Circuit-Level Modelling and Verification of Miller Encoding (Delay Modulation) Protocol for Serial Data Communication

Theory

The Miller Encoding protocol, also referred to as Delay Modulation, is a self-clocking binary encoding method designed to represent a digital bitstream while significantly reducing the required signal bandwidth. Unlike standard Non-Return-to-Zero (NRZ) data, which represents logic levels with static voltages, Miller encoding utilizes transitions to maintain synchronization without a dedicated clock wire. This specific implementation is designed to operate with a base clock frequency providing a defined bit period, ensuring stable asynchronous recovery within a mixed-signal simulation environment.

The theoretical foundation of this design relies on synchronized phase-shifts: a logic '1' is indicated by a transition in the middle of the bit interval, while a logic '0' has no middle transition. A transition is only appended at the boundary between two consecutive '0's to maintain timing integrity.

Principle of Operation

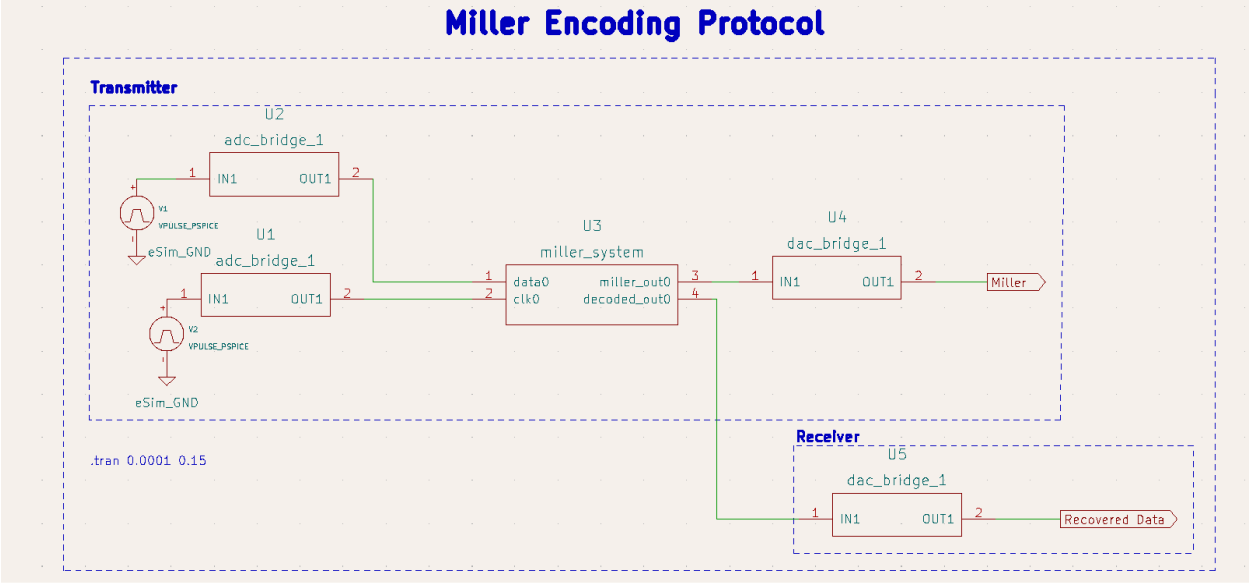
- **Self-Clocking Protocol:** The Miller system embeds timing information directly into the data stream, eliminating the need for a shared global clock between the transmitter and receiver.
- **Logical Encoding Rules:** * **Logic High (1):** A signal inversion occurs exactly at the midpoint of the bit interval.
    - **Logic Low (0):** The signal remains at its current level throughout the midpoint of the bit interval.
    - **Phase Synchronization:** If two '0's occur in succession, a transition is forced at the bit boundary to prevent synchronization loss.
- **Mixed-Signal Architecture:** The design utilizes an XOR-based logic network implemented via NgVeri (Verilog) to process digital bitstreams and convert them into analog-compatible spice models using adc_bridge and dac_bridge components.
- **Data Recovery Logic:** The receiver utilizes edge-detection and clock-division logic to sample the incoming Miller waveform. By monitoring transitions relative to the local bit window, the original NRZ bitstream is reconstructed.
- **Propagation Delay Verification:** To ensure proper synchronization, the system acknowledges a mandatory one-clock-period (T_{clk}) processing latency, which is required for the decoder to validate bit-middle transitions.

