

# Final System

## SYS CTRL States:

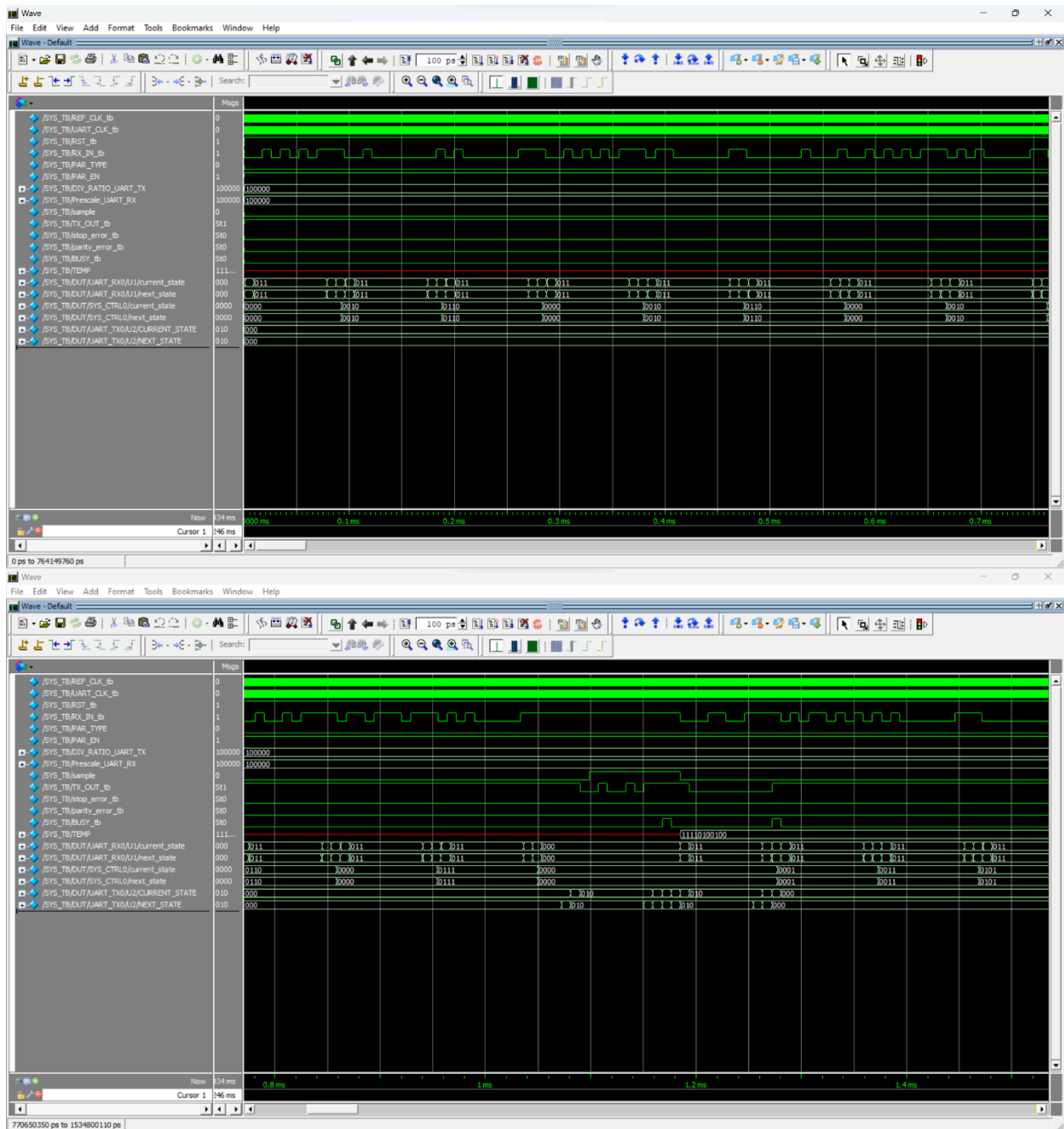
```
////////////////////////////////// FSM Gray Encoding ////////////////////////////////////  
  
localparam IDLE = 4'b0000;  
localparam OPERAND_A_state = 4'b0001;  
localparam OPERAND_B_state = 4'b0011;  
localparam RF_Wr_Addr_state = 4'b0010;  
localparam RF_Wr_Data_state = 4'b0110;  
localparam RF_Rd_Addr_state = 4'b0111;  
localparam ALU_FUN_state = 4'b0101;  
localparam send_ALU_output_least_state = 4'b0100;  
