



Faculty of Engineering - Ain Shams University

Digital Design – NTI

Project Report

Universal Asynchronous Receiver-Transmitter (UART) with APB Wrapper

Submitted by:

Mohammed Magdy Taher

1. Introduction

The Universal Asynchronous Receiver-Transmitter (UART) is a widely used hardware communication protocol enabling serial data exchange between digital systems. It transmits data one bit at a time, using configurable parameters such as baud rate, parity, stop bits, and data word length.

This project implements:

- **UART Transmitter:** Responsible for serializing and sending data frames.
- **UART Receiver:** Deserializes incoming serial data and validates it.
- **APB UART Wrapper:** Provides a standard AMBA APB interface to control the UART.

The design is verified using testbenches and simulated to validate correct operation.

2. Design Analysis

The UART follows asynchronous communication principles:

- **Transmitter:** Adds start bit, data bits and stop bits.
- **Receiver:** Detects start bit, samples incoming bits, checks stop conditions.
- **APB Wrapper:** Bridges APB bus transactions (write/read) to UART control and data registers.

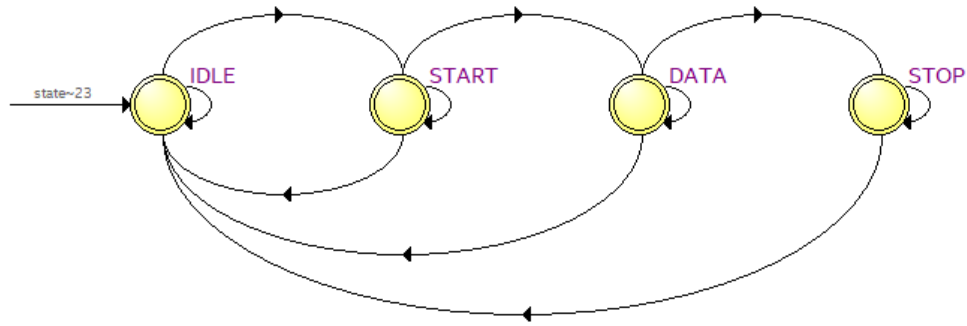
Key Design Parameters:

- Baud rate: Configurable via divider.
- Data width: 8 bits.
- Stop bits: 1.

3. State Diagrams

UART Transmitter FSM

- **IDLE → START → DATA → STOP → IDLE**



UART Receiver FSM

- **IDLE → START DETECT → DATA RECEIVE → STOP CHECK → IDLE**

4. Design Decisions

- Implemented FSM-based control for both TX and RX.
- Chose fixed 8N1 frame format for simplicity.
- APB wrapper provides register-based configuration and status reporting.
- Testbenches are used to automate verification.

5. Verification Strategy

- **Unit-level verification:** Each module (TX, RX, Wrapper) tested independently.
- **Testbenches:** Compare expected vs. actual outputs automatically.
- **Integration testing:** Verified APB transactions trigger correct UART operations.
- **Corner cases:** Tested back-to-back transmission, framing errors, and buffer overflows.

6. Simulation Results

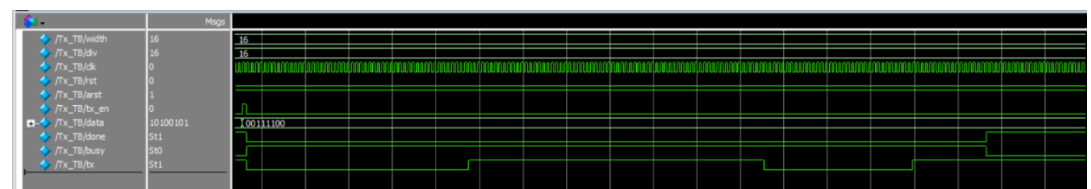
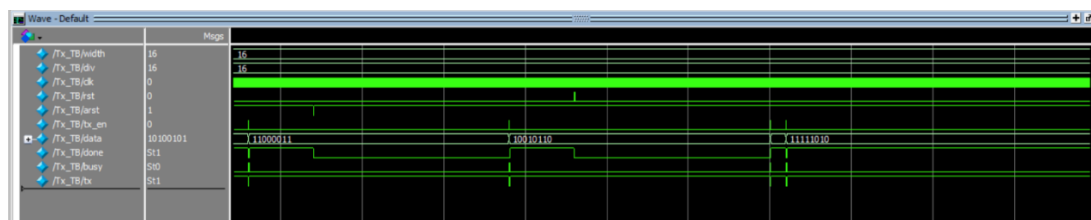
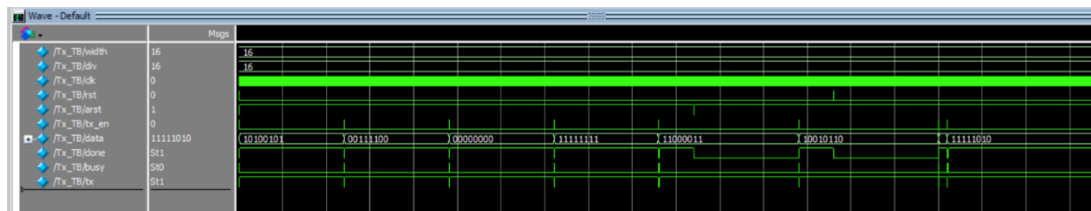
6.1 UART Transmitter (TX)

