IF : 4.547 | IC Value 80.26

Volume-6, Issue-4, April - 2017 • ISSN No 2277 - 8160

Original Research Paper

Engineering

Ethernet Preamble Detector

Sharat Chandra Musham	Qualcomm Austin
Astha Jain	University of Southern California 9USC), Los Angeles WanderMedia
Abhishek Banerjee	Tata Consultancy Services
Bala Chintamneedi	University of Texas Austin

KEYWORDS : Dexmedetomidine , Intranasal, Midazolam , premedication

Introduction

We implemented HDLC frame detector in our earlier work. HDLC frame detector basically detects the pre-amble, which precedes the frame. The detector output serves as a trigger to the main circuit which processes the packet. HDLC pre-amble is 0x7E. We implemented a state machine to detect the sequence. In project 1, we minimized the asynchronous state machine and derived the Logic equations for each state and the output. We further wrote Verilog code to implement the minimized circuit.

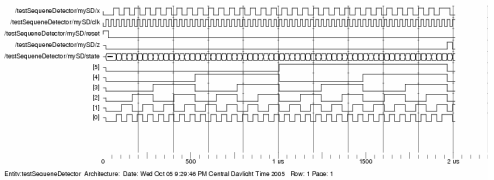
Unfortunately, we couldn't retain the same design, for the reasons obvious. After synthesis, the number of cells reduced drastically and it doesn't fit into these specifications of the current project.

To follow the same lines, we implemented Ethernet Frame detector. Ethernet pre-amble is not only similar to HDLC's, but much wider than HDLC pre-amble. Ethernet pre-amble is 0xaa aa aa aa aa aa aa ab.

Ethernet Pre-amble detector

As mentioned earlier, the Ethernet pre-amble detector detects the sequence of 0xaa aa aa aa aa aa aa ab and provides a trigger to the circuit which processes Ethernet frame. The sequence detector needs to detect 8 byte sequence. We need 64 states to remember the previous states. To represent 64 (26) states, we need 6 D-Flip flops to remember the state.

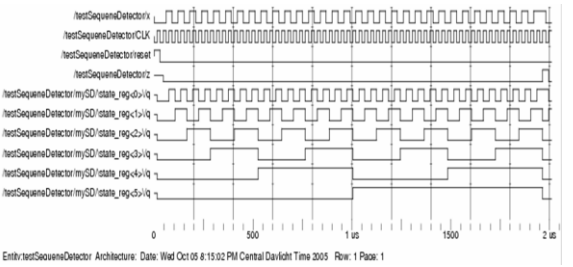
We implemented Verilog code and verified it with a test input that is same as the sequence of interest. We also tested various inputs and noticed the state transitions take place as desired. However, we produced the results that correspond to the sequence of interest. The following is the wave form from ModelSim.



Synthesis of Pre-amble detector

We followed tutorial provided for Synopsys in the course website. We provided the verilog file and synthesized it. The synthesized version consists of 69 cells. We wanted to modify the project to obtain 100 cells, but constrained ourselves to make the modification for final project. We verified that we cover diverse cells available in the cell library so that this project could serve a good example for the coming projects, when we need to develop cell libraries.

We took the verilog file generated by Synoposys and run ModelSim together with header.v file provided in the course website. We verified the result is same as what we obtained before synthesis. The following is the waveform generated after synthesis.



We highlighted some of the output from Synopsys.

The area of the cell structure is

Number of ports: 4
Number of nets: 79
Number of cells: 69
Number of references: 11

Combinational area: 74.000000
Noncombinational area: 49.000000
Net Interconnect area: undefined (No wire load specified)

Total cell area: 123.000000
Total area: undefined

The total Dynamic power dissipated by the cell structure is

Operating Conditions:
Wire Load Model Mode: top

Global Operating Voltage = 5
Power-specific unit information:
Voltage Units = 1V
Capacitance Units = 1.000000pf
Time Units = 1ns
Dynamic Power Units = 1mW (derived from V,C,T units)
Leakage Power Units = Unitless
Cell Internal Power = 0.0000 mW (0%)
Net Switching Power = 253.4641 mW (100%)

Total Dynamic Power = 253.4641 mW (100%)

