Internship at IIIT Allahabad

Universal Asynchronous Receiver-Transmitter (UART) with Power Management

Submitted By

ARYAN SINGH
SEMESTER - VII
ROLL NO: 220104009

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY SENAPATI, MANIPUR

OCTOBER, 2025

Declaration

In this submission, I have expressed my idea in my own words, and I have adequately cited and referenced any ideas or words that were taken from another source. I also declare that I adhere to all principles of academic honesty and integrity and that I have not misrepresented or falsified any ideas, data, facts, or sources in this submission. If any violation of the above is made, I understand that the institute may take disciplinary action. Such a violation may also engender disciplinary action from the sources which were not properly cited or permission not taken when needed.

ARYAN SINGH
220104009

Acknowledgement

I would like to express my sincere gratitude to Prof. Manish Goswami for his invaluable guidance, support, and mentorship throughout this project. His expertise in VLSI design and digital systems has been instrumental in shaping this work.

I am thankful to the Department of Electronics and Communication Engineering at IIIT Allahabad for providing the necessary resources and facilities that made this project possible. I also acknowledge the support of my fellow students and the laboratory staff who assisted during the implementation phase.

Finally, I extend my appreciation to my family and friends for their constant encouragement and support throughout my academic journey.

Aryan Singh

प्रज्ञानम् ब्रह्म

भारतीय सूचना प्रौद्योगिकी संस्थान, इलाहाबाद

Indian Institute of Information Technology, Allahabad

An Institute of National Importance by Act of Parliament

Deoghat, Jhalwa, Prayagraj-211015 (U.P.) INDIA

Ph.: 0532-2922025, 2922067, Fax : 0532-2922125, Web : www.iiita.ac.in, E-mail : contact@iiita.ac.in

F.NO: IIIT-A/INT/IT/2025/S-34 Dated 16/07/2025

To whom it may concern

This is to certify that **Aryan Singh S/o Shri Satya Narain Singh** from "IIIT Manipur" has successfully completed summer internship program on the topic of **"UART with Power Management"** from 16.05.2025 to 16.07.2025 at the Department of Electronics &Communication Engg. Indian Institute of Information Technology Allahabad, Prayagraj under the supervision of Prof. Manish Goswami.

The performance of the candidates has been Poor/Average/Satisfactory/[✓ Very Good]/Excellent during the internship.

We wish him all the best for his future endeavors.

Manish Goswami

Supervisor

Faculty-incharge-internship

Faculty Incharge (Internship)
Indian Institute of Information Technology Allahabad
Deoghat, Jhalwa, Prayagraj-211015 (UP) INDIA

Department of Electronics and Communication Engineering
Indian Institute of Information Technology Senapati, Manipur

Dr. Nagesh Ch
HOD, ECE

Email: supervisoremail@iiitmanipur.ac.in
Contact No: +91 9678554904

To Whom It May Concern

This is to certify that the project/internsip report entitled **"Internship at IIIT Allahabad",** submitted to the department of Electronics and Communication Engineering , Indian Institute of Information Technology Senapati, Manipur in partial fullfillment for the award of degree of Bachelor of Technology in Electronics and Communication Engineering is record bonafide work carried out by **Aryan Singh** bearing roll number ROLLNO. 220104009

Signature of Faculty Advisor

(Dr. Nagesh Ch)

Signature of the Examiner 1 ....
Signature of the Examiner 2 ....
Signature of the Examiner 3 ....
Signature of the Examiner 4 ....

Abstract

This report presents the complete implementation of a Universal Asynchronous Receiver-Transmitter (UART) enhanced with advanced power management features. The project details the entire design workflow, from RTL design and functional verification in Verilog to synthesis and implementation using the Xilinx Vivado Design Suite. The final design demonstrates robust, full-duplex serial communication with configurable parameters and intelligent, activity-based power optimization. The UART supports multiple baud rates (2400, 4800, 9600, 19200 bps), configurable parity modes (None, Odd, Even), and implements four distinct power modes (Normal, Low Power, Sleep, and Deep Sleep) to minimize power consumption in battery-powered applications.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

The Universal Asynchronous Receiver-Transmitter (UART) is a fundamental serial communication peripheral in modern digital systems. This project details the design and implementation of a UART with a primary focus on power optimization techniques suitable for battery-powered and low-power applications.