- Verified correct generation of start, data, and stop bits.
- Observed **serial waveform** on TX line in ModelSim

```

#
# --- Scenario 1: Normal transmission ---
# [155000] TX Start: a5
# [1855000] TX Done : a5
#
# --- Scenario 2: Normal transmission ---
# [1201865000] TX Start: 3c
# [1201865000] TX Done : 3c
#
# --- Scenario 3: All zeros ---
# [2401875000] TX Start: 00
# [2401875000] TX Done : 00
#
# --- Scenario 4: All ones ---
# [3601885000] TX Start: ff
# [3601885000] TX Done : ff
#
# --- Scenario 5: Async reset during transmission ---
# [4801895000] TX Start: c3
# [5201895000] ASYNC RESET asserted!
# [5202095000] ASYNC RESET deasserted!
#
# --- Scenario 6: Sync reset during transmission ---
# [6402205000] TX Start: 96
# [6802205000] SYNC RESET asserted!
# [6802405000] SYNC RESET deasserted!
#
# --- Scenario 7: Overlapping transmissions ---
# [8002415000] TX Start: 12
# [8102425000] Attempted overlapping TX: fa (should be ignored)
# [8102425000] TX Done : 12
#
# --- Simulation finished ---

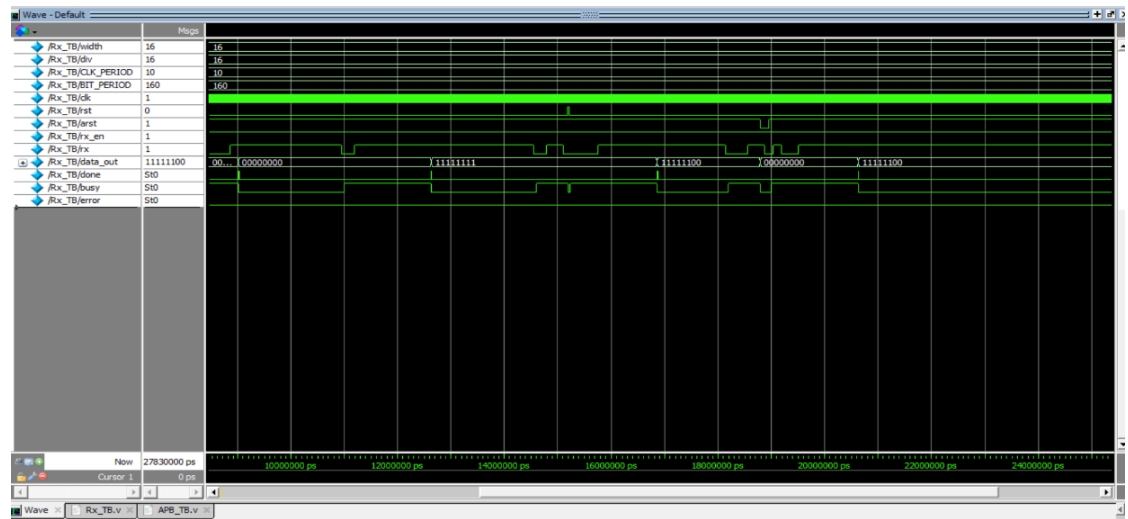
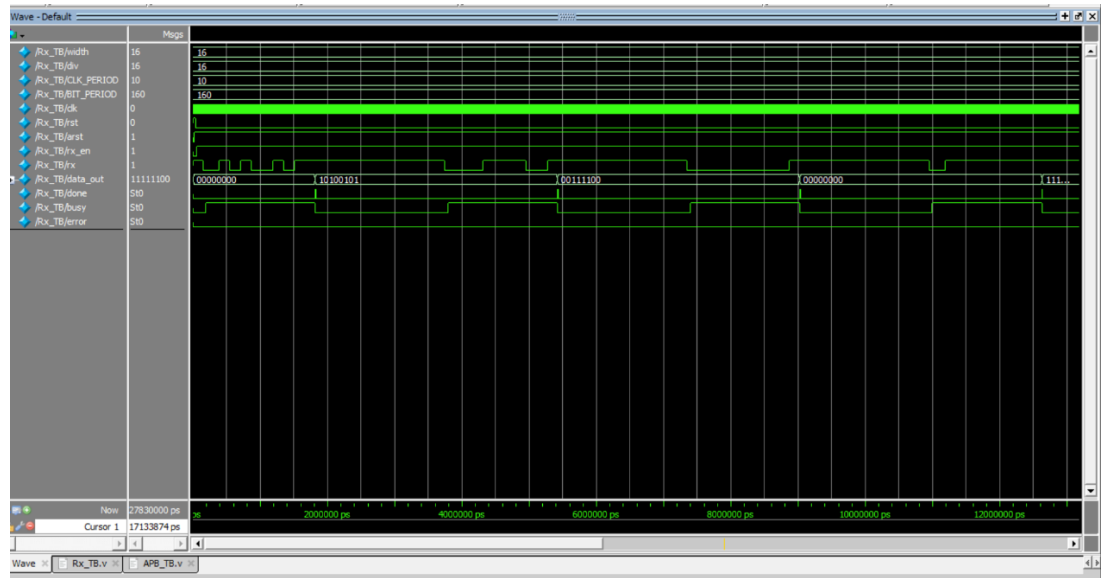
```



