1 Mapped Verilog Generated by the RTL Compiler

```
// Generated by Cadence Encounter(R) RTL Compiler RC14.13 - v14.10-s027_1

// Verification Directory fv/TCP_FSM

module sub_signed(A, B, Z);
  input [31:0] A;
  input [1:0] B;
  output [31:0] Z;
  wire [31:0] A;
  wire [1:0] B;
  wire [31:0] Z;
  wire n_1, n_3, n_5, n_7, n_9, n_11, n_13, n_15;
  wire n_17, n_19, n_21, n_23, n_25, n_27, n_29, n_31;
  wire n_33, n_35, n_37, n_39, n_41, n_43, n_45, n_47;
  wire n_49, n_51, n_53, n_55, n_57, n_59;
  XNOR2X1 g743(.A (A[31]), .B (n_59), .Y (Z[31]));
  OAI2BB1X1 g744(.A0N (A[30]), .A1N (n_57), .B0 (n_59), .Y (Z[30]));
  OR2XL g745(.A (A[30]), .B (n_57), .Y (n_59));
  OAI2BB1X1 g746(.A0N (A[29]), .A1N (n_55), .B0 (n_57), .Y (Z[29]));
  OR2XL g747(.A (A[29]), .B (n_55), .Y (n_57));
  OAI2BB1X1 g748(.A0N (A[28]), .A1N (n_53), .B0 (n_55), .Y (Z[28]));
  OR2XL g749(.A (A[28]), .B (n_53), .Y (n_55));
  OAI2BB1X1 g750(.A0N (A[27]), .A1N (n_51), .B0 (n_53), .Y (Z[27]));
  OR2XL g751(.A (A[27]), .B (n_51), .Y (n_53));
  OAI2BB1X1 g752(.A0N (A[26]), .A1N (n_49), .B0 (n_51), .Y (Z[26]));
  OR2XL g753(.A (A[26]), .B (n_49), .Y (n_51));
  OAI2BB1X1 g754(.A0N (A[25]), .A1N (n_47), .B0 (n_49), .Y (Z[25]));
  OR2XL g755(.A (A[25]), .B (n_47), .Y (n_49));
  OAI2BB1X1 g756(.A0N (A[24]), .A1N (n_45), .B0 (n_47), .Y (Z[24]));
  OR2XL g757(.A (A[24]), .B (n_45), .Y (n_47));
  OAI2BB1X1 g758(.A0N (A[23]), .A1N (n_43), .B0 (n_45), .Y (Z[23]));
  OR2XL g759(.A (A[23]), .B (n_43), .Y (n_45));
  OAI2BB1X1 g760(.A0N (A[22]), .A1N (n_41), .B0 (n_43), .Y (Z[22]));
  OR2XL g761(.A (A[22]), .B (n_41), .Y (n_43));
  OAI2BB1X1 g762(.A0N (A[21]), .A1N (n_39), .B0 (n_41), .Y (Z[21]));
  OR2XL g763(.A (A[21]), .B (n_39), .Y (n_41));
  OAI2BB1X1 g764(.A0N (A[20]), .A1N (n_37), .B0 (n_39), .Y (Z[20]));
  OR2XL g765(.A (A[20]), .B (n_37), .Y (n_39));
  OAI2BB1X1 g766(.A0N (A[19]), .A1N (n_35), .B0 (n_37), .Y (Z[19]));
  OR2XL g767(.A (A[19]), .B (n_35), .Y (n_37));
  OAI2BB1X1 g768(.A0N (A[18]), .A1N (n_33), .B0 (n_35), .Y (Z[18]));
  OR2XL g769(.A (A[18]), .B (n_33), .Y (n_35));
  OAI2BB1X1 g770(.A0N (A[17]), .A1N (n_31), .B0 (n_33), .Y (Z[17]));
  OR2XL g771(.A (A[17]), .B (n_31), .Y (n_33));
  OAI2BB1X1 g772(.A0N (A[16]), .A1N (n_29), .B0 (n_31), .Y (Z[16]));
  OR2XL g773(.A (A[16]), .B (n_29), .Y (n_31));
  OAI2BB1X1 g774(.A0N (A[15]), .A1N (n_27), .B0 (n_29), .Y (Z[15]));
  OR2XL g775(.A (A[15]), .B (n_27), .Y (n_29));
  OAI2BB1X1 g776(.A0N (A[14]), .A1N (n_25), .B0 (n_27), .Y (Z[14]));
  OR2XL g777(.A (A[14]), .B (n_25), .Y (n_27));
  OAI2BB1X1 g778(.A0N (A[13]), .A1N (n_23), .B0 (n_25), .Y (Z[13]));
