## Demonstration Setup, File Locations, and Instructions

- Load the bitstream into the FPGA by executing \DE1_SoC_PS2_DEMO \demo_batch\ DE1_SoC_PS2_DEMO.bat
- Plug in the PS/2 mouse
- Press KEY[0] to enable data transfer
- Press KEY[1] to clear the display data cache
- The 7-segment display should change when the PS/2 mouse moves. The LEDR[2:0] will blink according to Table 5-4 when the left-button, right-button, and/or middle-button is pressed.

**Table 5-4 Description of 7-segment Display and LED Indicators**

| *Indicator Name* | *Description* |
|---|---|
| **LEDR[0]** | **Left button press indicator** |
| **LEDR[1]** | **Right button press indicator** |
| **LEDR[2]** | **Middle button press indicator** |
| **HEX0** | **Low byte of X displacement** |
| **HEX1** | **High byte of X displacement** |
| **HEX2** | **Low byte of Y displacement** |
| **HEX3** | **High byte of Y displacement** |

## 5.8 IR Emitter LED and Receiver Demonstration

DE1-SoC system CD has an example of using the IR Emitter LED and IR receiver. This demonstration is coded in Verilog HDL.

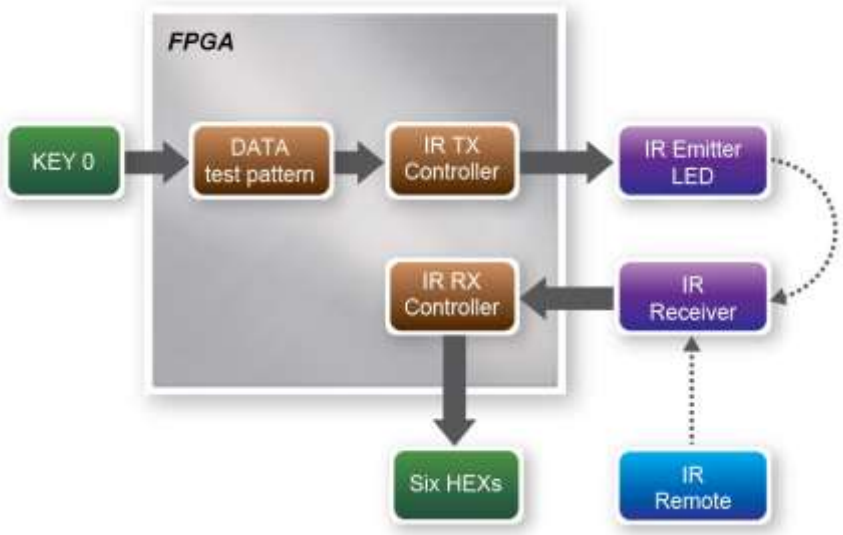


**Figure 5-12 Block diagram of the IR emitter LED and receiver demonstration**

Figure 5-12 shows the block diagram of the design. It implements a IR TX Controller and a IR RX Controller. When KEY0 is pressed, data test pattern generator will generate data to the IR TX Controller continuously. When IR TX Controller is active, it will format the data to be compatible with NEC IR transmission protocol and send it out through the IR emitter LED. The IR receiver will decode the received data and display it on the six HEXs. Users can also use a remote to send data to the IR Receiver. The main function of IR TX /RX controller and IR remote in this demonstration is described in the following sections.

## ■ IR TX Controller

Users can input 8-bit address and 8-bit command into the IR TX Controller. The IR TX Controller will encode the address and command first before sending it out according to NEC IR transmission protocol through the IR emitter LED. The input clock of IR TX Controller should be 50MHz.

The NEC IR transmission protocol uses pulse distance to encode the message bits. Each pulse burst is 562.5µs in length with a carrier frequency of 38kHz (26.3µs).

Figure 5-13 shows the duration of logical “1” and “0”. Logical bits are transmitted as follows:

- Logical '0' – a 562.5µs pulse burst followed by a 562.5µs space with a total transmit time of 1.125ms

- Logical '1' – a 562.5µs pulse burst followed by a 1.6875ms space with a total transmit time of 2.25ms

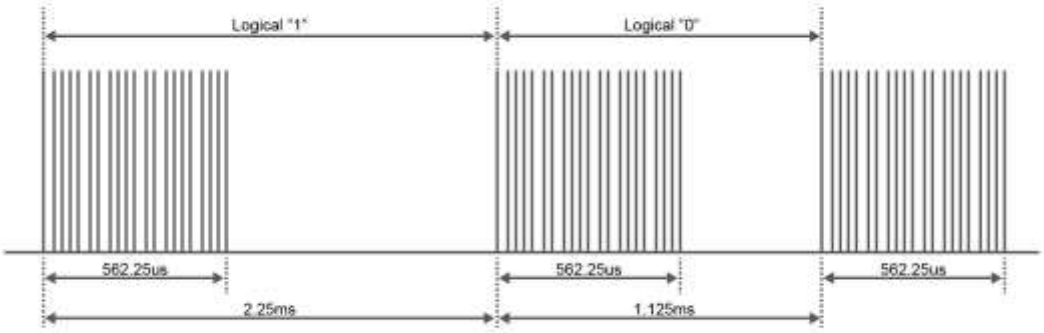


**Figure 5-13 Duration of logical “1”and logical “0”**

Figure 5-14 shows a frame of the protocol. Protocol sends a lead code first, which is a 9ms leading pulse burst, followed by a 4.5ms window. The second inversed data is sent to verify the accuracy of the information received. A final 562.5µs pulse burst is sent to signify the end of message transmission. Because the data is sent in pair (original and inverted) according to the protocol, the overall transmission time is constant.