Nomenclature

- **Transmitter (Tx) Side**
    - **Data_in:** The parallel NRZ input data provided to the encoder.
    - **CLK:** The primary system clock governing the bit period and midpoint timing.

    - **Miller_out:** The final delay-modulated serial data stream transmitted over the line.
- **Receiver (Rx) Side**
    - **Miller_in:** The encoded serial data received from the transmitter.
    - **Recovered_Data:** The final parallel NRZ output produced by the decoder once the bitstream is successfully processed.

Schematic

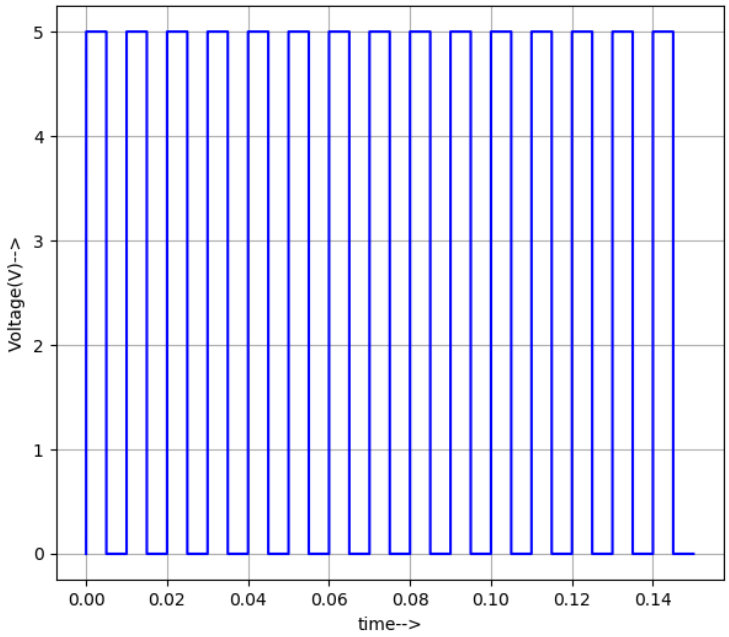


Data Flow and Transmission Sequence

1. The user provides the NRZ bitstream at **Data_in** and a synchronized clock at **CLK** to the NgVeri block.
2. The logic performs a phase-shift calculation using an internal frequency divider to determine transition points.
3. The **Miller_out** waveform is generated, adhering to the mid-bit transition rules for '1's and boundary rules for consecutive '0's.
4. The receiver (decoder) monitors the **Miller_in** line for edges relative to the bit window.
5. Once a transition is detected at a bit midpoint, the receiver latches a logic '1'; otherwise, it latches a logic '0'.
6. After a processing latency of exactly one clock period, the reconstructed data is output at **Recovered_Data**.