  OR2XL g779(.A (A[13]), .B (n_23), .Y (n_25));
  OAI2BB1X1 g780(.A0N (A[12]), .A1N (n_21), .B0 (n_23), .Y (Z[12]));
  OR2XL g781(.A (A[12]), .B (n_21), .Y (n_23));
  OAI2BB1X1 g782(.A0N (A[11]), .A1N (n_19), .B0 (n_21), .Y (Z[11]));
  OR2XL g783(.A (A[11]), .B (n_19), .Y (n_21));
  OAI2BB1X1 g784(.A0N (A[10]), .A1N (n_17), .B0 (n_19), .Y (Z[10]));
  OR2XL g785(.A (A[10]), .B (n_17), .Y (n_19));
  OAI2BB1X1 g786(.A0N (A[9]), .A1N (n_15), .B0 (n_17), .Y (Z[9]));
  OR2XL g787(.A (A[9]), .B (n_15), .Y (n_17));
  OAI2BB1X1 g788(.A0N (A[8]), .A1N (n_13), .B0 (n_15), .Y (Z[8]));
  OR2XL g789(.A (A[8]), .B (n_13), .Y (n_15));
  OAI2BB1X1 g790(.A0N (A[7]), .A1N (n_11), .B0 (n_13), .Y (Z[7]));
  OR2XL g791(.A (A[7]), .B (n_11), .Y (n_13));
  OAI2BB1X1 g792(.A0N (A[6]), .A1N (n_9), .B0 (n_11), .Y (Z[6]));
```

```
  OR2XL g793(.A (A[6]), .B (n_9), .Y (n_11));
  OAI2BB1X1 g794(.AON (A[5]), .A1N (n_7), .B0 (n_9), .Y (Z[5]));
  OR2XL g795(.A (A[5]), .B (n_7), .Y (n_9));
  OAI2BB1X1 g796(.AON (A[4]), .A1N (n_5), .B0 (n_7), .Y (Z[4]));
  OR2XL g797(.A (A[4]), .B (n_5), .Y (n_7));
  OAI2BB1X1 g798(.AON (A[3]), .A1N (n_3), .B0 (n_5), .Y (Z[3]));
  OR2XL g799(.A (A[3]), .B (n_3), .Y (n_5));
  OAI2BB1X1 g800(.AON (A[2]), .A1N (n_1), .B0 (n_3), .Y (Z[2]));
  OR2XL g801(.A (A[2]), .B (n_1), .Y (n_3));
  OAI2BB1X1 g802(.AON (A[0]), .A1N (A[1]), .B0 (n_1), .Y (Z[1]));
  OR2XL g803(.A (A[0]), .B (A[1]), .Y (n_1));
  TIELO tie_0_cell(.Y (Z[0]));
endmodule

module TCP_FSM(clk, reset, ACK, SYN, FIN, Buffer, Enable, Mode, Port,
     SYNout, ACKout, FINout, Protocol, StateNum);
  input clk, reset, ACK, SYN, FIN, Buffer, Enable, Mode;
  input [15:0] Port;
  output SYNout, ACKout, FINout;
  output [2:0] Protocol;
  output [3:0] StateNum;
  wire clk, reset, ACK, SYN, FIN, Buffer, Enable, Mode;
  wire [15:0] Port;
  wire SYNout, ACKout, FINout;
  wire [2:0] Protocol;
  wire [3:0] StateNum;
  wire [31:0] timeoutCounter;
  wire [3:0] state;
  wire UNCONNECTED, logic_1_1_net, n_0, n_1, n_2, n_3, n_4, n_5;
  wire n_6, n_7, n_8, n_9, n_10, n_11, n_12, n_13;
  wire n_14, n_15, n_16, n_17, n_18, n_19, n_20, n_21;
  wire n_22, n_23, n_24, n_25, n_26, n_27, n_28, n_29;
  wire n_30, n_31, n_32, n_33, n_34, n_35, n_36, n_37;
  wire n_38, n_39, n_40, n_41, n_42, n_43, n_44, n_45;
  wire n_46, n_47, n_48, n_49, n_50, n_51, n_52, n_53;
  wire n_54, n_55, n_56, n_57, n_58, n_59, n_60, n_61;
  wire n_62, n_63, n_64, n_65, n_66, n_67, n_68, n_69;
  wire n_70, n_71, n_72, n_73, n_74, n_75, n_76, n_77;
  wire n_78, n_79, n_80, n_81, n_82, n_83, n_84, n_85;
