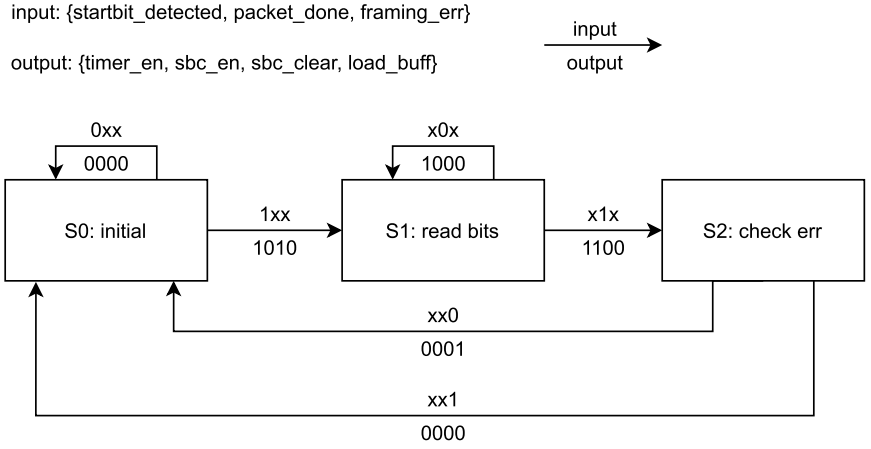
RCU state diagram

input: {startbit_detected, packet_done, framing_err}

output: {timer_en, sbc_en, sbc_clear, load_buff}



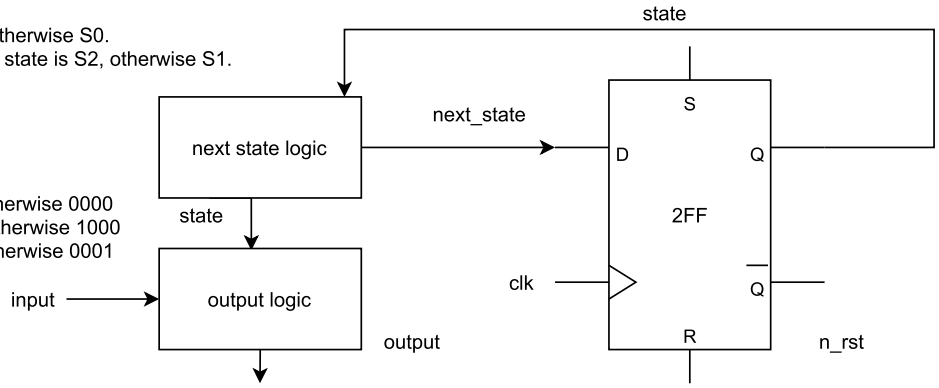
RCU RTL Diagram.

input: {startbit_detected, packet_done, framing_err}

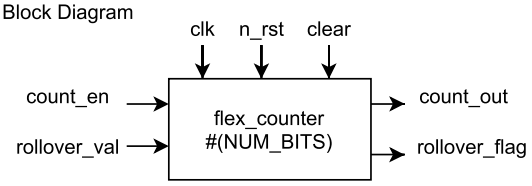
output: {timer_en, sbc_en, sbc_clear, load_buff}

next state logic:
if S==S0, start bit detected, next state is S1, otherwise S0.
else if S==S1, packet transmission done, next state is S2, otherwise S1.
else if S==S2, next state is S0

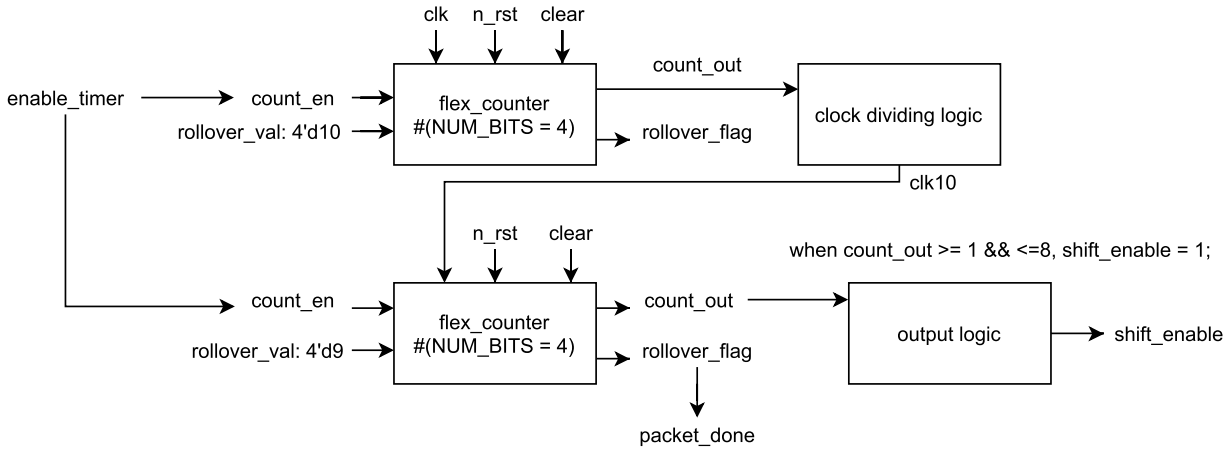
output logic:
if S==S0, start bit detected, output= 1010, otherwise 0000
else if S == S1, packet_done, output 1100, otherwise 1000
else if S==S2, framming_err, output 0000, otherwise 0001

