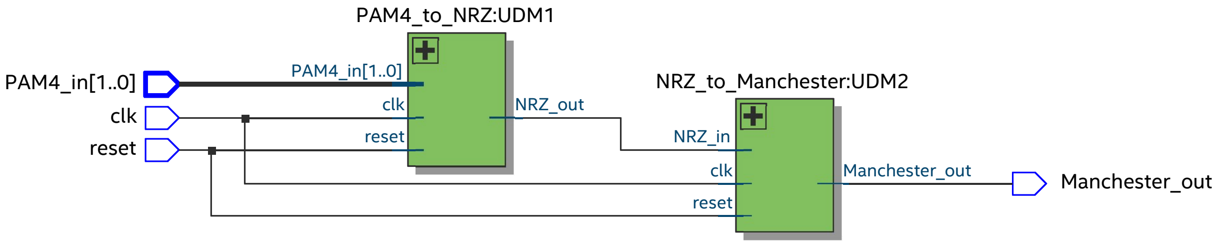
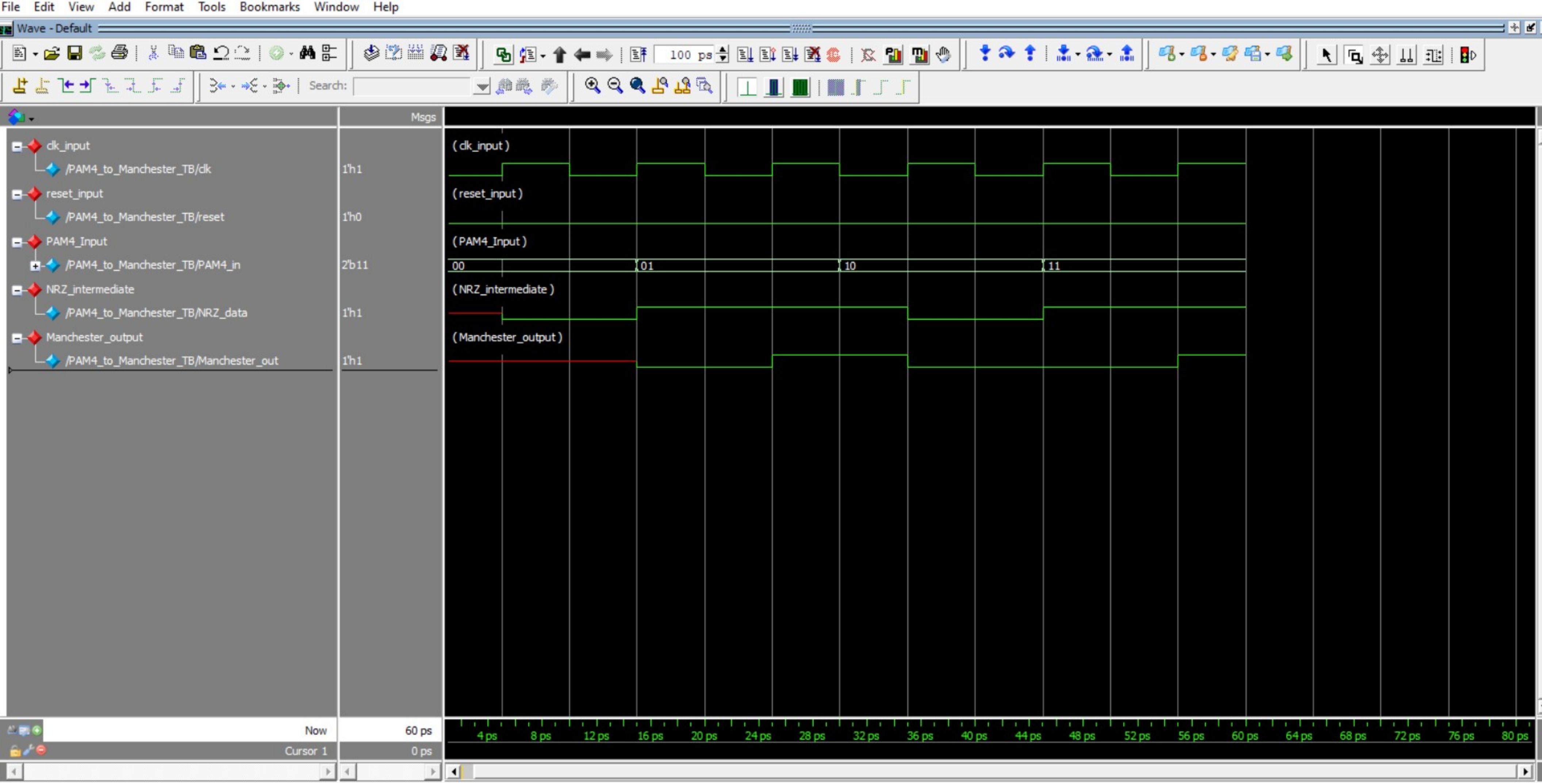
```
module PAM4_to_Manchester (
                           Manchester_out,
                           PAM4_in,
                           clk,
                           reset
                           );

//assigen the inputs and outputs 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 wire       Manchester_out;      //declare as wire since it will be connected to module

//internal probe
wire NRZ_data;             //output signal from the PAM4_to NRZ module

//instantiate the PAM4-NRZ and NRZ-Manchester modules as user define modules (UDM1 and UDM2)
PAM4_to_NRZ UDM1 (
                  .NRZ_out(NRZ_data),                        //note NRZ_out becomes NRZ_data
                  .PAM4_in(PAM4_in),
                  .clk(clk),
                  .reset(reset)
                 );

NRZ_to_Manchester UDM2 (
                        .Manchester_out(Manchester_out),
                        .NRZ_in(NRZ_data),                   //note NRZ_in becomes NRZ_data
                        .clk(clk),
                        .reset(reset)
                       );

endmodule
```

