

Final Project Description

The objective of the final project is to utilize multiple DSP kernels for a complete system that aims at performing vibration analysis of a target. The project will consist of several phases with the end design consisting of 3 stages:

1. Reading 3 axis acceleration values from an accelerometer/gyroscope IC (MPU6050) into an FPGA using SPI communication.
2. Extracting power spectral density of each acceleration stream by using filtering, windowing, and FFT DSP kernels.
3. Sending both the filtered time-series data and spectral energy data to the PC to be displayed in MATLAB using UART over USB communication.

In order to produce the final implementation, the process will be split up into several phases.

Phase 1: Simulated Data & Spectral Analysis in MATLAB

Phase one will consist of performing the entire spectral analysis and graphical display in MATLAB using simulated data. The simulated data will mimic the vibration seen by a target such a sine wave at a dominant frequency with added noise. A MATLAB script will be needed to filter, window, perform FFT, and finally extract the spectral energy to be displayed graphically along with the filtered time-series.

1. Data will be generated at a sampling frequency of 500 Hz for three streams to represent acceleration values.
2. The raw data will be band-pass filtered to pass the frequencies from 10 to 200 Hz with minimal band-pass attenuation.
3. The filtered data will then be windowed using a window of your choice (Hamming, Hanning, Blackman, etc.). The size of the window will depend on the length of the FFT.
4. The windowed time-series data will then be converted to the frequency domain using an FFT. The length of the FFT must be set such that a frequency resolution of 0.5 Hz is achieved.

Note: The time-series data will be generated continuously (a sample at a time) whereas the spectral analysis will require a window of data. For the graphical representation it is advised to display several seconds of the time-series but to perform spectral analysis only on a smaller fraction of the most recent data. In addition, the windows should overlap between 50-80%. That is the window will consist of 50-80% of old data with 20-50% new data.

Phase 2: Simulated Data & Spectral Analysis in FPGA to MATLAB

In phase 2, simulated data generated in MATLAB will be initialized onto the FPGA in BRAMs. The spectral analysis will now be performed on the actual FPGA. The results along with the raw time-series data must then be sent to MATLAB using UART communication to be displayed.

Phase 3: Reading Acceleration from FPGA via SPI

In the third phase, you will develop a master SPI controller in order to configure the MPU6050 IC as well as read the acceleration values from the MPU6050. Configuration and readings are done by reading and writing to internal registers within the MPU6050. Please refer to [MPU6050 Datasheet](#) and [MPU6050 Register Map](#) for details.

MPU6050 Configuration:

- Enable all three accelerometer axis x, y, and z.
- Disable the temperature and gyroscope sensors.
- Use either internal clock or gyroscope clock.
- Set the sampling frequency to 500 Hz.
- Optional: Configure the interrupt to be notified when new samples are available instead of polling.

Phase 4: Full System

The final phase will merge the designs used in the previous phases. The final system will read the acceleration values from the MPU6050 to the FPGA, spectral analysis will be performed on the FPGA, and the results along with raw time-series data for all 3 axis will be sent over UART to MATLAB to be displayed.