6.2 UART Receiver (RX)

- Correctly detected start bit, sampled 8 data bits, and validated stop bit.
- Compared received data against transmitted data

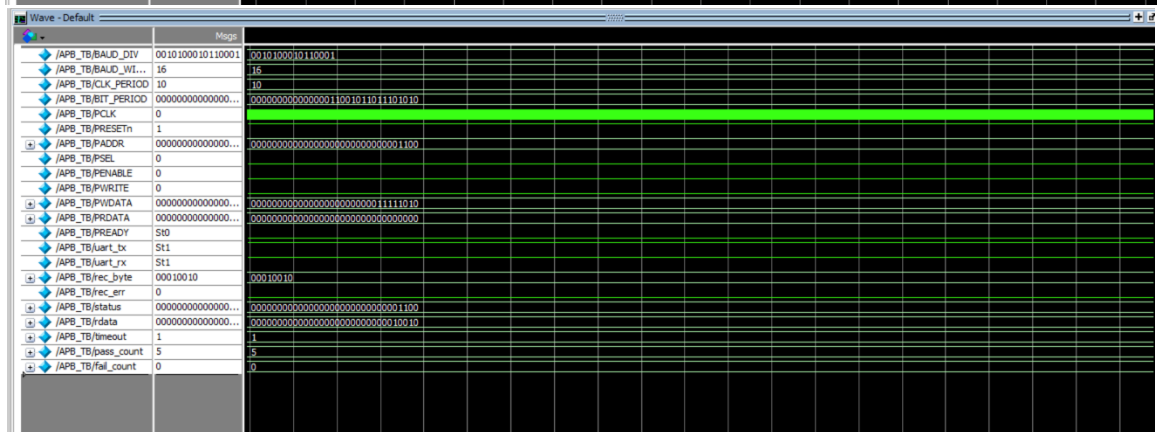
```
#
# --- Scenario 1: Normal RX ---
# [150000] Sending UART byte: a5
# [1830000] RX Done: Got=a5 Expected=a5 PASS
#
# --- Scenario 2: Normal RX ---
# [3750000] Sending UART byte: 3c
# [5430000] RX Done: Got=3c Expected=3c PASS
#
# --- Scenario 3: All zeros ---
# [7350000] Sending UART byte: 00
# [9030000] RX Done: Got=00 Expected=00 PASS
#
# --- Scenario 4: All ones ---
# [10950000] Sending UART byte: ff
# [12630000] RX Done: Got=ff Expected=ff PASS
#
# --- Scenario 5: Soft reset mid-frame ---
# [14550000] Sending UART byte: c3
# [15190000] SYNC RESET asserted!
# [15210000] SYNC RESET deasserted!
#
# --- Scenario 6: Hard reset mid-frame ---
# [18150000] Sending UART byte: 96
# [18790000] ASYNC RESET asserted!
# [18950000] ASYNC RESET deasserted!
#
# --- Simulation finished ---
```