localparam send_ALU_output_most_state = 4'b1100;  
localparam receive_data_reg_file_state = 4'b1110;
```

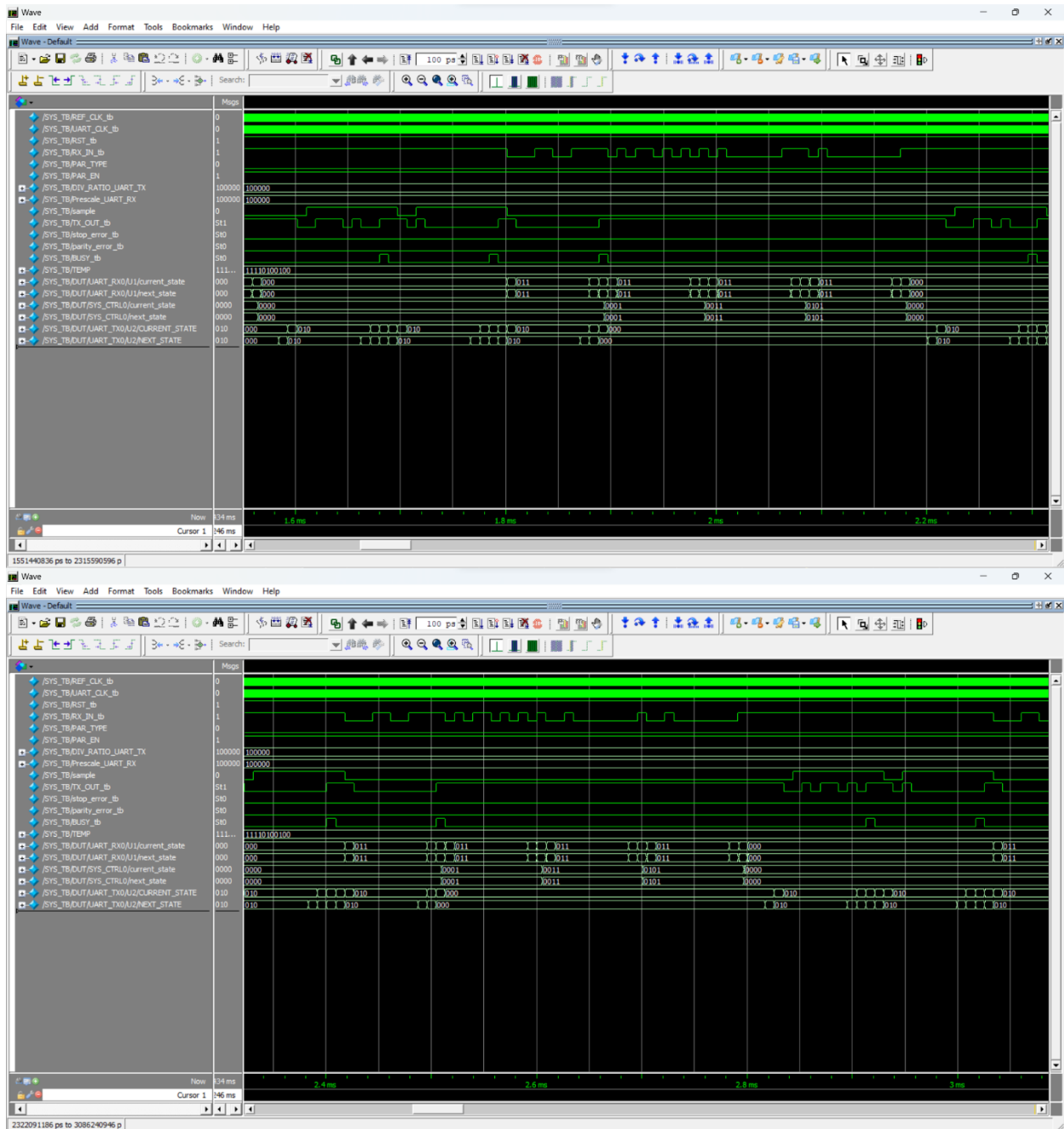
## UART RX states:

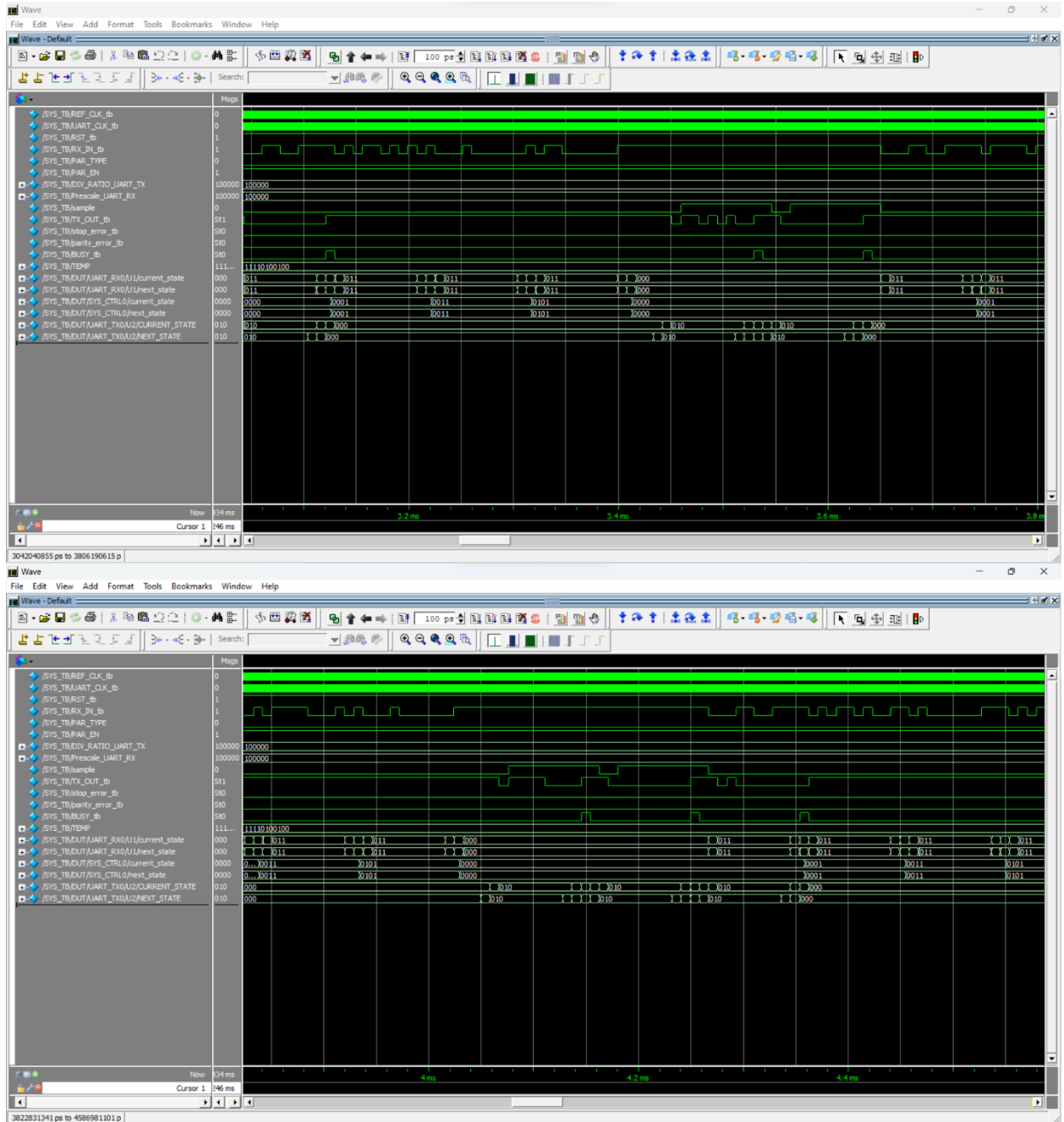
```
//////////////////////////////////state Gray encoding//////////////////////////////////  
localparam IDLE = 3'b000;  
localparam strt_bit_state = 3'b001;  
localparam data_bits_state = 3'b011;  
localparam stp_bit_state = 3'b010;  
localparam parity_bit_state = 3'b110;
```