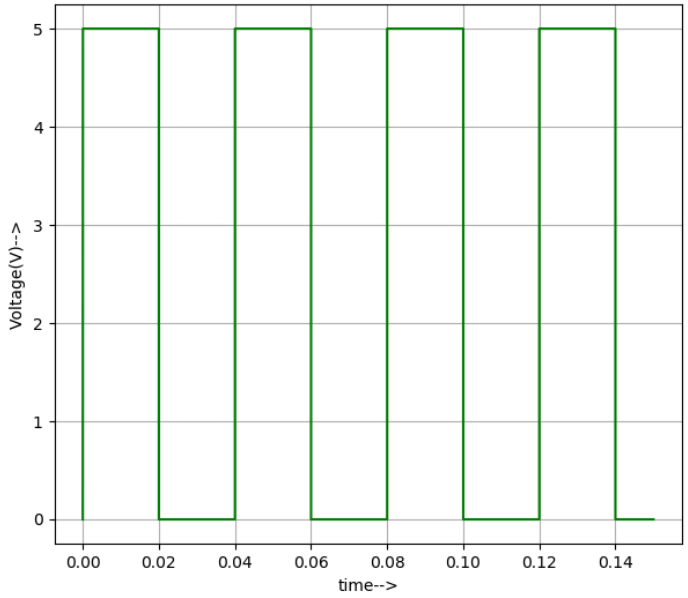
Simulation Results

Figure 1: CLK Signal



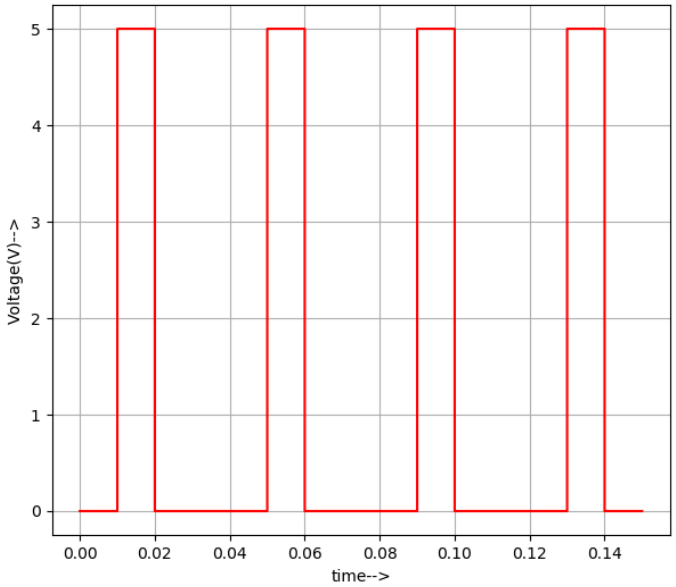
The primary system clock governing the bit period and midpoint timing for the encoding logic.

Figure 2: DATA_IN Signal



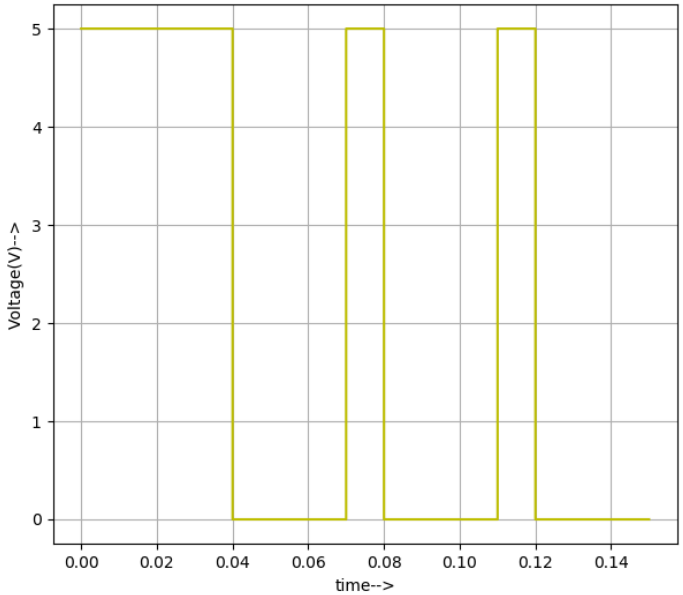
The original parallel NRZ (Non-Return-to-Zero) binary input data provided to the encoder.

Figure 3: Miller_out Signal



The final delay-modulated serial bitstream showing transitions at bit midpoints for '1's and bit boundaries for consecutive '0's.

Figure 4: Recovered_Data Signal



The reconstructed NRZ bitstream produced by the receiver, demonstrating successful data transfer with a one-clock-period propagation delay.

Conclusion

With this implementation of the Miller Encoding protocol, binary data is successfully transferred and recovered while ensuring high bandwidth efficiency. The simulation results confirm that the system maintains frame synchronization and accurately reproduces the input bitstream with a predictable propagation delay.

References

- **Hecht, M.** (1969). "Delay Modulation". *Proceedings of the IEEE*, Vol. 57, No. 7, pp. 1314-1316.
- **J. P. S. S. Beckers (Assigned to U.S. Philips Corporation)** “Modified Miller Code Encoder”(https://patents.google.com/patent/US4227184A/en)
- **Xiong, Fuqin.** (2000). *Digital Modulation Techniques*. Artech House, pp. 131-135.
- **FOSSEE eSim User Manual:** IIT Bombay (To get familiar with mixed-signal simulation).