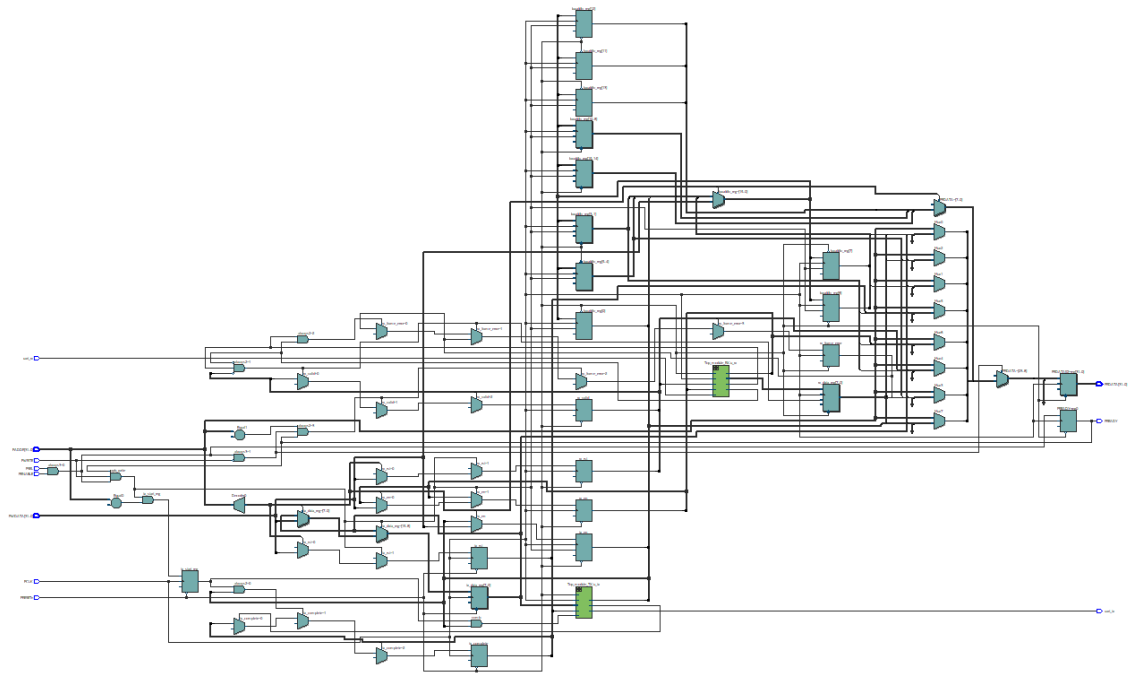
6.3 APB Wrapper

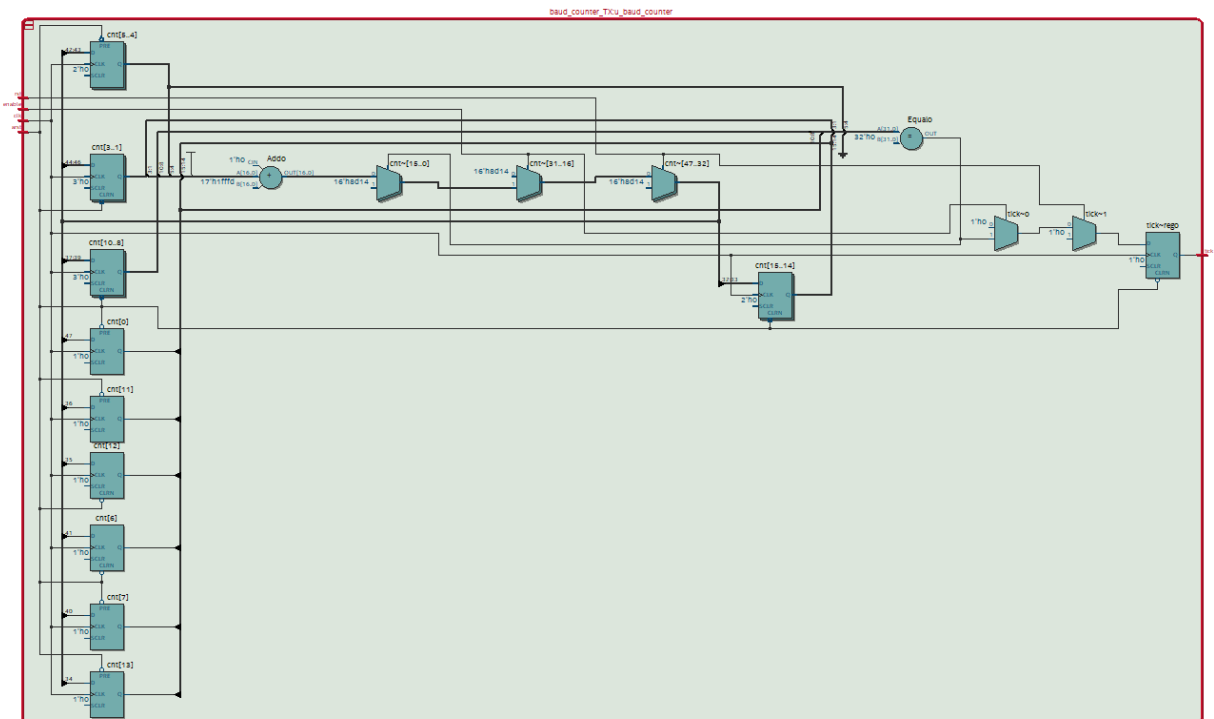
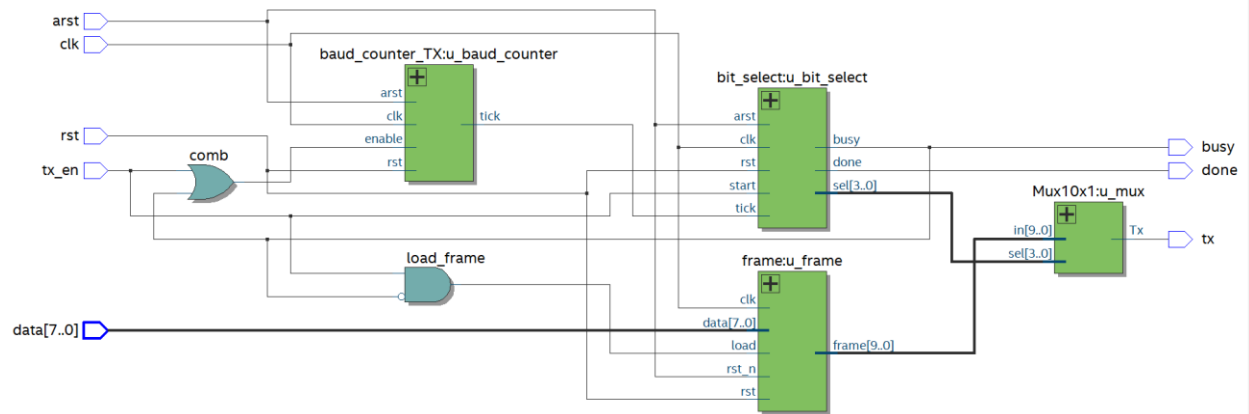
- Verified APB write/read transactions triggered UART operations.