Cell Leakage Power = 0.0000

Verilog generated by Synopsys

Synopsys mapped the Verilog file we provided as input and converted the high level description about the state machine to logic circuit. The following is the mapped verilog file.

```
module EthPreambleDetector ( x, clk, reset, z );
input x;
input clk;
input reset;
output z;
wire   N392, N393, N394, N395, N396, N397, N398, n9, n10, n11, n12,
n13, n14,
n15, n16, n17, n18, n19, n20, n21, n22, n23, n24, n25, n26, n27, n28,
n29, n30, n31, n32, n33, n34, n35, n36, n37, n38, n39, n40, n41, n42,

n43, n44, n45, n46, n47, n48, n49, n50, n51, n52, n53, n54, n55, n56,
n57, n58, n59, n60, n61, n62, n63, n64, n65, n66, n67, n68, n69, n70,
n71;
wire   [5:0] state;
assign N392 = x;

inv U3 ( .in(n9), .out(n65) );
nand4 U4 ( .a(n10), .b(state[2]), .c(n11), .d(n12), .out(n9) );
nand2 U5 ( .a(n13), .b(n14), .out(n12) );
nand4 U6 ( .a(z), .b(n15), .c(n16), .d(n17), .out(n13) );
inv U7 ( .in(n18), .out(n11) );
aoi12 U8 ( .a(n14), .b(n19), .c(N393), .out(n18) );
nor2 U9 ( .a(reset), .b(n20), .out(N398) );
nor3 U10 ( .a(n21), .b(n22), .c(n23), .out(n20) );
oai22 U11 ( .a(n17), .b(n24), .c(n25), .d(n26), .out(n23) );
aoi12 U12 ( .a(n29), .b(n27), .c(n28), .out(n25) );
aoi12 U13 ( .a(N392), .b(n30), .c(n31), .out(n22) );
nand2 U14 ( .a(n29), .b(n32), .out(n30) );
nor2 U15 ( .a(n17), .b(state[4]), .out(n29) );
oai12 U16 ( .a(n34), .b(N392), .c(n33), .out(n21) );
inv U17 ( .in(n35), .out(N397) );
nand2 U18 ( .a(n36), .b(n15), .out(n35) );
nand4 U19 ( .a(n37), .b(n34), .c(n38), .d(n39), .out(n36) );
nand4 U20 ( .a(n28), .b(state[0]), .c(n40), .d(n41), .out(n39) );
inv U21 ( .in(n42), .out(n38) );
oai22 U22 ( .a(n26), .b(n43), .c(n24), .d(n16), .out(n42) );
nand2 U23 ( .a(N392), .b(n44), .out(n24) );
aoi22 U24 ( .a(n16), .b(n27), .c(n14), .d(n28), .out(n43) );
nand4 U25 ( .a(n19), .b(n10), .c(n40), .d(n41), .out(n34) );
nand2 U26 ( .a(n45), .b(n40), .out(n37) );
nand3 U27 ( .a(n31), .b(n33), .c(n46), .out(n45) );
nand2 U28 ( .a(n28), .b(n32), .out(n46) );
nor2 U29 ( .a(n16), .b(state[5]), .out(n28) );
nand3 U30 ( .a(n19), .b(n10), .c(n47), .out(n33) );
nand2 U31 ( .a(n32), .b(n19), .out(n31) );
nor2 U32 ( .a(n17), .b(n16), .out(n19) );
inv U33 ( .in(state[4]), .out(n16) );
inv U34 ( .in(state[5]), .out(n17) );
aoi12 U35 ( .a(reset), .b(n48), .c(n49), .out(N396) );
oai12 U36 ( .a(n40), .b(n50), .c(n51), .out(n49) );
nor3 U37 ( .a(n14), .b(state[2]), .c(n44), .out(n51) );
inv U38 ( .in(n52), .out(n50) );
aoi22 U39 ( .a(n47), .b(n53), .c(n54), .d(state[3]), .out(n48) );
inv U40 ( .in(n55), .out(N395) );
nand2 U41 ( .a(n15), .b(n56), .out(n55) );
nand3 U42 ( .a(n57), .b(n58), .c(n59), .out(n56) );
inv U43 ( .in(n60), .out(n59) );
nor2 U44 ( .a(N392), .b(n52), .out(n60) );
nand2 U45 ( .a(n27), .b(n32), .out(n52) );
nor2 U46 ( .a(n14), .b(n41), .out(n27) );
inv U47 ( .in(state[3]), .out(n14) );
nand3 U48 ( .a(n47), .b(n40), .c(n32), .out(n58) );
nor2 U49 ( .a(n41), .b(state[3]), .out(n47) );
aoi22 U50 ( .a(state[2]), .b(n54), .c(n41), .d(n53), .out(n57) );
inv U51 ( .in(n26), .out(n53) );
nand2 U52 ( .a(n10), .b(n40), .out(n26) );
nor2 U53 ( .a(n61), .b(n44), .out(n10) );
inv U54 ( .in(state[1]), .out(n61) );
inv U55 ( .in(state[2]), .out(n41) );
nor2 U56 ( .a(n40), .b(state[0]), .out(n54) );
inv U57 ( .in(reset), .out(n15) );
nor3 U58 ( .a(reset), .b(n62), .c(n63), .out(N394) );
aoi12 U59 ( .a(n40), .b(state[1]), .c(n44), .out(n63) );
nor2 U60 ( .a(N392), .b(n32), .out(n62) );
nor2 U61 ( .a(n44), .b(state[1]), .out(n32) );
inv U62 ( .in(state[0]), .out(n44) );
nor2 U63 ( .a(n40), .b(reset), .out(N393) );
inv U64 ( .in(N392), .out(n40) );
dff z_reg ( .d(n65), .gclk(clk), .rnot(1'b1), .q(z) );
dff \state_reg<0> ( .d(N393), .gclk(clk), .rnot(1'b1), .q(state[0]) );
dff \state_reg<1> ( .d(N394), .gclk(clk), .rnot(1'b1), .q(state[1]) );
dff \state_reg<2> ( .d(N395), .gclk(clk), .rnot(1'b1), .q(state[2]) );
dff \state_reg<3> ( .d(N396), .gclk(clk), .rnot(1'b1), .q(state[3]) );
dff \state_reg<4> ( .d(N397), .gclk(clk), .rnot(1'b1), .q(state[4]) );
dff \state_reg<5> ( .d(N398), .gclk(clk), .rnot(1'b1), .q(state[5]) );
endmodule
```