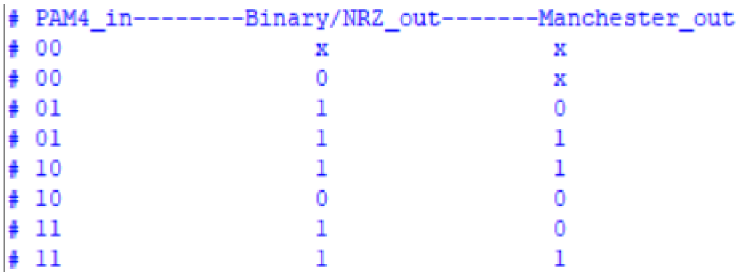

PAM4_to_NRZ:UDM1
PAM4_in[1..0]
PAM4_in[1..0]
clk
clk
reset
reset
NRZ_out
NRZ_to_Manchester:UDM2
NRZ_in
clk
reset
Manchester_out
Manchester_out

```
module PAM4_to_Manchester_TB ();

//define the registers=wire and wires=registers from desing-to-testbench
reg       clk, reset;             //clk and reset signal
reg [1:0] PAM4_in;               //2 bits PAM4_in input signal
wire       Manchester_out;

//define the unit under test UUT
PAM4_to_Manchester UUT (
                        .Manchester_out(Manchester_out),
                        .PAM4_in(PAM4_in),
                        .clk(clk),
                        .reset(reset)
                        );

//monitor internal probe
assign NRZ_data = UUT.UDM1.NRZ_out;

//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-------Manchester_out");
  $monitor("%b          %b           %b ",PAM4_in, NRZ_data,
Manchester_out);
 end
endmodule
```



	Msgs
clk_input	
/PAM4_to_Manchester_TB/clk	1'h1
reset_input	
/PAM4_to_Manchester_TB/reset	1'h0
PAM4_Input	
/PAM4_to_Manchester_TB/PAM4_in	2'b11
NRZ_intermediate	
/PAM4_to_Manchester_TB/NRZ_data	1'h1
Manchester_output	
/PAM4_to_Manchester_TB/Manchester_out	1'h1
Now	60 ps
Cursor 1	0 ps

Output table



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

PAM4 to Manchester out

- This module instantiates 2 different modules
- The final modules take PAM4 data converts the data to NRZ
- The NRZ data is converter to Manchester code as final output