Serial communication protocols like UART are essential in embedded systems, IoT devices, and microcontroller applications where multiple devices need to exchange data over limited connection lines. The asynchronous nature of UART makes it particularly suitable for applications where clock synchronization between communicating devices is challenging or impossible.

1.2 Objectives

The primary objectives of this project are:

- Design and implement a fully functional UART communication system
- Integrate advanced power management features with multiple power modes
- Achieve configurability in baud rates and parity settings
- Implement comprehensive error detection and handling mechanisms
- Verify the design through extensive simulation and testing
- Synthesize and implement the design on FPGA hardware
- Analyze resource utilization and power consumption

1.3 Key Design Features

The implemented UART system incorporates the following features:

- **Communication Protocol**: Full-duplex, asynchronous communication with 8 data bits, 1 start bit, and 1 stop bit
- **Configurability**: User-selectable baud rates (2400, 4800, 9600, 19200 bps) and parity modes (None, Odd, Even)
- **Error Handling**: Integrated detection for parity, framing, and overrun errors
- **Power Management**: An intelligent controller manages four distinct power modes: Normal, Low Power, Sleep, and Deep Sleep
- **Power Optimization**: Fine-grained clock gating employed to minimize dynamic power consumption during idle periods

1.4 UART Communication Protocol

UART is a point-to-point communication protocol that transmits data asynchronously between two devices [2]. Unlike synchronous protocols, UART does not require a shared clock signal between the transmitter and receiver. Instead, both devices must be configured to operate at the same baud rate.

The basic UART frame structure consists of:

- Start bit: Always logic '0', signals the beginning of data transmission
- Data bits: Typically 5-9 bits, with 8 bits being most common
- Parity bit (optional): Used for error detection
- Stop bits: One or more '1' bits, signals the end of frame transmission

1.5 Power Management in Digital Systems

Power consumption in digital circuits consists of two main components [1]:

- **Dynamic Power**: Consumed during signal transitions, proportional to switching activity
- **Static Power**: Consumed due to leakage currents, independent of switching activity

Clock gating is a widely used technique to reduce dynamic power consumption by selectively disabling clock signals to unused circuit blocks [3]. This project implements intelligent clock gating based on system activity and user-defined power modes.

Chapter 2

Design Architecture and Implementation

2.1 System Overview

The UART is designed with a hierarchical and modular architecture to facilitate testing, integration, and reuse. The system consists of the following key modules [2]:

- **Duplex**: Top-level wrapper integrating the transmitter and receiver
- **TxUnit** & **RxUnit**: Core transmitter and receiver modules
- **PowerManager**: State machine controller for power modes and clock gating
- **BaudGen**: Configurable baud rate generator
- **ErrorCheck**: Parity generation and error detection logic
- **PISO** & **SIPO**: Shift registers for parallel-to-serial and serial-to-parallel data conversion
- **ClockGate**: Custom clock gating cells for power savings