Report from Synopsys showing Cell count

As mentioned earlier in the description, the whole Ethernet Preamble detector synthesized to consist 69 cells. The following is the report from Synopsys.

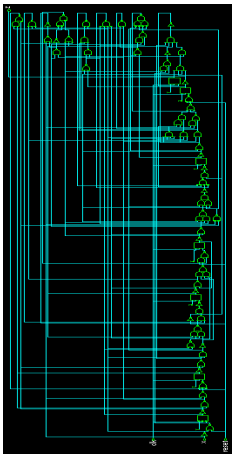
```
Report : cell
Design : EthPreambleDetector
Version: V-2003.12
Date   : Wed Oct  5 21:09:37 2005

Attributes:
b - black box (unknown)
h - hierarchical
n - noncombinational
r - removable
u - contains unmapped logic

Cell  Reference  Library  Area  Attributes

                U3                inv          library
1.000000
                U4                nand4        library
1.000000
                U5                nand2        library
1.000000
                U6                nand4        library
1.000000
                U7                inv          library
1.000000
                U8          aoi12              library
2.000000
U9          nor2     library     1.000000
                U10               nor3         library
1.000000
                U11               oai22        library
2.000000
                U12               aoi12        library
2.000000
                U13               aoi12        library
2.000000
                U14               nand2        library
1.000000
                U15               nor2         library
1.000000
                U16               oai12        library
2.000000
                U17               inv          library
1.000000
                U18               nand2        library
1.000000
                U19               nand4        library
1.000000
                U20               nand4        library
1.000000
                U21               inv          library
1.000000
                U22               oai22        library
2.000000
                U23               nand2        library
1.000000
                U24               aoi22        library
2.000000
                U25               nand4        library
1.000000
                U26               nand2        library
```

