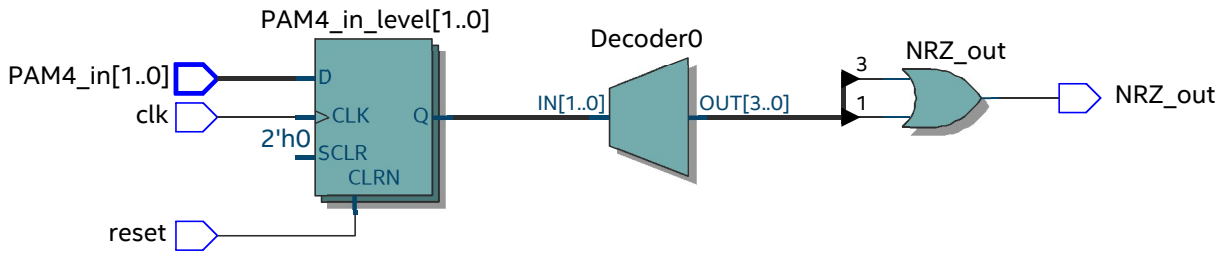
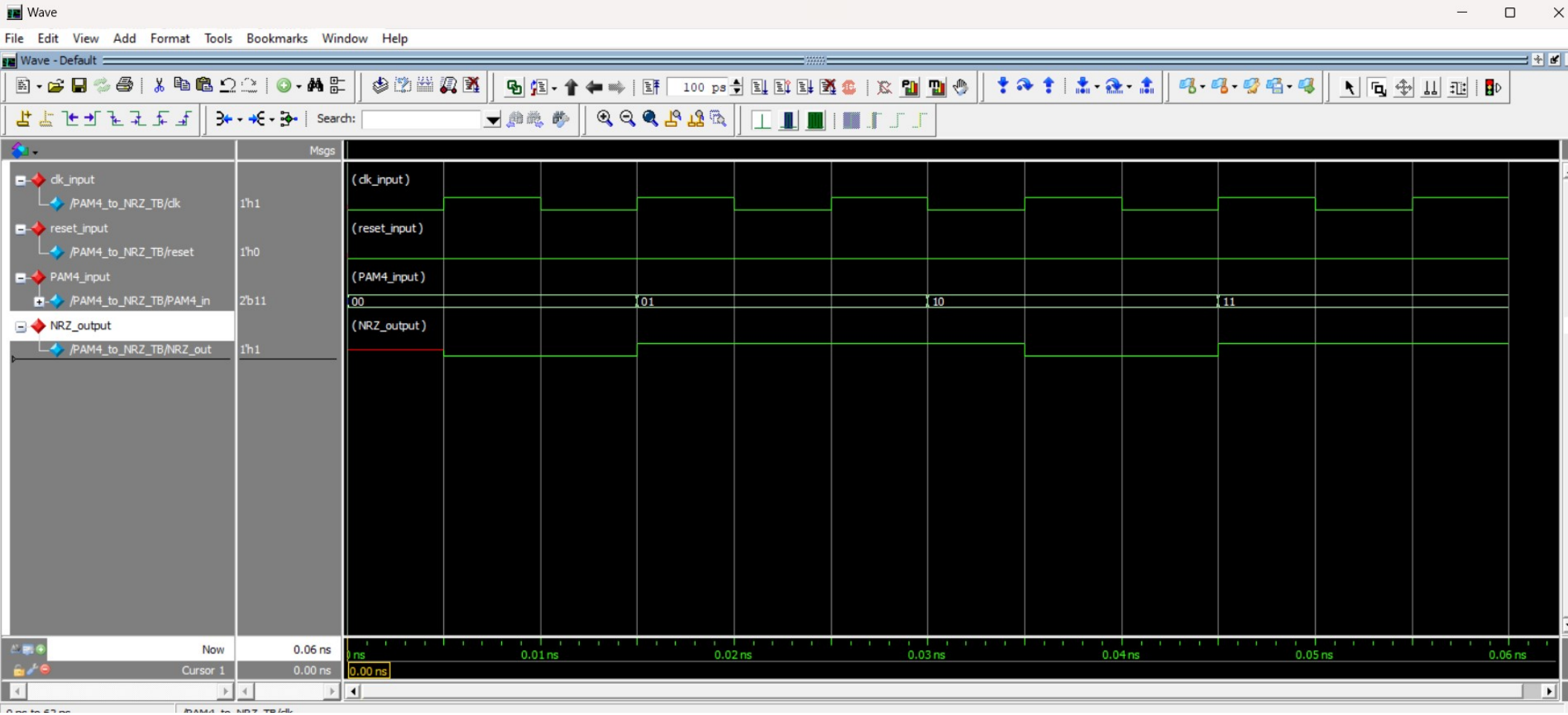
```
/*--------------------------------------------------------------------------------
Name Lamin Jammeh
CLass: EE417 Summer 2024
Lesson 04 HW Question 5 Q2 Part1
Group: Ron Kalin/ Lamin Jammeh
Project Description: the portion takes in PAM4_in with 2 bits input and 4 different
levels and converts it to NRZ
--------------------------------------------------------------------------------*/

module PAM4_to_NRZ (
                     NRZ_out,
                     PAM4_in,
                     clk,
                     reset
                     );
//assigen the inputs and putputs as registers and wires
input wire clk, reset;           //clk and reset signal
input wire [1:0] PAM4_in;           //2 bits PAM4_in input signal
output reg NRZ_out;

//4 different PAM4_in levels
reg [1:0] PAM4_in_level;

//diffrent PAM4_in states
parameter S0 = 2'b00;
parameter S1 = 2'b01;
parameter S2 = 2'b10;
parameter S3 = 2'b11;

//sequential logic updating the state
always @ (posedge clk or posedge reset)      //asynchronous reset
   if (reset) PAM4_in_level <= S0;
   else       PAM4_in_level <= PAM4_in;

 // Next state logic and output logic
    always @(*) begin
        case (PAM4_in_level)
            S0:   NRZ_out = 1'b0;
            S1:   NRZ_out = 1'b1;
            S2:   NRZ_out = 1'b0;
            S3:   NRZ_out = 1'b1;
        endcase
    end

endmodule
```