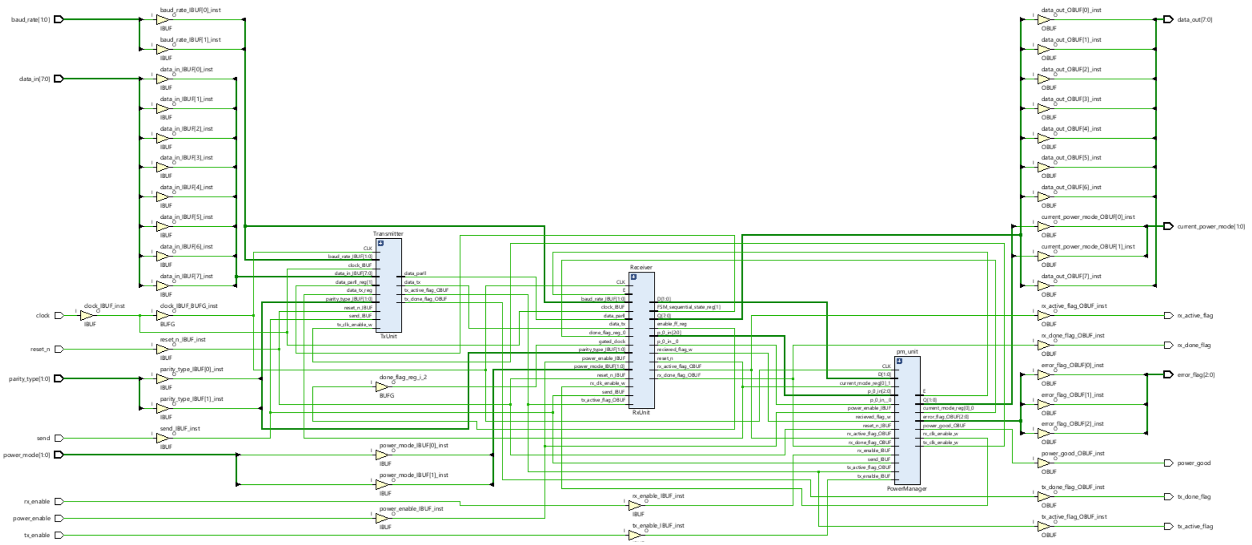


Figure 2.1: System Architecture Schematic

2.2 Top-Level Module Interface

The `Duplex` module serves as the primary interface for the UART system, providing both standard UART functionality and power management controls.

```
module Duplex (
    // Standard UART Interface
    input   wire        reset_n,
    input   wire        send,
    input   wire        clock,
    input   wire [1:0]  parity_type,
    input   wire [1:0]  baud_rate,
    input   wire [7:0]  data_in,
    output  wire [7:0]  data_out,
    output  wire        tx_done_flag,
    output  wire        rx_done_flag,
    output  wire [2:0]  error_flag,

    // Power Management Interface
    input   wire        power_enable,
    input   wire        tx_enable,
    input   wire        rx_enable,
    input   wire [1:0]  power_mode,
    output  wire        power_good,
    output  wire [1:0]  current_power_mode
);
    // Internal module instantiations and wiring...
endmodule
```

Listing 2.1: Top-Level Module Interface (Duplex.v)

2.3 Power Management Implementation

Power efficiency is achieved through a sophisticated state machine in the `PowerManager` module and explicit `ClockGate` cells. The manager transitions between four power modes based on system activity, enabling module clocks only when required [1, 3].

```
always @(posedge clock or negedge reset_n) begin
    if (~reset_n) begin
        current_mode <= DEEP_SLEEP_MODE;
    end else begin
        if (power_enable) begin
            // If any activity is detected, switch to Normal Mode
            if (tx_active || rx_active || send_request || data_received) begin
                current_mode <= NORMAL_MODE;
            // Otherwise, enter the user-requested low-power mode
            end else begin
                current_mode <= power_mode;
            end
        end else begin
            current_mode <= DEEP_SLEEP_MODE;
        end
    end
end
```

Listing 2.2: Power Manager State Machine (PowerManager.v)

2.4 Baud Rate Generator

The baud rate generator creates the timing reference for serial data transmission and reception. It supports four configurable baud rates and includes clock gating for power optimization.

```
always @(posedge gated_clock or negedge reset_n)
begin
  if(~reset_n) begin
    clock_ticks <= 14'd0;
    baud_clk    <= 1'b0;
  end
  else if (clk_enable) begin
    if(clock_ticks >= final_value) begin
      baud_clk    <= ~baud_clk;
      clock_ticks <= 14'd0;
    end
    else begin
      clock_ticks <= clock_ticks + 1'd1;
    end
  end
```

```
end
```