1.000000	U27	nand3	library
1.000000	U28	nand2	library
1.000000	U29	nor2	library
1.000000	U30	nand3	library
1.000000	U31	nand2	library
1.000000	U32	nor2	library
1.000000	U33	inv	library
1.000000	U34	inv	library
1.000000	U35	aoi12	library
2.000000	U36	oai12	library
2.000000	U37	nor3	library
1.000000	U38	inv	library
1.000000	U39	aoi22	library
2.000000	U40	inv	library
1.000000	U41	nand2	library
1.000000	U42	nand3	library
1.000000	U43	inv	library
1.000000	U44	nor2	library
1.000000	U45	nand2	library
1.000000	U46	nor2	library
1.000000	U47	inv	library
1.000000	U48	nand3	library
1.000000	U49	nor2	library
1.000000	U50	aoi22	library
2.000000	U51	inv	library
1.000000	U52	nand2	library
1.000000	U53	nor2	library
1.000000	U54	inv	library
1.000000	U55	inv	library
1.000000	U56	nor2	library
1.000000	U57	inv	library
1.000000	U58	nor3	library
1.000000	U59	aoi12	library
2.000000	U60	nor2	library
1.000000	U61	nor2	library
1.000000	U62	inv	library
1.000000	U63	nor2	library
1.000000	U64	inv	library
7.000000 n	state_reg<0>	dff	library
7.000000 n	state_reg<1>	dff	library
7.000000 n	state_reg<2>	dff	library
7.000000 n	state_reg<3>	dff	library
7.000000 n	state_reg<4>	dff	library
7.000000 n	state_reg<5>	dff	library
7.000000 n	z_reg	dff	library

