generated files: VHDL code files, VHDL testbench and XDC file to Moodle (an assignment will be created). DO NOT submit the whole Vivado Project.



TA signature: _____

Date: _____