Listing 2.3: Baud Rate Generator (BaudGen.v)

2.5 Error Detection and Parity Logic

The ErrorCheck module handles parity generation for transmission and verification for reception, supporting three parity modes: None, Odd, and Even.

```
always @(*)
begin
    if (clk_enable) begin
        case (parity_type)
          NOPARITY00, NOPARITY11: begin
            error_parity = 1'b0;
            parity_bit_out = 1'b1;
          end
          ODD: begin
            error_parity = (^raw_data == parity_bit) ? 1'b0 : 1'b1;
            parity_bit_out = ^raw_data;
          end
          EVEN: begin
            error_parity = (^raw_data == parity_bit) ? 1'b1 : 1'b0;
            parity_bit_out = ~(^raw_data);
          end
          default: begin
            error_parity = 1'b1;
            parity_bit_out = 1'b1;
          end
        endcase
    end
    else begin
        error_parity = 1'b0;
        parity_bit_out = 1'b1;
    end
end
```

Listing 2.4: Parity Calculation in ErrorCheck Module (ErrorCheck.v)

2.6 Data Conversion Modules

2.6.1 Parallel-In, Serial-Out (PISO)

The PISO shift register in the transmitter serializes the parallel data frame for transmission over the serial line.

```
always @(posedge baud_clk or negedge reset_n) begin
    if (~reset_n) begin
```

```
        state <= IDLE;
        data_tx <= 1'b1;
        active_flag <= 1'b0;
        done_flag <= 1'b1;
        stop_count <= 4'd0;
        frame_r <= 11'h7FF;
    end
    else if (clk_enable) begin
        case (state)
            IDLE: begin
                if (send) begin
                    frame_r <= frame;
                    state <= ACTIVE;
                end
            end
            ACTIVE: begin
                if (stop_count >= 4'd10) begin
                    state <= IDLE;
                end
                else begin
                    data_tx <= frame_r[0];
                    frame_r <= frame_r >> 1;
                    stop_count <= stop_count + 1'b1;
                    active_flag <= 1'b1;
                    done_flag <= 1'b0;
                end
            end
        endcase
    end
end
```

Listing 2.5: PISO State Machine for Transmission (PISO.v)

2.6.2 Serial-In, Parallel-Out (SIPO)

The SIPO register in the receiver converts the incoming serial data stream back into parallel form for processing.

```
always @(posedge baud_clk or negedge reset_n)
begin
  if (~reset_n) begin
    state <= IDLE;
    active_flag <= 1'b0;
    // other resets...
  end
  else if (clk_enable) begin
    case (state)
      IDLE: begin
        if(~data_tx) begin // Start bit detected
          state <= CENTER;
          active_flag <= 1'b1;
        end
      end
      CENTER: begin // Find the center of the start bit
        // Logic to sample at bit center...
        state <= FRAME;
      end
      FRAME: begin // Shift in the data frame
        if(frame_counter == 4'd10) begin
          recieved_flag <= 1'b1;
          state <= HOLD;
          active_flag <= 1'b0;
        end
        else begin
          // Shift logic...
        end
      end
      HOLD: begin // Wait before returning to idle
          state <= IDLE;
      end
    endcase
  end
end
```

Listing 2.6: SIPO State Machine for Reception (SIPO.v)

2.7 Timing Constraints

The timing behavior of the design is defined using Xilinx Design Constraints (XDC) [1]. The primary constraint establishes the main system clock specification.

```
# Define a 100 MHz system clock on the 'clock' port
create_clock -period 10.000 -name sys_clk -waveform {0.000 5.000} [get_ports clock]
```

Listing 2.7: Primary Clock Constraint (Constraint.xdc)

This constraint specifies:

- A clock named sys_clk on the top-level port clock
- A clock period of 10.0 ns, corresponding to a 100 MHz frequency
- A 50% duty cycle waveform, rising at 0.0 ns and falling at 5.0 ns

Chapter 3

Verification and Testing

3.1 Testbench Architecture

A comprehensive Verilog testbench was developed to verify the UART functionality across various operational scenarios. The testbench implements automated testing procedures to validate:

- Data integrity across different data patterns and parity configurations
- Dynamic baud rate switching capabilities
- Power management state transitions and controls
- System recovery mechanisms after reset conditions
- Stress testing with rapid, consecutive transmissions

3.2 Test Cases and Scenarios

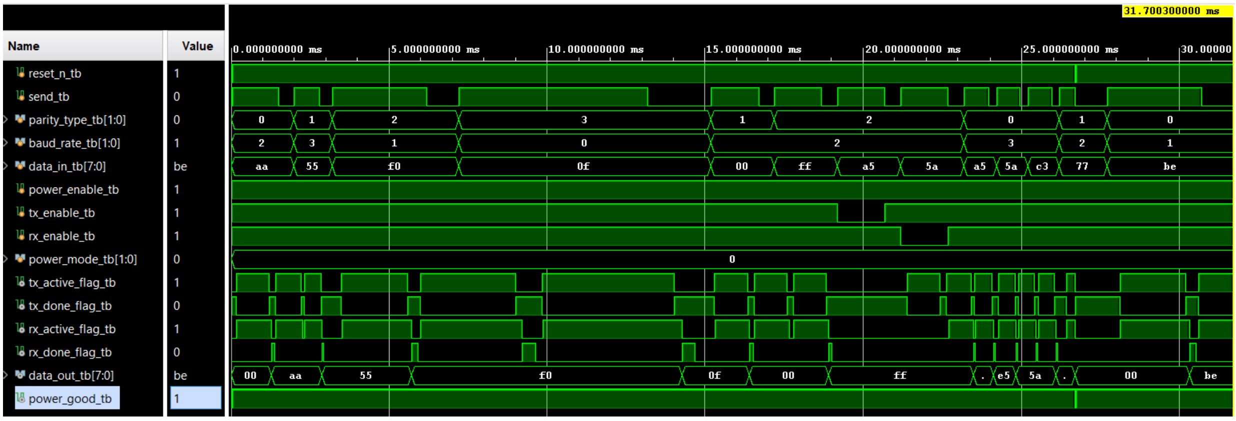
The verification process includes eleven distinct test cases:

1. Basic transmission with no parity (0xAA)
2. Odd parity transmission (0x55)
3. Even parity transmission (0xF0)
4. Slow baud rate testing (0x0F)
5. Edge case: All zeros (0x00)
6. Edge case: All ones (0xFF)
7. Power management: TX disabled
8. Power management: RX disabled

9. Rapid sequence testing
10. Reset during transmission
11. Final comprehensive test (0xBE)

3.3 Simulation Results

Functional simulation using Vivado Simulator confirmed correct operation across all test scenarios [1].



Name	Value
reset_n_tb	1
send_tb	0
parity_type_tb[1:0]	0
baud_rate_tb[1:0]	1
data_in_tb[7:0]	be
power_enable_tb	1
tx_enable_tb	1
rx_enable_tb	1
power_mode_tb[1:0]	0
tx_active_flag_tb	1
tx_done_flag_tb	0
rx_active_flag_tb	1
rx_done_flag_tb	0
data_out_tb[7:0]	be
power_good_tb	1

Figure 3.1: Simulation Waveform showing single byte transmission with key signals including clock, send command, input data, serial output, and completion flag

The simulation results validated:

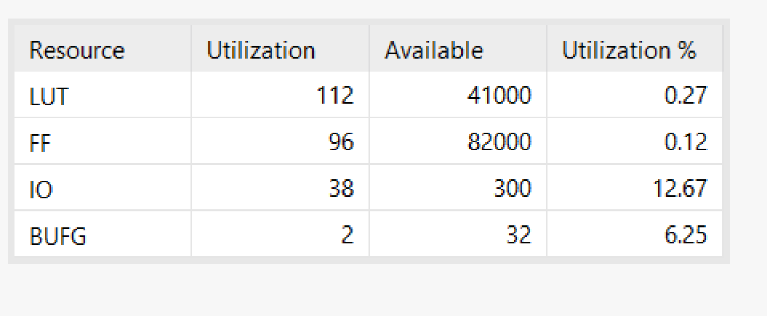
- **Data Framing**: Correct serial stream formatting with start, data, parity, and stop bits
- **Timing Accuracy**: Precise baud rate timing for all configured speeds
- **Power Control**: Accurate power mode transitions based on system activity
- **Error Detection**: Proper identification and flagging of injected transmission errors

Chapter 4

Implementation Results

4.1 Hardware Utilization Analysis

The FPGA implementation demonstrates efficient resource utilization, making the design suitable for resource-constrained applications [1].



Resource	Utilization	Available	Utilization %
LUT	112	41000	0.27
FF	96	82000	0.12
IO	38	300	12.67
BUFG	2	32	6.25

The low utilization percentages indicate that the design is highly optimized and can be easily integrated into larger systems without significant resource constraints.

4.2 Power Consumption Analysis

Power analysis was conducted using Vivado Power Analyzer with realistic operating conditions including a 100 MHz system clock and 25% signal toggle rate [1].

Table 4.1: Power Consumption Analysis Across Different Modes

Power Mode	**Dynamic** (mW)	**Static** (mW)	**Total** (mW)
Normal Mode	85.0	72.4	157.4
Low Power Mode	42.0	72.4	114.4
Sleep Mode	15.0	72.4	87.4
Deep Sleep Mode	1.5	72.4	73.9

The power analysis demonstrates significant power savings, with Deep Sleep mode achieving a 53% reduction in total power consumption compared to Normal mode operation.

Chapter 5

Conclusion and Future Work

5.1 Project Summary

This project successfully demonstrates the design, verification, and implementation of a power-aware UART communication system. The final implementation achieves all specified objectives:

- Full-duplex asynchronous communication with configurable parameters [2]
- Comprehensive error detection and handling capabilities
- Intelligent power management with four distinct power modes
- Efficient resource utilization suitable for embedded applications
- Robust verification through extensive simulation testing

5.2 Key Achievements

The project deliverables include:

- A functionally robust UART design meeting all timing requirements
- Effective power reduction through intelligent clock gating and power mode management
- A modular, reusable IP core architecture suitable for integration in larger systems
- Comprehensive documentation and verification methodology

5.3 Future Enhancements

Potential areas for future development include:

- Implementation of FIFO buffers for enhanced data throughput
- Addition of flow control mechanisms (RTS/CTS)
- Support for additional data frame formats (7-bit, 9-bit data)
- Integration with advanced power management standards
- Development of software drivers for embedded processors
- Expansion to support multiple UART channels

5.4 Applications

The developed UART core is suitable for various applications including:

- IoT sensor networks requiring low-power communication
- Battery-powered embedded systems
- Industrial automation and control systems
- Debugging and diagnostic interfaces in digital systems
- Educational platforms for digital design learning

Bibliography

[1] Xilinx Inc., "Vivado Design Suite User Guide: Power Analysis and Optimization (UG907)," v2024.1, 2024. [Online]. Available: https://www.xilinx.com/support/documentation

[2] Texas Instruments, "KeyStone Architecture Universal Asynchronous Receiver/Transmitter (UART) User Guide," Literature Number SPRUGP1, November 2010.

[3] Intel Corporation, "Intel® Stratix® 10 Power Management User Guide: Clock Gating," Version 21.1, 2021. [Online]. Available: https://www.intel.com/content/www/us/en/docs/programmable/683418/21-1/clock-gating.html