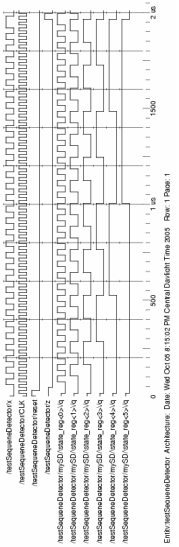
123.000000

Total69cells

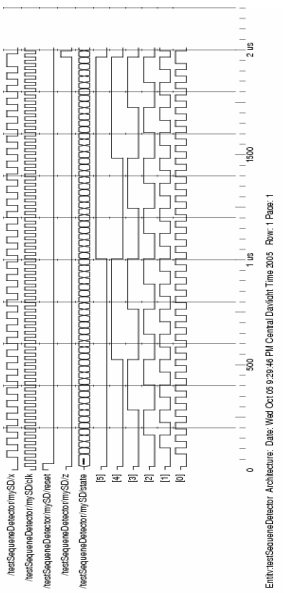
Cell Layout



Waveform generated from Mapped verilog for detailed analysis



Original Waveform without Synthesis for detailed analysis



Behavioral verilog code for our new design

```
module EthPreambleDetector (x, clk, reset, z);
input x,clk,reset;
output z;

reg z;
reg [5:0] state;

parameter state_1 = 6'b00_0000;
parameter state_2 = 6'b00_0001;
parameter state_3 = 6'b00_0010;
parameter state_4 = 6'b00_0011;
parameter state_5 = 6'b00_0100;
parameter state_6 = 6'b00_0101;
parameter state_7 = 6'b00_0110;
parameter state_8 = 6'b00_0111;
parameter state_9 = 6'b00_1000;
parameter state_10 = 6'b00_1001;
parameter state_11 = 6'b00_1010;
parameter state_12 = 6'b00_1011;
parameter state_13 = 6'b00_1100;
parameter state_14 = 6'b00_1101;
parameter state_15 = 6'b00_1110;
parameter state_16 = 6'b00_1111;
parameter state_17 = 6'b01_0000;
parameter state_18 = 6'b01_0001;
parameter state_19 = 6'b01_0010;
parameter state_20 = 6'b01_0011;
parameter state_21 = 6'b01_0100;
parameter state_22 = 6'b01_0101;
parameter state_23 = 6'b01_0110;
parameter state_24 = 6'b01_0111;
parameter state_25 = 6'b01_1000;
parameter state_26 = 6'b01_1001;
parameter state_27 = 6'b01_1010;
parameter state_28 = 6'b01_1011;
parameter state_29 = 6'b01_1100;
parameter state_30 = 6'b01_1101;
parameter state_31 = 6'b01_1110;
parameter state_32 = 6'b01_1111;
parameter state_33 = 6'b10_0000;
parameter state_34 = 6'b10_0001;
parameter state_35 = 6'b10_0010;
parameter state_36 = 6'b10_0011;
parameter state_37 = 6'b10_0100;
parameter state_38 = 6'b10_0101;
parameter state_39 = 6'b10_0110;
parameter state_40 = 6'b10_0111;
parameter state_41 = 6'b10_1000;
parameter state_42 = 6'b10_1001;
parameter state_43 = 6'b10_1010;
parameter state_44 = 6'b10_1011;
parameter state_45 = 6'b10_1100;
parameter state_46 = 6'b10_1101;
parameter state_47 = 6'b10_1110;
parameter state_48 = 6'b10_1111;
parameter state_49 = 6'b11_0000;
parameter state_50 = 6'b11_0001;
parameter state_51 = 6'b11_0010;
parameter state_52 = 6'b11_0011;
parameter state_53 = 6'b11_0100;
parameter state_54 = 6'b11_0101;
parameter state_55 = 6'b11_0110;
parameter state_56 = 6'b11_0111;
parameter state_57 = 6'b11_1000;
parameter state_58 = 6'b11_1001;
parameter state_59 = 6'b11_1010;
parameter state_60 = 6'b11_1011;
parameter state_61 = 6'b11_1100;
parameter state_62 = 6'b11_1101;
parameter state_63 = 6'b11_1110;
parameter state_64 = 6'b11_1111;

always@ (posedge clk)
    begin
    if (reset)
                begin
            state = state_1;
      z = 0;
      end
    else case(state)
        state_1:
          begin
            if (x)
              begin
              state = state_2;
              z = 0;
              end
            else
              begin
              state = state_1;
              z = 0;
              end
          end
        state_2:
          begin
            if (x)
              begin
              state = state_2;
              z = 0;
              end
            else
              begin
              state = state_3;
              z = 0;
              end
          end
        state_3:
          begin
            if (x)
              begin
              state = state_4;
              z = 0;
              end
            else
              begin
              state = state_1;
              z = 0;
              end
          end
        state_4:
          begin
            if (x)
              begin
              state = state_2;
              z = 0;
              end
            else
              begin
              state = state_5;
              z = 0;
              end
          end
        state_5:
          begin
            if (x)
              begin
              state = state_6;
              z = 0;
              end
            else
              begin
              state = state_1;
              z = 0;
              end
          end
        state_6:
          begin
            if (x)
              begin
              state = state_2;
              z = 0;
              end
            else
              begin
              state = state_7;
              z = 0;
              end
          end
        state_7:
          begin
            if (x)
              begin
              state = state_8;
              z = 0;
              end
            else
              begin
              state = state_1;
              z = 0;
              end
          end
        state_8:
          begin
            if (x)
              state = state_2;
            else        state = state_9;
```

```
    end
     state_9:
    begin
      if (x)
        begin
        state = state_10;
        z = 0;
        end
      else
        begin
        state = state_1;
        z = 0;
        end
    end
  state_10:
    begin
      if (x)
        begin
        state = state_2;
        z = 0;
        end
      else
        begin
        state = state_11;
        z = 0;
        end
    end
  state_11:
    begin
      if (x)
        begin
        state = state_12;