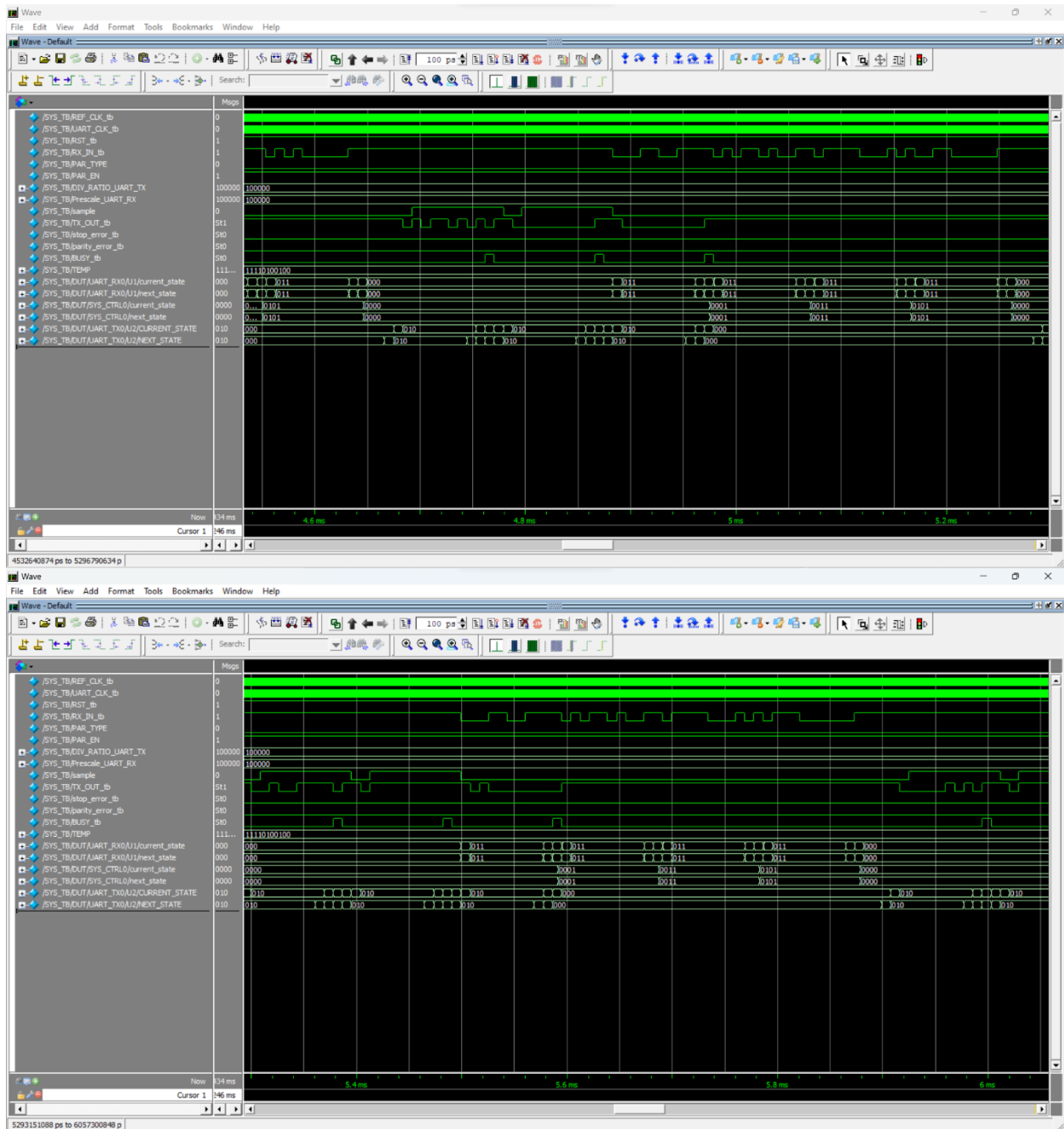
## UART TX state:

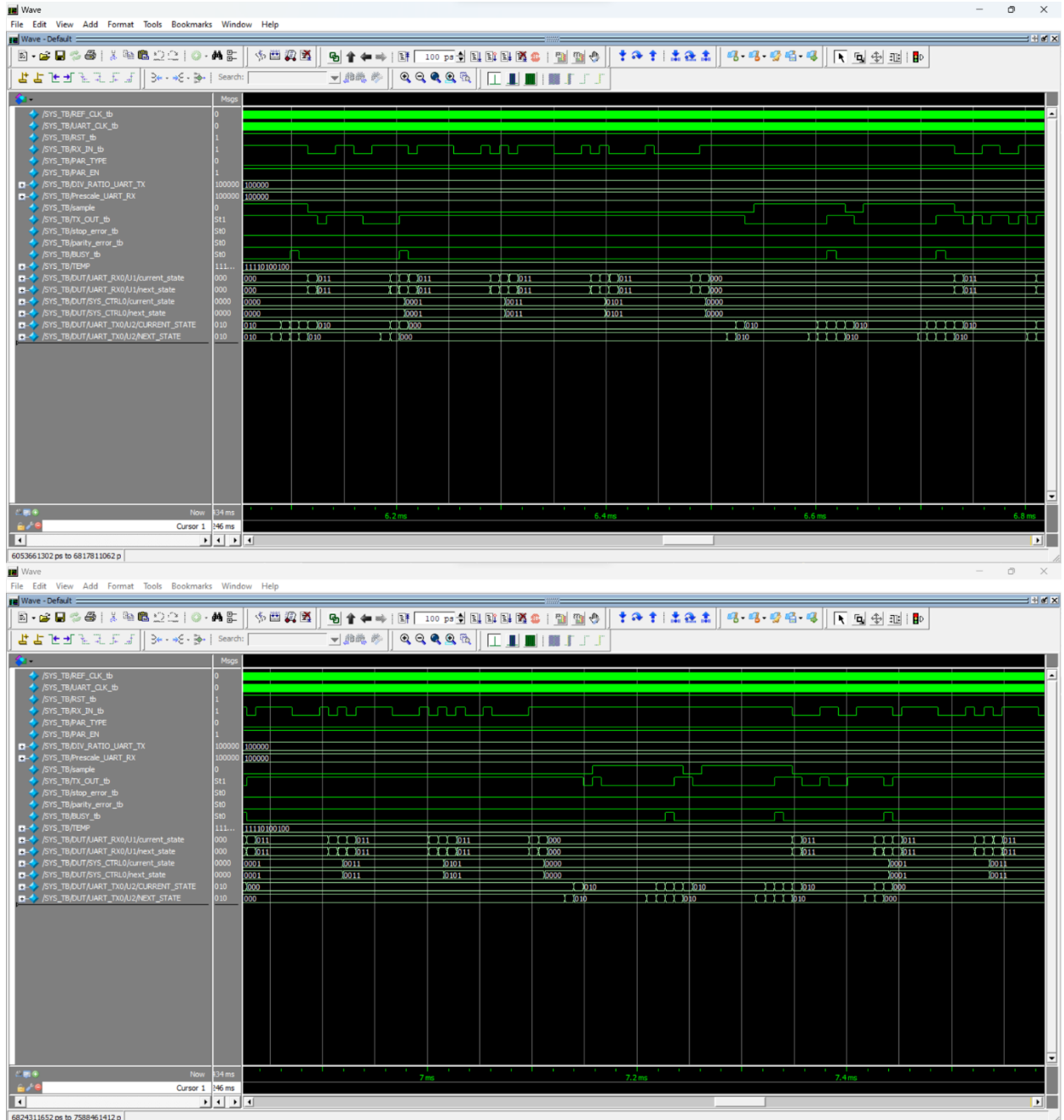
```
//FSM DECODING  
localparam IDLE= 3'b000;  
localparam start_bit_state= 3'b001;  
localparam data_bits_state= 3'b010;  
localparam stop_bit_state= 3'b011;  
localparam parity_bit_state= 3'b111;
```

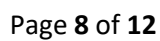




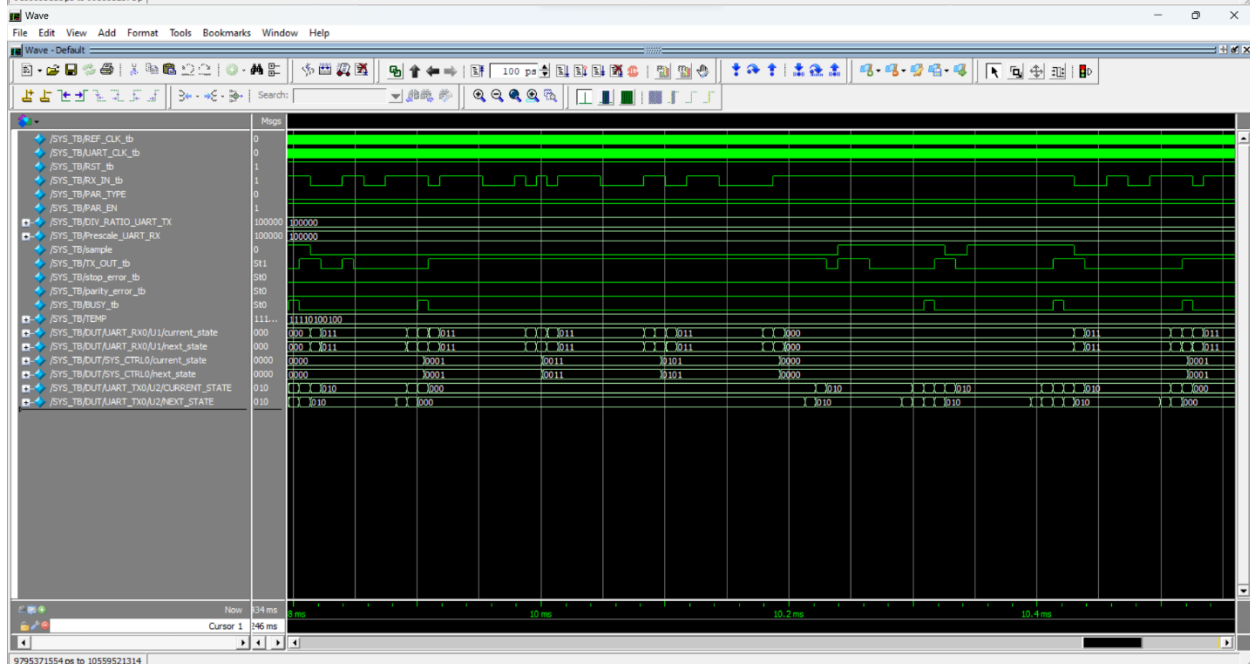


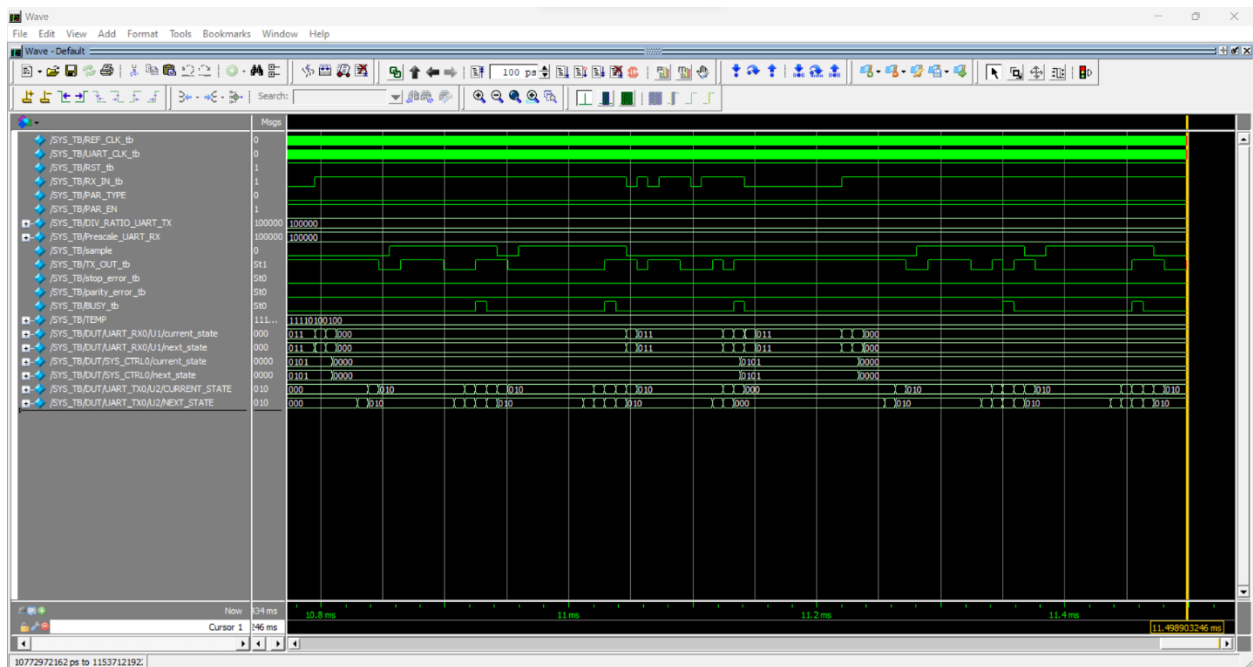
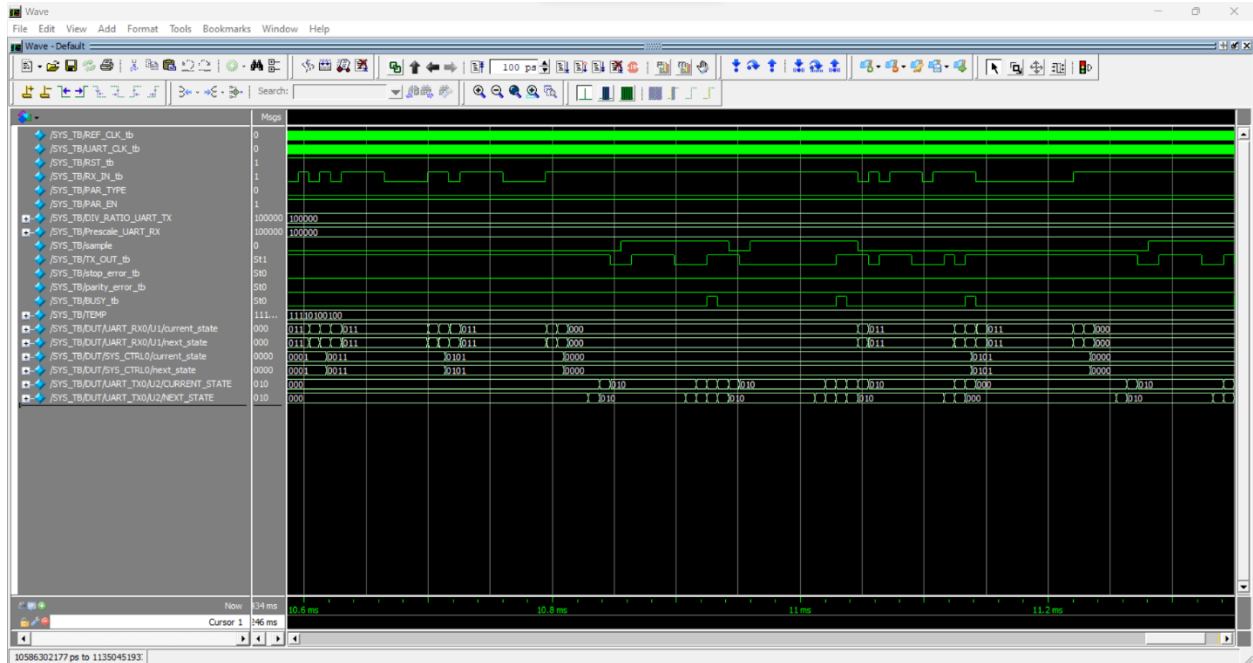