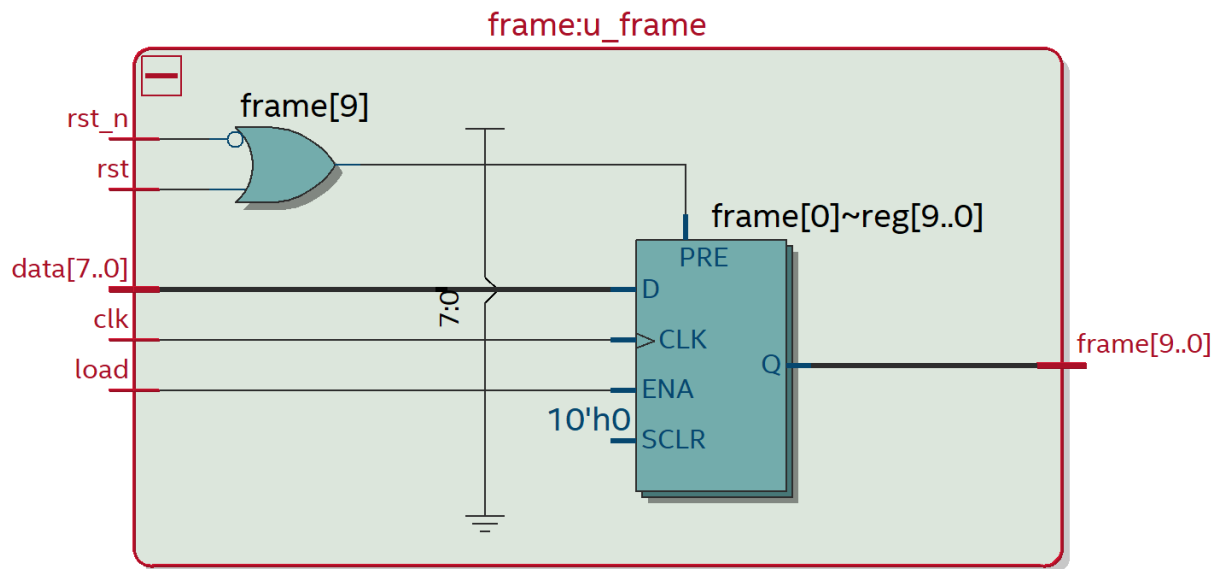
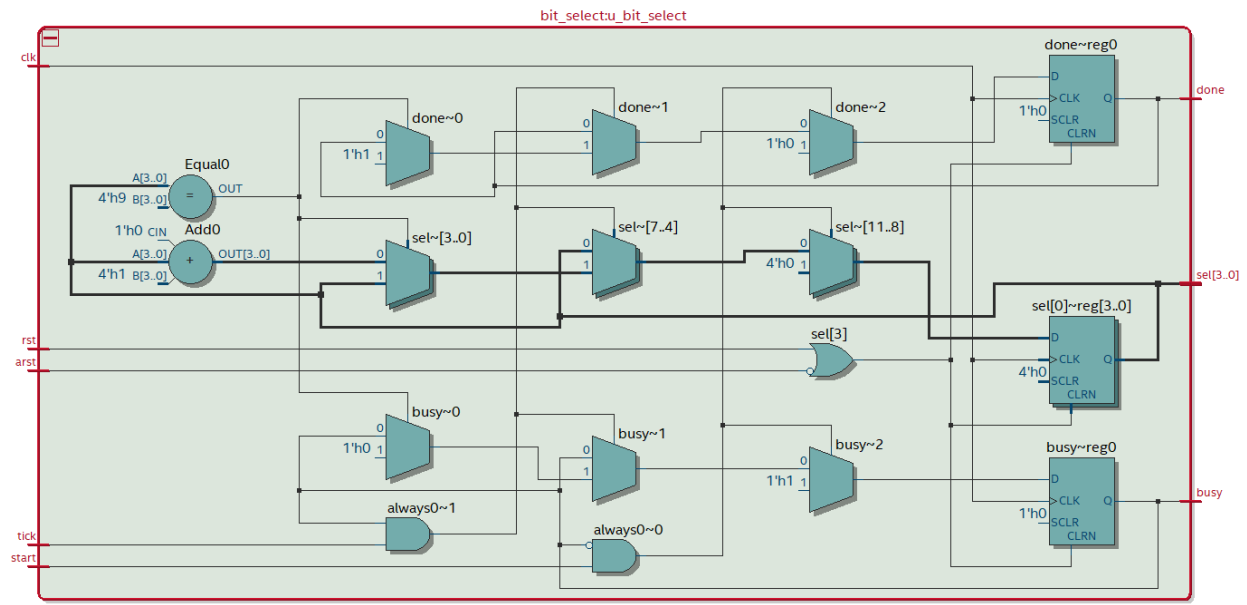
```
VSIM 63> run -all
# [285000] STATUS Read: PRDATA=00000000, rx_valid=0
# [335000] TX Start: tx_data_reg=a5
# [2025000] RX Done: rx_data_wire=a5, rx_error=0
# [2025000] Setting rx_valid=1
# [1250395000] STATUS Read: PRDATA=0000000c, rx_valid=1
# [1250435000] Clearing rx_valid on RXDATA read
# [1250435000] Scenario 1 Sent=a5 Rec=a5 PASS
# [2500505000] STATUS Read: PRDATA=00000004, rx_valid=0
# [2500555000] TX Start: tx_data_reg=3c
# [2502245000] RX Done: rx_data_wire=3c, rx_error=0
# [2502245000] Setting rx_valid=1
# [3750615000] STATUS Read: PRDATA=0000000c, rx_valid=1
# [3750655000] Clearing rx_valid on RXDATA read
# [3750655000] Scenario 2 Sent=3c Rec=3c PASS
# [5000725000] STATUS Read: PRDATA=00000004, rx_valid=0
# [5000775000] TX Start: tx_data_reg=00
# [5002465000] RX Done: rx_data_wire=00, rx_error=0
# [5002465000] Setting rx_valid=1
# [6250835000] STATUS Read: PRDATA=0000000c, rx_valid=1
# [6250875000] Clearing rx_valid on RXDATA read
# [6250875000] Scenario 3 Sent=00 Rec=00 PASS
# [7500945000] STATUS Read: PRDATA=00000004, rx_valid=0
# [7500995000] TX Start: tx_data_reg=ff
# [7502685000] RX Done: rx_data_wire=ff, rx_error=0
# [7502685000] Setting rx_valid=1
# [8751055000] STATUS Read: PRDATA=0000000c, rx_valid=1
# [8751095000] Clearing rx_valid on RXDATA read
# [8751095000] Scenario 4 Sent=ff Rec=ff PASS
# [10001165000] STATUS Read: PRDATA=00000004, rx_valid=0
# [10001215000] TX Start: tx_data_reg=12
# [10002905000] RX Done: rx_data_wire=12, rx_error=0
# [10002905000] Setting rx_valid=1
# [10313745000] STATUS Read: PRDATA=0000000c, rx_valid=1
# [10313795000] TX Start: tx_data_reg=fa
# [10315485000] RX Done: rx_data_wire=fa, rx_error=0
# [11563855000] STATUS Read: PRDATA=0000000c, rx_valid=1
# [11563895000] Clearing rx_valid on RXDATA read
# [11563895000] Scenario 5 Sent=12 Rec=12 PASS
# -----
# TEST SUMMARY: PASS=5 FAIL=0
# -----
```