        z = 0;
        end
      else
        begin
        state = state_1;
        z = 0;
        end
    end
  state_12:
    begin
      if (x)
        begin
        state = state_2;
        z = 0;
        end
      else
        begin
        state = state_13;
        z = 0;
        end
    end
  state_13:
    begin
      if (x)
        begin
        state = state_14;
        z = 0;
        end
      else
        begin
        state = state_1;
        z = 0;
        end
    end
  state_14:
    begin
      if (x)
        begin
        state = state_2;
        z = 0;
        end
      else
        begin
        state = state_15;
        z = 0;
        end
    end
  state_15:
    begin
      if (x)
        begin
        state = state_16;
        z = 0;
        end
      else
        begin
        state = state_1;
        z = 0;
        end
    end
  state_16:
    begin
      if (x)
        begin
        state = state_2;
        z = 0;
        end
      else
        begin
        state = state_17;
        z = 0;
        end
    end
  state_17:
    begin
      if (x)
        begin
        state = state_18;
        z = 0;
        end
      else
        begin
        state = state_1;
        z = 0;
        end
    end
  state_18:
    begin
      if (x)
        begin
        state = state_2;
        z = 0;
        end
      else
        begin
        state = state_19;
        z = 0;
        end
    end
  state_19:
    begin
      if (x)
        begin
        state = state_20;
        z = 0;
        end
      else
        begin
        state = state_1;
        z = 0;
        end
    end
  state_20:
    begin
      if (x)
        begin
        state = state_2;
        z = 0;
        end
      else
        begin
        state = state_21;
        z = 0;
        end
    end
  state_21:
    begin
      if (x)
        begin
        state = state_22;
        z = 0;
        end
      else
        begin
        state = state_1;
        z = 0;
        end
    end
  state_22:
    begin
      if (x)
        begin
        state = state_2;
        z = 0;
        end
      else
```

```
        begin
        state=state_23;
        z=0;
        end
    end
  state_23:
    begin
      if(x)
        begin
        state=state_24;
        z=0;
        end
      else
        begin
        state=state_1;
        z=0;
        end
    end
  state_24:
    begin
      if(x)
        begin
        state=state_2;
        z=0;
        end
      else
        begin
        state=state_25;
        z=0;
        end
    end
  state_25:
    begin
      if(x)
        begin
        state=state_26;
        z=0;
        end
      else
        begin
        state=state_1;
        z=0;
        end
    end
  state_26:
    begin
      if(x)
        begin
        state=state_2;
        z=0;
        end
      else
        begin
        state=state_27;
        z=0;
        end
    end
  state_27:
    begin
      if(x)
        begin
        state=state_28;
        z=0;
        end
      else
        begin
        state=state_1;
        z=0;
        end
    end
  state_28:
    begin
      if(x)
        begin
        state=state_2;
        z=0;
        end
      else
        begin
        state=state_29;
        z=0;
        end
    end
  state_29:
    begin
      if(x)
        begin
        state=state_30;
        z=0;
        end
      else
        begin
        state=state_1;
        z=0;
        end
    end
  state_30:
    begin
      if(x)
        begin
        state=state_2;
        z=0;
        end
      else
        begin
        state=state_31;
        z=0;
        end
    end
  state_31:
    begin
      if(x)
        begin
        state=state_32;
        z=0;
        end
      else
        begin
        state=state_1;
        z=0;
        end
    end
  state_32:
    begin
      if(x)
        begin
        state=state_2;
        z=0;
        end
      else
        begin
        state=state_33;
        z=0;
        end
    end
  state_33:
    begin
      if(x)
        begin
        state=state_34;
        z=0;
        end
      else
        begin
        state=state_1;
        z=0;
        end
    end
  state_34:
    begin
      if(x)
        begin
        state=state_2;
        z=0;
        end
      else
        begin
        state=state_35;
        z=0;
        end
    end
  state_35:
    begin
      if(x)
        begin
        state=state_36;
        z=0;
        end
      else
        begin
        state=state_1;
        z=0;
        end
    end
  state_36:
    begin
      if(x)
        begin
```

```
        state=state_2;
        z=0;
        end
      else
        begin
        state=state_37;
        z=0;
        end
    end
  state_37:
    begin
      if(x)
        begin
        state=state_38;
        z=0;
        end
      else
        begin
        state=state_1;
        z=0;
        end
    end
  state_38:
    begin
      if(x)
        begin
        state=state_2;
        z=0;
        end
      else
        begin
        state=state_39;
        z=0;
        end
    end
  state_39:
    begin
      if(x)