PAM4_in_level[1..0]
PAM4_in[1..0]
clk
2'h0
reset
D
CLK
Q
SCLR
CLRN
Decoder0
IN[1..0]
OUT[3..0]
3
1
NRZ_out
NRZ_out

```
/*--------------------------------------------------------------------------------
Name Lamin Jammeh
CLass: EE417 Summer 2024
Lesson 04 HW Question 5 Q2 Part1
Group: Ron Kalin/ Lamin Jammeh
Project Description: test bench for part1
--------------------------------------------------------------------------------*/

module PAM4_to_NRZ_TB ();

//define the registers and wires
reg clk, reset;
reg [1:0] PAM4_in;
wire NRZ_out;

//define the unit under test UUT
PAM4_to_NRZ UUT (
                 .NRZ_out(NRZ_out),
                 .PAM4_in(PAM4_in),
                 .clk(clk),
                 .reset(reset)
                 );

//instantiate the clk signal
initial
   begin
      clk = 1'b0;
      forever #5 clk = ~clk;     //10ns clk period
   end

//instantiate the reset signal
initial
   begin
           reset = 1'b0;       //togel the reset signal on
           PAM4_in  = 2'b00;
      #100 reset = 1'b1;      //toggle the reset signal off
   end

//instantiate all the posibble states for PAM4_in with time intervals
initial
   begin
      PAM4_in = 2'b00; #15;
      PAM4_in = 2'b01; #15;
      PAM4_in = 2'b10; #15;
      PAM4_in = 2'b11; #15;

      $stop;
   end

//display the results
initial begin
  $display("PAM4_in--------Binary/NRZ_out");
  $monitor("%b           %b ",PAM4_in, NRZ_out);
 end
endmodule
```


Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Search:
Msgs
clk_input
/PAM4_to_NRZ_TB/clk
1'h1
reset_input
/PAM4_to_NRZ_TB/reset
1'h0
PAM4_input
/PAM4_to_NRZ_TB/PAM4_in
2'b11
NRZ_output
/PAM4_to_NRZ_TB/NRZ_out
1'h1
(clk_input)
(reset_input)
(PAM4_input)
00
01
10
11
(NRZ_output)
Now
0.06 ns
Cursor 1
0.00 ns
0.00 ns
0.01 ns
0.02 ns
0.03 ns
0.04 ns
0.05 ns
0.06 ns

Output Table

```
# PAM4_in--------Binary/NRZ_out
# 00                    x
# 00                    0
# 01                    1
# 10                    1
# 10                    0
# 11                    1
```

PAM4 to NRZ Summary

- This portion of the design takes in PAM4 data and converts it to NRZ data.
- The PAM4 has four levels of input S0, S1, S2, and S3 corresponding to [00,01,10,11]