  wire n_86, n_87, n_88, n_89, n_90, n_91, n_92, n_93;
  wire n_94, n_95, n_96, n_97, n_98, n_99, n_100, n_101;
  wire n_102, n_103, n_118, n_119, n_120, n_121, n_122, n_123;
  wire n_124, n_125, n_126, n_127, n_128, n_129, n_130, n_131;
  wire n_132, n_133, n_134, n_135, n_136, n_137, n_138, n_139;
  wire n_140, n_141, n_142, n_143, n_144, n_145, n_146, n_147;
  wire n_148, n_170, n_171;
  sub_signed sub_60_43(.A (timeoutCounter), .B ({Protocol[2],
       logic_1_1_net}), .Z ({n_118, n_119, n_120, n_121, n_122, n_123,
       n_124, n_125, n_126, n_127, n_128, n_129, n_130, n_131, n_132,
       n_133, n_134, n_135, n_136, n_137, n_138, n_139, n_140, n_141,
       n_142, n_143, n_144, n_145, n_146, n_147, n_148, UNCONNECTED}));
  OAI2BB1X1 g1589(.AON (state[0]), .A1N (Protocol[1]), .B0 (n_102), .Y
       (Protocol[0]));
  OAI31X1 g1590(.A0 (n_3), .A1 (state[2]), .A2 (n_99), .B0 (n_103), .Y
       (ACKout));
  OAI22XL g1591(.A0 (n_170), .A1 (n_101), .B0 (n_3), .B1 (n_99), .Y
       (StateNum[1]));
  NAND2BXL g1592(.AN (Protocol[1]), .B (n_102), .Y (StateNum[3]));
  OAI21XL g1593(.A0 (state[0]), .A1 (n_171), .B0 (n_99), .Y
       (StateNum[0]));
  INVX1 g1594(.A (n_103), .Y (FINout));
  NAND3BXL g1595(.AN (n_99), .B (n_3), .C (state[2]), .Y (n_103));
  MX2XL g1596(.A (n_97), .B (n_100), .S0 (n_98), .Y (StateNum[2]));
  NAND2BXL g1597(.AN (n_98), .B (n_100), .Y (n_102));
  NOR2BX1 g1598(.AN (state[3]), .B (n_171), .Y (Protocol[1]));
  AND2XL g1599(.A (state[1]), .B (n_97), .Y (SYNout));
  OA21X1 g1600(.A0 (state[1]), .A1 (state[3]), .B0 (n_171), .Y (n_101));
  NOR2XL g1601(.A (state[3]), .B (n_10), .Y (n_100));
```

```
NAND2X1 g1602(.A (n_3), .B (n_10), .Y (n_171));
OR2X1 g1603(.A (state[3]), .B (state[0]), .Y (n_99));
NAND2XL g1604(.A (state[1]), .B (state[0]), .Y (n_98));
NOR2XL g1605(.A (state[2]), .B (state[3]), .Y (n_97));
SDFFQX1 \timeoutCounter_reg[0] (.CK (clk), .D (n_33), .SI (reset),
     .SE (timeoutCounter[0]), .Q (timeoutCounter[0]));
DFFHQX1 \timeoutCounter_reg[10] (.CK (clk), .D (n_45), .Q
     (timeoutCounter[10]));
DFFHQX1 \timeoutCounter_reg[11] (.CK (clk), .D (n_44), .Q
     (timeoutCounter[11]));
DFFHQX1 \timeoutCounter_reg[12] (.CK (clk), .D (n_43), .Q
     (timeoutCounter[12]));
DFFHQX1 \timeoutCounter_reg[13] (.CK (clk), .D (n_42), .Q
     (timeoutCounter[13]));
DFFHQX1 \timeoutCounter_reg[14] (.CK (clk), .D (n_41), .Q
     (timeoutCounter[14]));
DFFHQX1 \timeoutCounter_reg[15] (.CK (clk), .D (n_40), .Q
     (timeoutCounter[15]));
DFFHQX1 \timeoutCounter_reg[16] (.CK (clk), .D (n_39), .Q
     (timeoutCounter[16]));