        begin
        state=state_40;
        z=0;
        end
      else
        begin
        state=state_1;
        z=0;
        end
    end
  state_40:
    begin
      if(x)
        begin
        state=state_2;
        z=0;
        end
      else
        begin
        state=state_41;
        z=0;
        end
    end
  state_41:
    begin
      if(x)
        begin
        state=state_42;
        z=0;
        end
      else
        begin
        state=state_1;
        z=0;
        end
    end
  state_42:
    begin
      if(x)
        begin
        state=state_2;
        z=0;
        end
      else
        begin
        state=state_43;
        z=0;
        end
    end
  state_43:
    begin
      if(x)
        begin
        state=state_44;
        z=0;
        end
      else
        begin
        state=state_1;
        z=0;
        end
    end
  state_44:
    begin
      if(x)
        begin
        state=state_2;
        z=0;
        end
      else
        begin
        state=state_45;
        z=0;
        end
    end
  state_45:
    begin
      if(x)
        begin
        state=state_46;
        z=0;
        end
      else
        begin
        state=state_1;
        z=0;
        end
    end
  state_46:
    begin
      if(x)
        begin
        state=state_2;
        z=0;
        end
      else
        begin
        state=state_47;
        z=0;
        end
    end
  state_47:
    begin
      if(x)
        begin
        state=state_48;
        z=0;
        end
      else
        begin
        state=state_1;
        z=0;
        end
    end
  state_48:
    begin
      if(x)
        begin
        state=state_2;
        z=0;
        end
      else
        begin
        state=state_49;
        z=0;
        end
    end
  state_49:
    begin
      if(x)
        begin
        state=state_50;
        z=0;
        end
      else
        begin
        state=state_1;
        z=0;
        end
    end
```

```
state_50:
  begin
    if(x)
      begin
      state=state_2;
      z=0;
      end
    else
      begin
      state=state_51;
      z=0;
      end
  end
state_51:
  begin
    if(x)
      begin
      state=state_52;
      z=0;
      end
    else
      begin
      state=state_1;
      z=0;
      end
  end
state_52:
  begin
    if(x)
      begin
      state=state_2;
      z=0;
      end
    else
      begin
      state=state_53;
      z=0;
      end
  end
state_53:
  begin
    if(x)
      begin
      state=state_54;
      z=0;
      end
    else
      begin
      state=state_1;
      z=0;
      end
  end
state_54:
  begin
    if(x)
      begin
      state=state_2;
      z=0;
      end
    else
      begin
      state=state_55;
      z=0;
      end
  end
state_55:
  begin
    if(x)
      begin
      state=state_56;
      z=0;
      end
    else
      begin
      state=state_1;
      z=0;
      end
  end
state_56:
  begin
    if(x)
      begin
      state=state_2;
      z=0;
      end
    else
      begin
      state=state_57;
      z=0;
      end
  end
state_57:
  begin
    if(x)
      begin
      state=state_58;
      z=0;
      end
    else
      begin
      state=state_1;
      z=0;
      end
  end
state_58:
  begin
    if(x)
      begin
      state=state_2;
      z=0;
      end
    else
      begin
      state=state_59;
      z=0;
      end
  end
state_59:
  begin
    if(x)
      begin
      state=state_60;
      z=0;
      end
    else
      begin
      state=state_1;
      z=0;
      end
  end
state_60:
  begin
    if(x)
      begin
      state=state_2;
      z=0;
      end
    else
      begin
      state=state_61;
      z=0;
      end
  end
state_61:
  begin
    if(x)
      begin
      state=state_62;
      z=0;
      end
    else
      begin
      state=state_1;
      z=0;
      end
  end
state_62:
  begin
    if(x)
      begin
      state=state_2;
      z=0;
      end
    else
      begin
      state=state_63;
      z=0;
      end
  end
state_63:
  begin
    if(x)
      begin
      state=state_64;
      z=0;
      end
    else
      begin
```

```
             state = state_1;
             z = 0;
             end
         end
       state_64:
         begin
           if (x)
             begin
               state = state_2;
               z = 1;
             end
           else
             begin
               state = state_1;
                z = 0;
             end
         end
   endcase
   end

 endmodule
```

Test Bench for the Behavioral code

```
//Testbench of above code

module testSequeneDetector ();
  reg x, CLK, reset;
  wire z;

 EthPreambleDetector mySD(x,CLK,reset,z);
 //initial
 //begin
 //
 //#1 CLK = 0;
 //forever
 //   begin
 //    #30 CLK = ~CLK;
 //  end
 //end

 //always  @(CLK or x)
 //begin
 //   forever
 //   begin
 //    #30 CLK = ~CLK;
 //  end
 //#25 CLK = ~CLK;
 //end

 initial
 begin

 CLK=0;
 reset = 1;
 x = 0;
 #30reset = 0;

 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 0;
 #30 x = 1;
 #30 x = 1;
#30 x = 0;
//#30 x =
 //#30 x =
 //#30 x =
 //#30 x =
 //#30 x =
 //#30 x =
 //#30 x =
 #30 $finish;
end
always #15 CLK = ~CLK;
endmodule
```