

## Lab 7: Decoding NEC Protocol from an IR receiver

**Objective:** The goal of this assignment is to design and implement a system that decodes NEC protocol IR signals using a MicroBlaze-based FPGA system. An IR sensor connected through GPIO will serve as the input device for receiving NEC protocol commands.

**Background Information:** The NEC protocol is widely used in infrared remote control systems. It involves modulated signals with specific timing patterns to represent binary data. The key aspects of the NEC protocol include:

1. **Start Pulse:** A long pulse (9 ms) followed by a gap (4.5 ms).
2. **Data Transmission:** Logical "1" is represented by a 560  $\mu$ s pulse followed by a 1.69 ms gap, while logical "0" is represented by a 560  $\mu$ s pulse followed by a 560  $\mu$ s gap.
3. **Address and Command:** A 32-bit data packet consists of an 8-bit address, an 8-bit inverse address, an 8-bit command, and an 8-bit inverse command.

You are tasked to decode these signals and display the received address and command on a terminal connected to the FPGA system. Multiple descriptions of the NEC protocol are available online, here is an example <https://techdocs.altium.com/display/FPGA/NEC+Infrared+Transmission+Protocol>

### Tasks:

#### 1. Signal Processing:

- Capture the signal from the IR sensor connected to the GPIO pin.
- Implement by means of a timer a signal sampling mechanism to accurately measure the pulse durations.

#### 2. NEC Protocol Decoding:

- Start familiarizing with the timing: e.g. identify the start pulse and validate its timing.
- Decode the 32-bit data packet based on pulse and gap durations.
- Extract the address, inverse address, command, and inverse command from the received packet.
- Validate the data integrity by checking that the inverse values are correct.

#### 3. Software Implementation:

- Write C code for a function that implements the decoding algorithm.
- Use an interrupt or polling mechanism to read the GPIO input and process the signal.
- Store and print the decoded address and command values on a terminal connected to the MicroBlaze system.

#### 4. Testing and Validation:

- Test the system with a standard NEC-compatible remote control.
- Verify that the address and command values match the expected output for the buttons pressed.