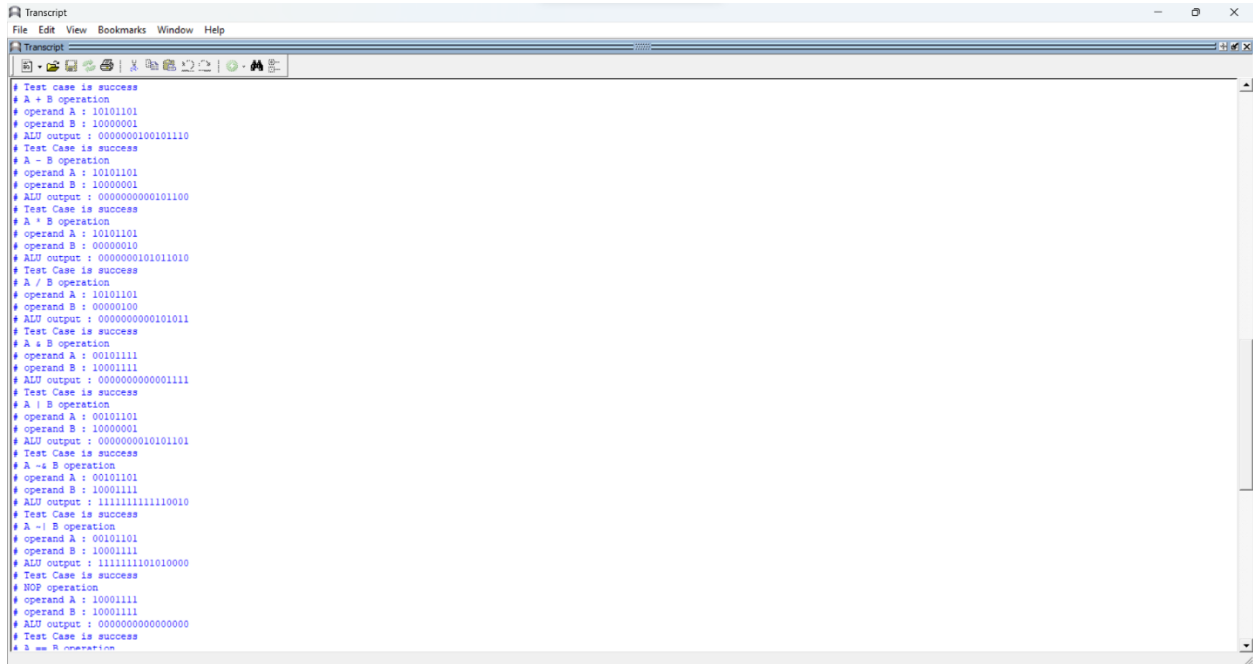




# Test Bench:

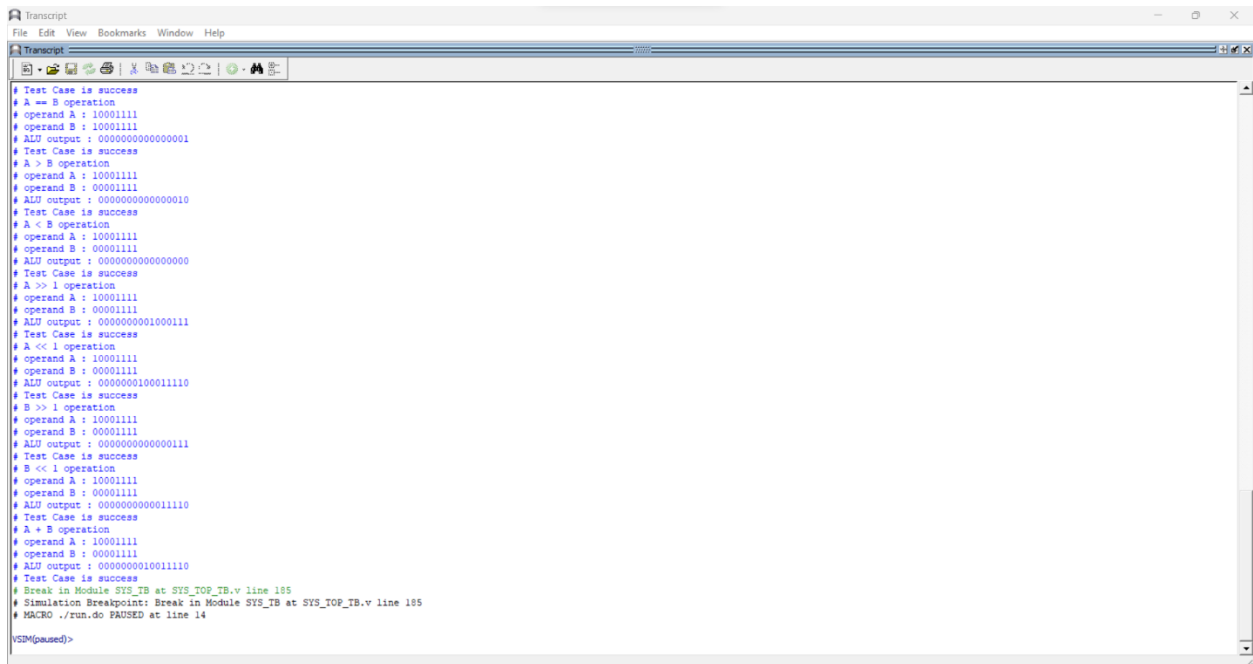
Test cases:

1. Write in register File
2. Read form Register file
3. All ALU operation
4. ALU operation with no operands



```
Transcript
File Edit View Bookmarks Window Help

# Test case is success
# A + B operation
operand A : 10101101
operand B : 10000001
ALU output : 0000000100101110
# Test Case is success
# A - B operation
operand A : 10101101
operand B : 10000001
ALU output : 0000000000101100