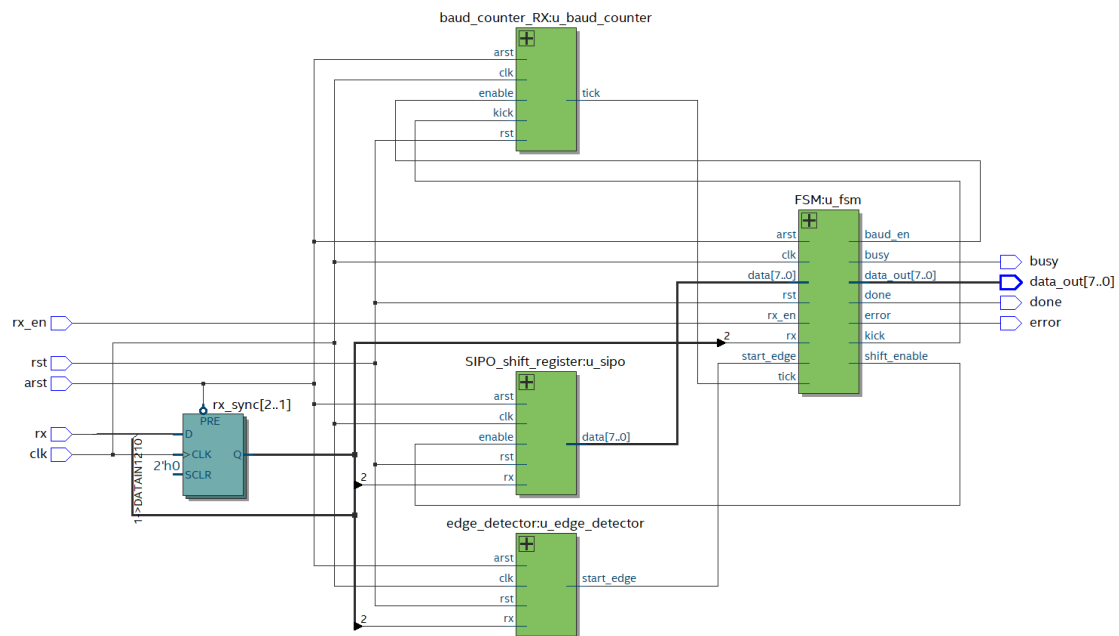
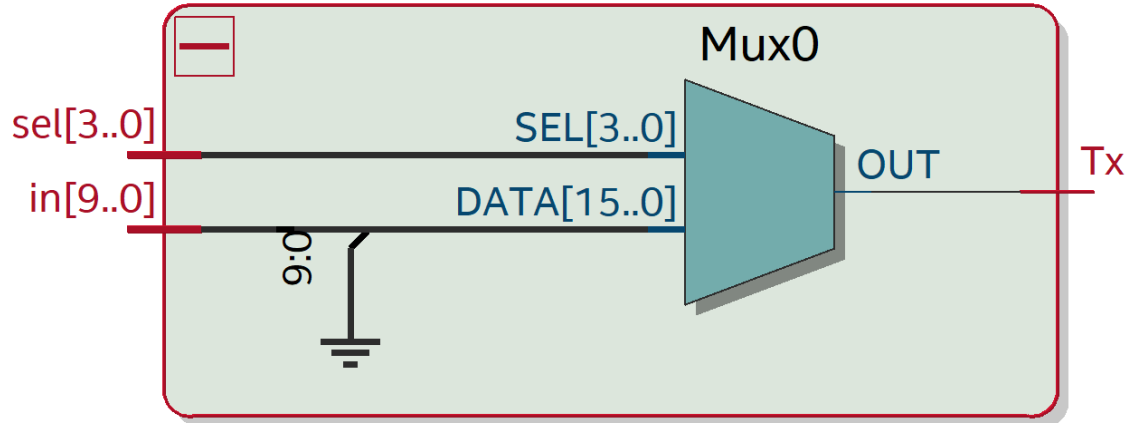
6.4 Quartus Prime Results

