EE450/550: Digital Communication Systems Design and Implementation
Laboratory Assignment 3 – Group 1

Gaurav Shetty (#50169987)
Ketul Surati (#50166476)

Q.1. Explain with your own words, what you have done in this guided assignment and what you have learnt. You only need one paragraph, please be concise.

Ans. The guided assignment provides an insight as to how to design and implement a BFSK modulator in a FPGA module. To visualize the BFSK signal on the oscilloscope, a static sinusoidal wave wouldn't be sufficient and hence movement was to be given to the wave generated and displayed. This was enabled by shifting the phase (i.e. initializing the starting point '*j for every cycle*' of the sinusoidal wave with '*mov*') by a predefined amount for every frame (i.e. at every negative edge of the vertical synchronization signal). The signal generator was so designed that the same module was capable of generating two frequencies, depending on the bit value in the data stream sequence. Multiple frequencies were ensured by updating the *sineAddr* value by 1 and 2 for *195.3 kHz* and *390.6kHz*, for a bit value corresponding to 0 and 1 respectively. The data stream is initialized in the signal generator module itself with a symbol duration of *5.12µs* which determine the number of clock cycles (in our case *5.12µs/20ns=256*) during the symbol duration. This is important because it defines the number of sinusoidal cycles displayed for each bit value 0 or 1. As an extension to the project a switch input was defined for the oscilloscope module, which helps you to hold the signal or keep it moving.

Q.2. With your actual design the BFSK signal should be moving from right to left. What would you change from this design to display a BFSK signal moving from left to right?

Ans. The movement to the sinusoidal wave is provided by the introduction of the variable *mov* which determines the phase of the sinusoidal wave for each new frame. In other words, by assigning *mov* to j we define the first vector component for each frame. So therefore, if increasing the variable move from 0 to $(nc \times 256 - nf)$, creates a right to left motion, then decreasing the variable from the $(nc \times 256 - nf)$ to 0, will initialize the value of j in such a way that it creates a left to right motion as shown in the figure 1.

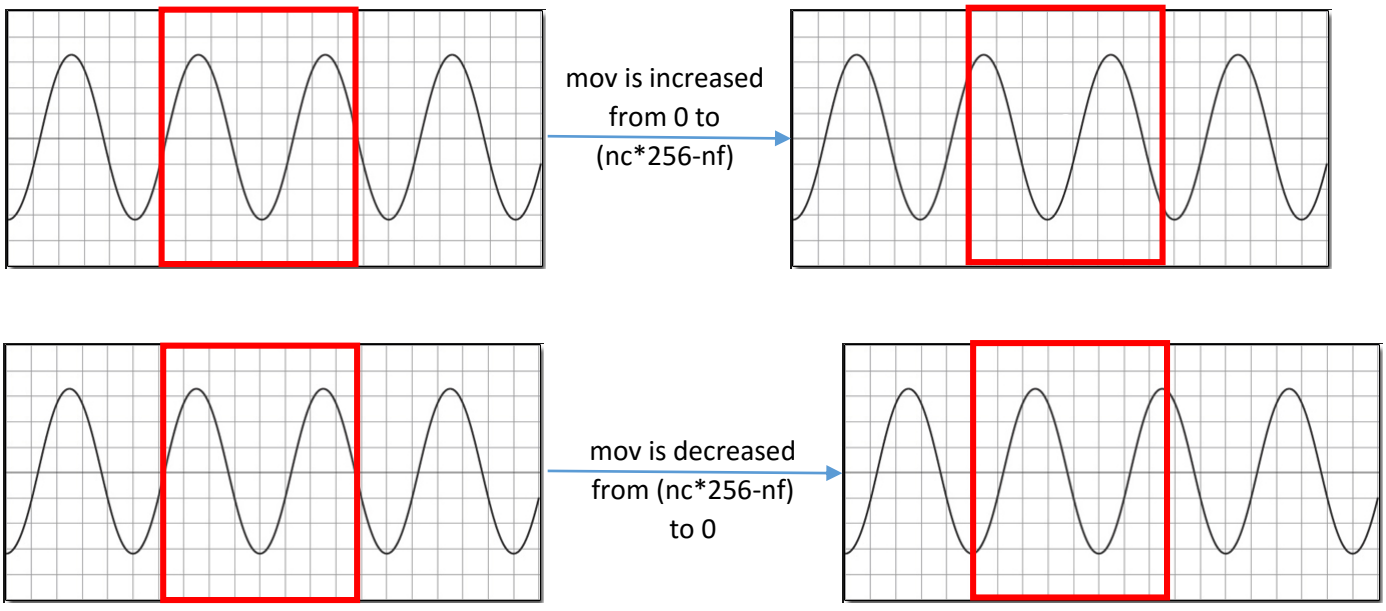

Figure 1 Effect of 'mov' on direction of motion where the red box defines the frame being displayed

Q.3. Consider a design where two sinusoidal wave generators are used to generate each one of the frequencies. What are the main changes that you should do in your code to generate and display the BFSK signal in the digital oscilloscope? Do not give details.

Ans. Separate signal generator modules will be created for each generator say *signal_generator1.v* and *signal_generator2.v* which outputs signals (i.e. *signal1* and *signal2*) of two frequencies (f_1 and f_2). A separate module say *BFSK_modulator.v* is created which takes in three inputs *CLOCK_50*, *signal1* and *signal2* and the same logic of checking the bit value and deciding the frequency is implemented in this module for every positive edge of clock. The only change is instead of the statements $sinAddr <= sinAddr + 1$ and $sinAddr <= sinAddr + 2$ we write $signal <= signal1$ and $signal <= signal1$ respectively. The register signal acts as an output of the current module. The last change required is to instantiate *signal_generator1, signal_generator2* whose outputs are fed to the instance of the *BFSK_modulator*. The output of the *BFSK_modulator* is further fed to the *oscilloscope* instance in the top level design.