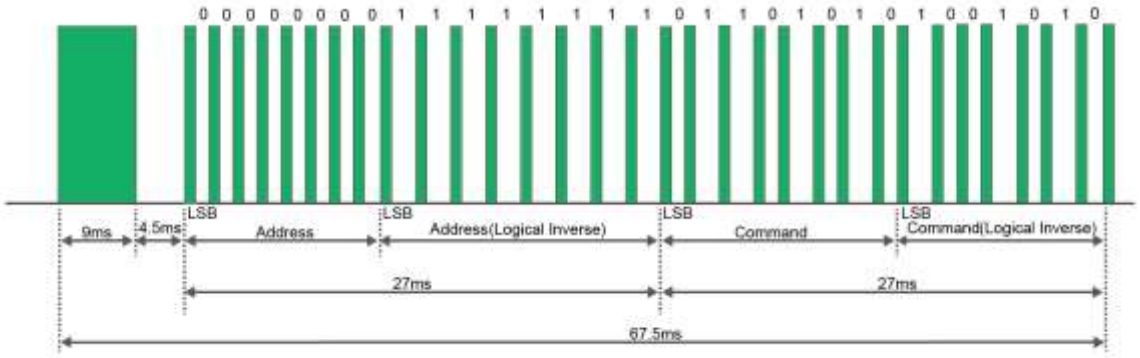


**Figure 5-14 Typical frame of NEC protocol**

Note: The signal received by IR Receiver is inverted. For instance, if IR TX Controller sends a lead code 9 ms high and then 4.5 ms low, IR Receiver will receive a 9 ms low and then 4.5 ms high lead code.

## ■ IR Remote

When a key on the remote controller show in **Figure 5-15** is pressed, the remote controller will emit a standard frame, as shown in **Table 5-5**. The beginning of the frame is the lead code, which represents the start bit, followed by the key-related information. The last bit end code represents the end of the frame. The value of this frame is completely inverted at the receiving end.

**Figure 5-15 The remote controller used in this demonstration**

**Table 5-5 Key Code Information for Each Key on the Remote**

| *Key* | *Key Code* | *Key* | *Key Code* | *Key* | *Key Code* | *Key* | *Key Code* |
|---|---|---|---|---|---|---|---|
| A | 0x0F | B | 0x13 | C | 0x10 | ⏻ | 0x12 |
| 1 | 0x01 | 2 | 0x02 | 3 | 0x03 | ▲ | 0x1A |
| 4 | 0x04 | 5 | 0x05 | 6 | 0x06 | ▼ | 0x1E |
| 7 | 0x07 | 8 | 0x08 | 9 | 0x09 | ▲ | 0x1B |
| ⏏ | 0x11 | 0 | 0x00 | ↩ | 0x17 | ▼ | 0x1F |
| ⏯ | 0x16 | ◀ | 0x14 | ▶ | 0x18 | 🔇 | 0x0C |

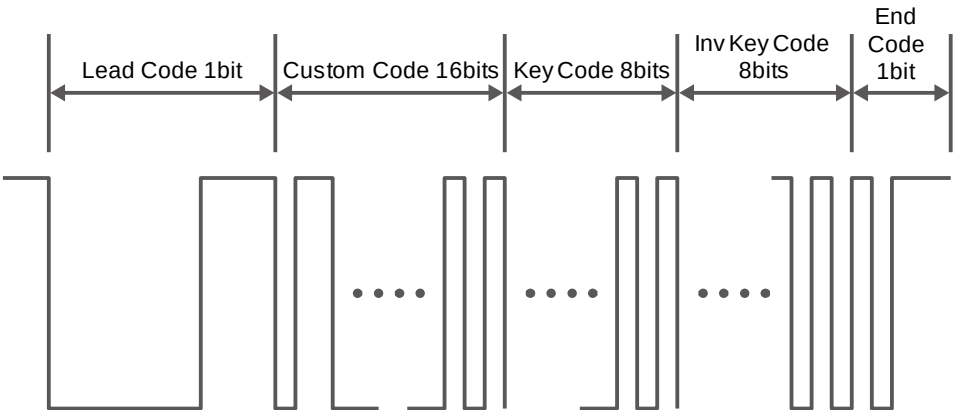


**Figure 5-16 The transmitting frame of the IR remote controller**

## ■ IR RX Controller

The following demonstration shows how to implement the IP of IR receiver controller in the FPGA. **Figure 5-17** shows the modules used in this demo, including Code Detector, State Machine, and Shift Register. At the beginning the IR receiver demodulates the signal inputs to the Code Detector . The Code Detector will check the Lead Code and feedback the examination result to the State Machine.