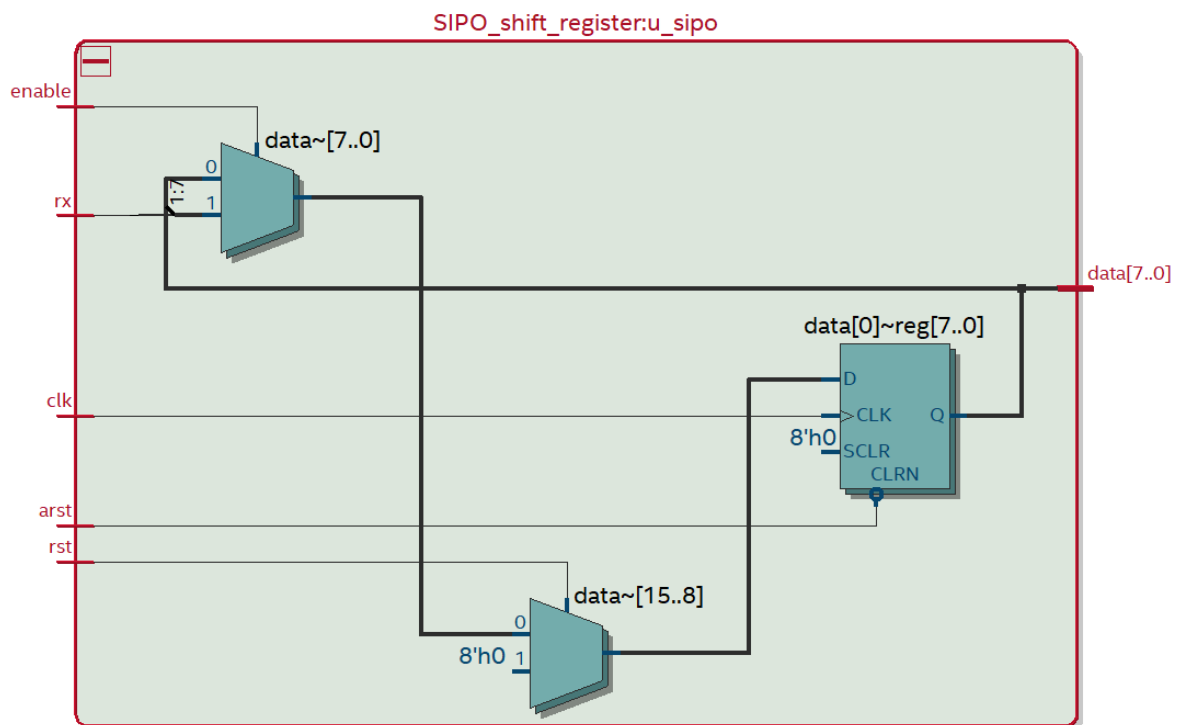
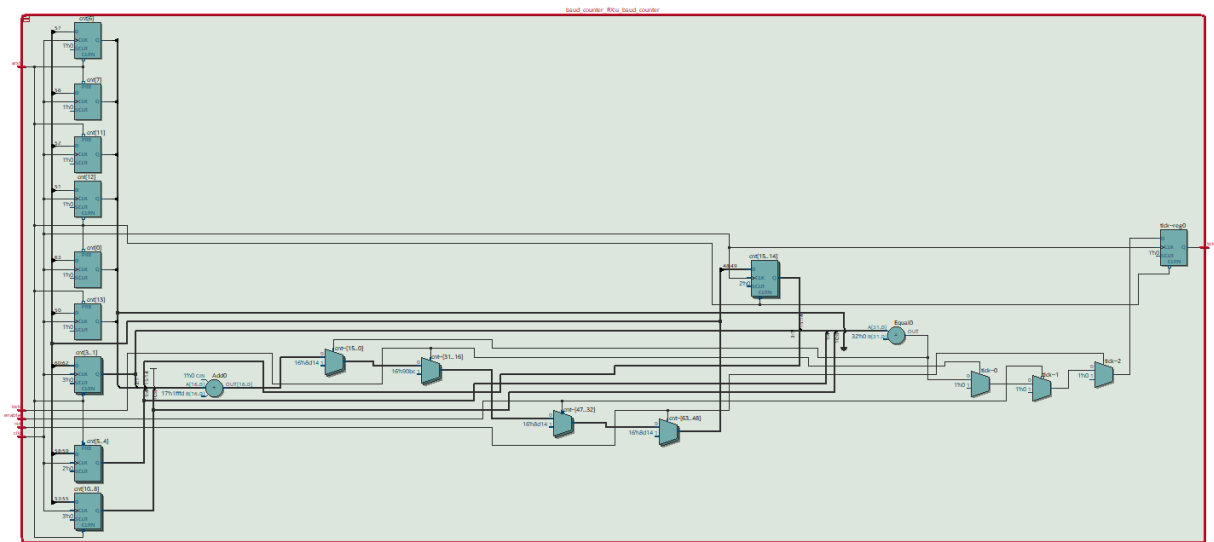