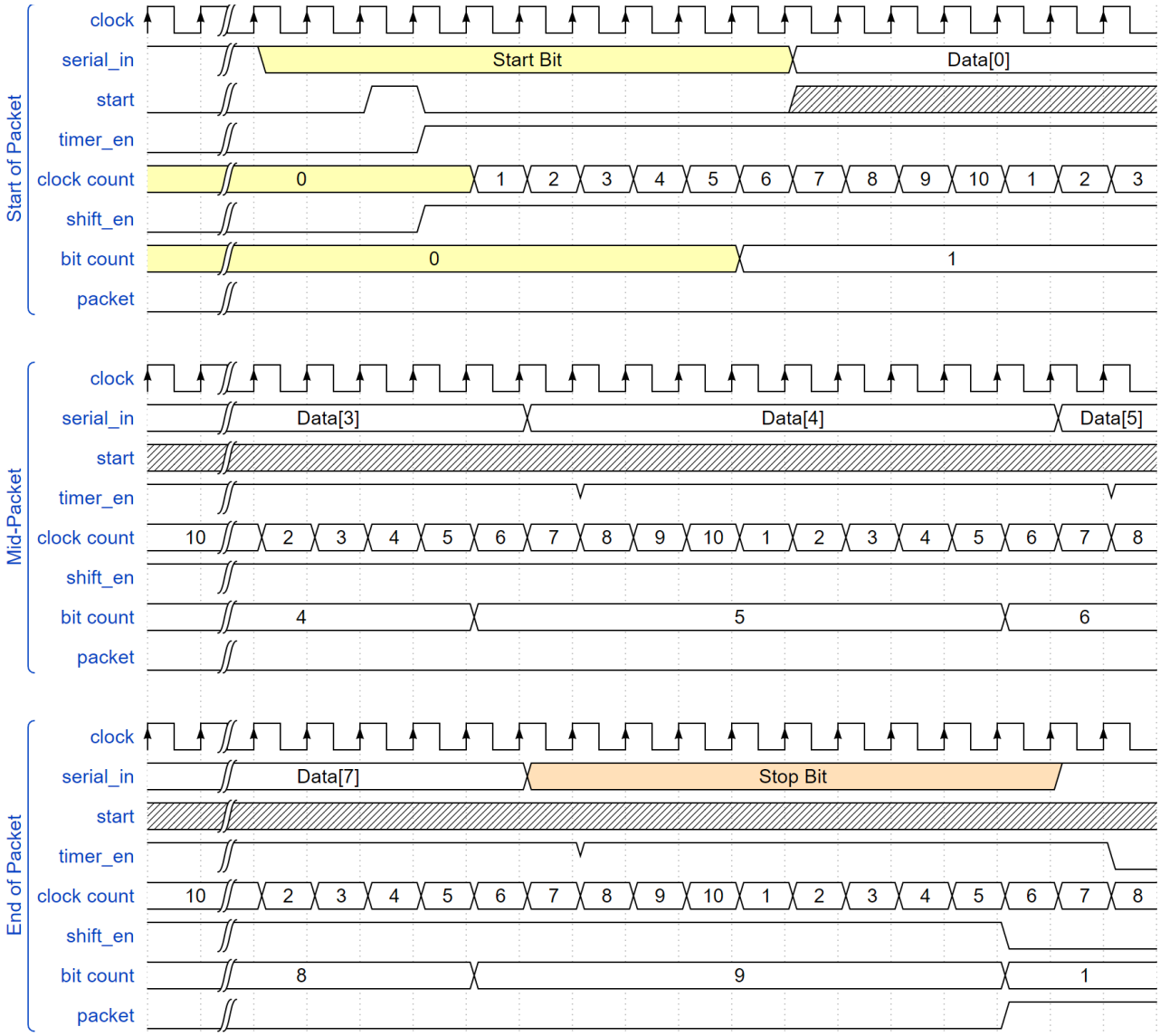


Timer Block Diagram



clock divide logic:
count_out < 5: clk = 1'b1
count_out > 5: clk = 1'b0




Start of Packet
clock
serial_in
start
timer_en
clock count
shift_en
bit count
packet
Start Bit
Data[0]
0 1 2 3 4 5 6 7 8 9 10 1 2 3
0
1
Mid-Packet
Data[3]
Data[4]
Data[5]
10 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8
4
5
6
End of Packet
Data[7]
Stop Bit
10 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8
8
9
1