# Test Case is success
# A * B operation
operand A : 10101101
operand B : 00000010
ALU output : 0000000101011010
# Test Case is success
# A / B operation
operand A : 10101101
operand B : 00000100
ALU output : 0000000000101011
# Test Case is success
# A & B operation
operand A : 00101111
operand B : 10001111
ALU output : 0000000000001111
# Test Case is success
# A | B operation
operand A : 00101101
operand B : 10000001
ALU output : 000000010101101
# Test Case is success
# A ~ B operation
operand A : 00101101
operand B : 10001111
ALU output : 111111111110010
# Test Case is success
# A ~| B operation
operand A : 00101101
operand B : 10001111
ALU output : 1111111101010000
# Test Case is success
# NOR operation
operand A : 10001111
operand B : 10001111
ALU output : 0000000000000000
# Test Case is success
# s.a B operation
```



```
Transcript
File Edit View Bookmarks Window Help

# Test Case is success
# A == B operation
operand A : 10001111
operand B : 10001111
ALU output : 0000000000000001
# Test Case is success
# A > B operation
operand A : 10001111
operand B : 00001111
ALU output : 0000000000000010
# Test Case is success
# A < B operation
operand A : 10001111
operand B : 00001111
ALU output : 0000000000000000
# Test Case is success
# A >> 1 operation
operand A : 10001111
operand B : 00001111
ALU output : 0000000001000111
# Test Case is success
# A << 1 operation
operand A : 10001111
operand B : 00001111
ALU output : 000000010001110
# Test Case is success
# B >> 1 operation
operand A : 10001111
operand B : 00001111
ALU output : 0000000000000011
# Test Case is success
# B << 1 operation
operand A : 10001111
operand B : 00001111
ALU output : 0000000000001110
# Test Case is success
# A + B operation
operand A : 10001111
operand B : 00001111
ALU output : 000000010001110
# Test Case is success
# Break in Module SYS_TB at SYS_TOP_TB.v line 185
# Simulation Breakpoint: Break in Module SYS_TB at SYS_TOP_TB.v line 185
# MACRO ./run.do PAUSED at line 14

VSDM(paused)
```

