Project Checklist:

Video Part

Camera Input and Retrieval:
1. Display the results of the camera input on the monitor to check if the camera input is being stored and accessed correctly in the ZBT.

Video Processing and Visualization:
1. Color Detection: Display blue pixels registered by the program with white pixels on the monitor. This allows us to see how the color thresholds should be adjusted to register the user's hands. Additionally, demonstrate filtering random pixels registered as blue out from the image by comparing the case when the filter is applied and not applied.
2. Weighted Average: Demonstrate its operation in Modelsim. Next, use it in the Position Calculator and test.
3. Position Calculator: Demonstrate its operation in Modelsim. Next, display a box on the screen at each of the calculated coordinates to visually demonstrate the result of the calculations when done on the FPGA.
4. Overall: Demonstrate the block's operation on the monitor when connecting all the modules together. The two blocks on screen should follow the motion of the two lights and visually be at the center of each hand's area picked up by the camera. The results are overlayed over each other so it's easy to check how accurate the calculations are.

Motion Analyzer:
1. Beat Detection: Demonstrate its operation in Modelsim, showing both the start and the end of a beat. For testing with the FPGA, use the monitor to display blocks wherever a beat starts or ends. Using this visualization we can adjust parameters to change the sensitivity of the system to registering beats and see the results visually.
2. Motion Analysis: Show its operation in Modelsim. Next connect it with the Beat Detection module to show the overall operation in Modelsim.

Overall:
1. Finally, connect all parts together and check its correct operation on the FPGA. This testing is done by displaying the amplitude, acceleration, and beat_period results on the logic analyzer.

Audio Part

LPF/HPF:
The LPF/HPF will take in an audio input and output a high-passed and a low-passed version of the signal. The demonstration of these modules will include the comparison of the FPGA digital filter versus a digital filter from Matlab. The two should be almost identical. Also, when the low-pass and high-pass audio is added together, the final signal should be almost identical to the original.

Beat Generator/Detector:
The beat generator should generate a 1 clock-cycle active high signal to indicate when the start of a beat begins. The value of a beat period will be able to be programmed by the

user. If time permits, a beat detector will be incorporated into the system - the reliability of such a device is questionable.

Volume and Articulation Modulator:
This module will change the volume and articulation of one beat of audio, given an acceleration and velocity. Demonstrate this functionality by feeding in audio, acceleration, and velocity. If the velocity fed in is less, then the volume should be less; if the acceleration is less, then the audio should be smoother.

ROM FSM and RAM FSM:
These module interacts with the ROM , which stores the audio. Demonstrate the ability of the ROM FSM to read one beat at a time from memory and to pass it to the RAM. Use a push button to emulate a user-beat. The push button should allow the audio to output one beat at a time.

Tempo Modulator:
This module will change the tempo of the audio, given a user-beat and a music-beat (original beats). Demonstrate this functionality by feeding in a user-beat by using a push-button on the FPGA to see if the tempo of song can be changed without changing the intonation. There should be virtually no gaps or lag in playback and the module should be resilient to sudden changes in tempo.

Overall operation:
Use push buttons and switches to demonstrate the Audio Processing Module in its entirety. Push buttons will represent user-beats, and switches will specify different accelerations and velocities. The audio should play to the beat given by the user-beats and sound natural; the audio should be resilient to sudden changes in tempo (no gaps in playback, and no lag). The audio should be louder if the velocity is greater and the articulation should be shorter if the acceleration is greater.

Final Checkoff

Demonstrate the operation of the virtual conducting system. The tempo of the audio should be controlled by the frequency of the gestures, and the start of a gesture should mark the start of a beat. The system should be resilient to sudden changes in tempo and there should be little if any lag gaps in playback. The larger the motion of the gesture should result in a louder playback. Also, a gesture with more acceleration should result in a more pronounced articulation in the playback.