The State Machine block will change the state from IDLE to GUIDANCE once the Lead Code is detected. If the Code Detector detects the Custom Code status, the current state will change from GUIDANCE to DATAREAD state. The Code Detector will also save the receiving data and output to the Shift Register and display on the 7-segment. **Figure 5-18** shows the state shift diagram of State Machine block. The input clock should be 50MHz.

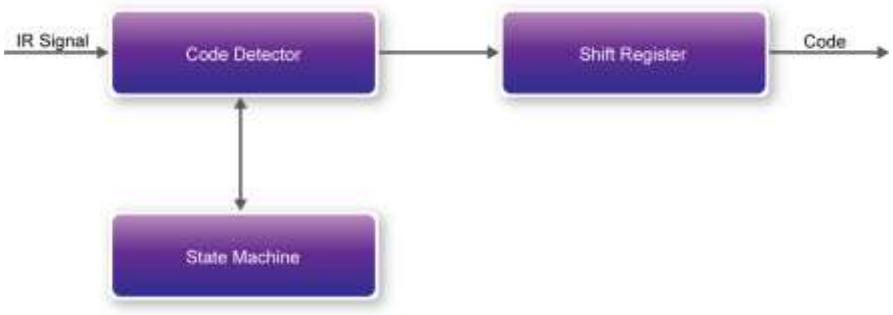


**Figure 5-17 Modules in the IR Receiver controller**

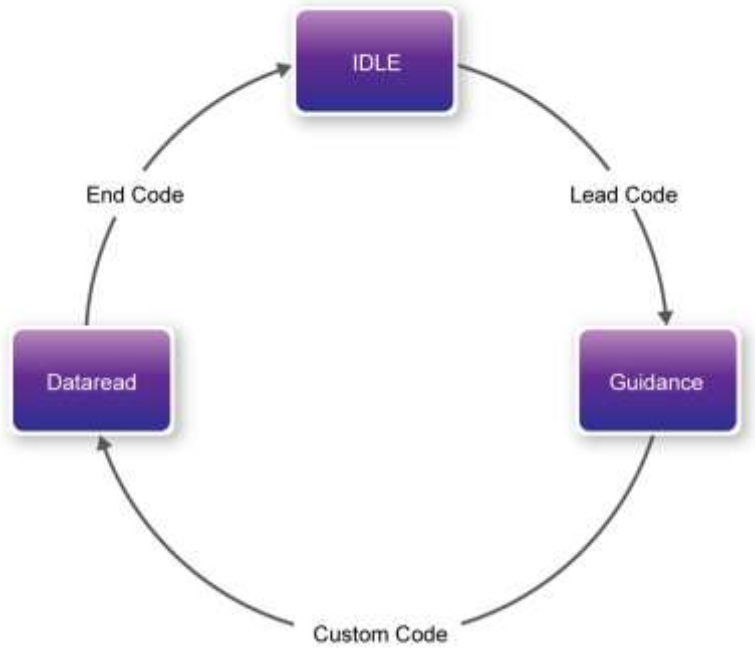


**Figure 5-18 State shift diagram of State Machine block**

## Demonstration Source Code

- Project directory: DE1_SoC_IR
- Bitstream used: DE1_SOC_IR.sof

## Demonstration Batch File

Demo batch file directory: DE1_SoC_IR \demo_batch

The folder includes the following files:

- Batch file: DE1_SoC_IR.bat
- FPGA configuration file : DE1_SOC_IR.sof

## Demonstration Setup, File Locations, and Instructions

- Load the bitstream into the FPGA by executing DE1_SoC_IR \demo_batch\ DE1_SoC_IR.bat
- Keep pressing KEY[0] to enable the pattern to be sent out continuously by the IR TX Controller.
- Observe the six HEXs according to **Table 5-6**
- Release KEY[0] to stop the IR TX.
- Point the IR receiver with the remote and press any button

- Observe the six HEXs according to **Table 5-6**

**Table 5-6 Detailed Information of the Indicators**

| *Indicator Name* | *Description* |
|---|---|
| **HEX5** | **Inversed high byte of DATA(Key Code)** |
| **HEX4** | **Inversed low byte of DATA(Key Code)** |
| **HEX3** | **High byte of ADDRESS(Custom Code)** |
| **HEX2** | **Low byte of ADDRESS(Custom Code)** |
| **HEX1** | **High byte of DATA(Key Code)** |
| **HEX0** | **Low byte of DATA (Key Code)** |

## 5.9 ADC Reading

This demonstration illustrates steps to evaluate the performance of the 8-channel 12-bit A/D Converter ADC7928. The DC 5.0V on the 2x5 header is used to drive the analog signals by a trimmer potentiometer. The voltage can be adjusted within the range between 0 and 5.0V. The 12-bit voltage measurement is displayed on the NIOS II console. **Figure 5-19** shows the block diagram of this demonstration.

If the input voltage is -2.5V ~ 2.5V, a pre-scale circuit can be used to adjust it to 0 ~ 5V.