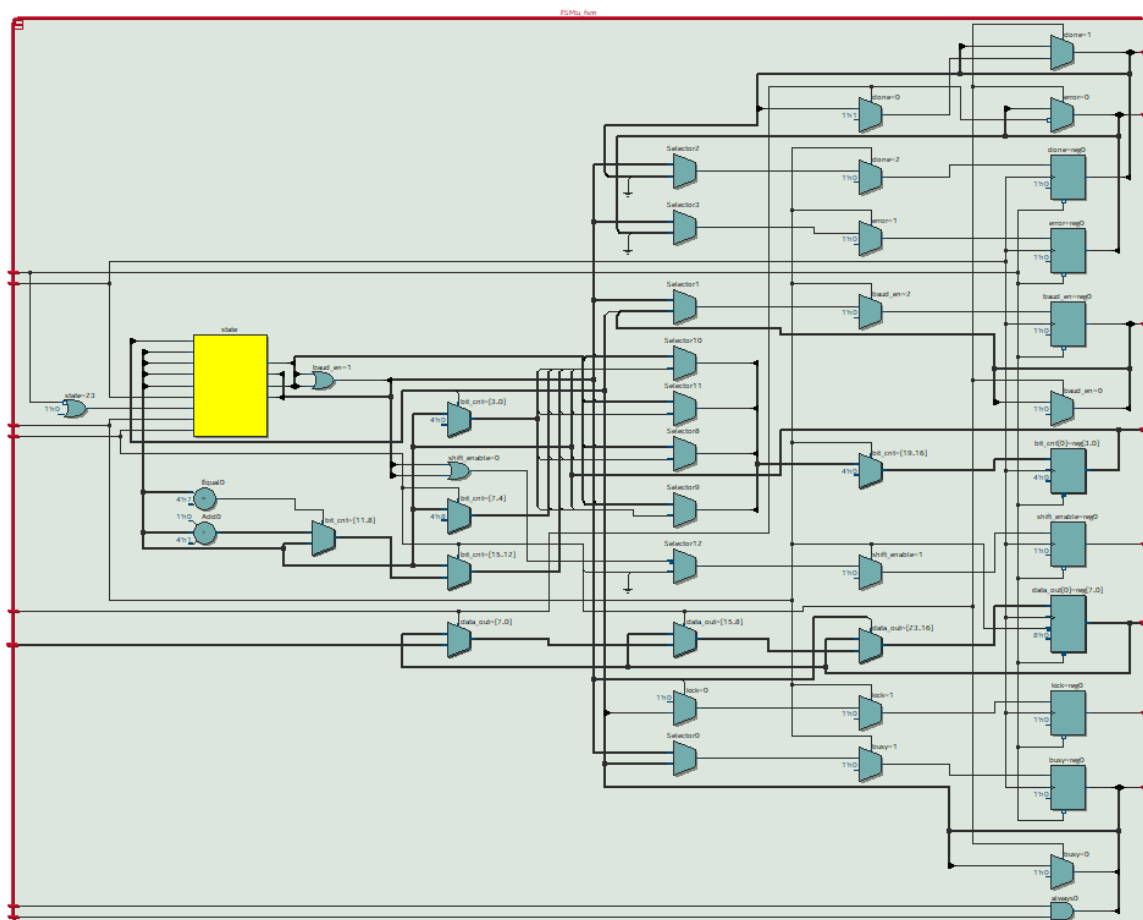
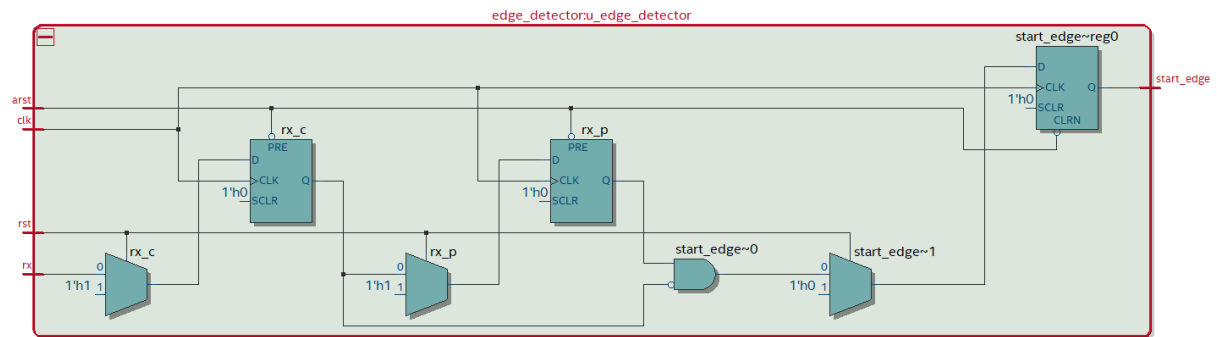




Mux10x1:u_mux







7. Conclusion

The ModelSim simulation results confirm that:

- **UART TX** outputs correct bitstream.
- **UART RX** successfully reconstructs transmitted data.
- **APB wrapper** correctly manages register access.

Quartus Prime synthesis further confirms the design's **hardware feasibility**, showing reasonable resource usage and timing closure.