DFFHQX1 \timeoutCounter_reg[17] (.CK (clk), .D (n_67), .Q
     (timeoutCounter[17]));
DFFHQX1 \timeoutCounter_reg[18] (.CK (clk), .D (n_66), .Q
     (timeoutCounter[18]));
DFFHQX1 \timeoutCounter_reg[19] (.CK (clk), .D (n_65), .Q
     (timeoutCounter[19]));
DFFHQX1 \timeoutCounter_reg[1] (.CK (clk), .D (n_72), .Q
     (timeoutCounter[1]));
DFFHQX1 \timeoutCounter_reg[20] (.CK (clk), .D (n_64), .Q
     (timeoutCounter[20]));
DFFHQX1 \timeoutCounter_reg[21] (.CK (clk), .D (n_63), .Q
     (timeoutCounter[21]));
DFFHQX1 \timeoutCounter_reg[22] (.CK (clk), .D (n_62), .Q
     (timeoutCounter[22]));
DFFHQX1 \timeoutCounter_reg[23] (.CK (clk), .D (n_61), .Q
     (timeoutCounter[23]));
DFFHQX1 \timeoutCounter_reg[24] (.CK (clk), .D (n_60), .Q
     (timeoutCounter[24]));
DFFHQX1 \timeoutCounter_reg[25] (.CK (clk), .D (n_59), .Q
     (timeoutCounter[25]));
DFFHQX1 \timeoutCounter_reg[26] (.CK (clk), .D (n_58), .Q
     (timeoutCounter[26]));
DFFHQX1 \timeoutCounter_reg[27] (.CK (clk), .D (n_57), .Q
     (timeoutCounter[27]));
DFFHQX1 \timeoutCounter_reg[28] (.CK (clk), .D (n_56), .Q
     (timeoutCounter[28]));
DFFHQX1 \timeoutCounter_reg[29] (.CK (clk), .D (n_55), .Q
     (timeoutCounter[29]));
DFFHQX1 \timeoutCounter_reg[2] (.CK (clk), .D (n_54), .Q
     (timeoutCounter[2]));
DFFHQX1 \timeoutCounter_reg[30] (.CK (clk), .D (n_53), .Q
     (timeoutCounter[30]));
DFFHQX1 \timeoutCounter_reg[31] (.CK (clk), .D (n_52), .Q
     (timeoutCounter[31]));
DFFHQX1 \timeoutCounter_reg[3] (.CK (clk), .D (n_69), .Q
     (timeoutCounter[3]));
DFFHQX1 \timeoutCounter_reg[4] (.CK (clk), .D (n_51), .Q
     (timeoutCounter[4]));
DFFHQX1 \timeoutCounter_reg[5] (.CK (clk), .D (n_50), .Q
     (timeoutCounter[5]));
DFFHQX1 \timeoutCounter_reg[6] (.CK (clk), .D (n_49), .Q
     (timeoutCounter[6]));
DFFHQX1 \timeoutCounter_reg[7] (.CK (clk), .D (n_48), .Q
     (timeoutCounter[7]));
DFFHQX1 \timeoutCounter_reg[8] (.CK (clk), .D (n_47), .Q
     (timeoutCounter[8]));
DFFHQX1 \timeoutCounter_reg[9] (.CK (clk), .D (n_46), .Q
     (timeoutCounter[9]));
```

```
OAI2BB1X1 g4377(.A0N (n_7), .A1N (n_30), .B0 (n_95), .Y (n_96));
OAI211X1 g4380(.A0 (n_90), .A1 (state[0]), .B0 (n_14), .C0 (n_37), .Y
     (n_95));
OAI31X1 g4381(.A0 (n_21), .A1 (n_27), .A2 (state[2]), .B0 (n_92), .Y
     (n_94));
AOI221X1 g4382(.A0 (SYN), .A1 (n_31), .B0 (n_20), .B1 (n_89), .C0
     (n_34), .Y (n_93));
AOI22X1 g4384(.A0 (Buffer), .A1 (n_2), .B0 (n_20), .B1 (n_26), .Y
     (n_92));
AOI21X1 g4385(.A0 (n_29), .A1 (n_87), .B0 (n_6), .Y (n_91));
OAI211X1 g4386(.A0 (Port[3]), .A1 (n_85), .B0 (n_22), .C0 (n_86), .Y
     (n_90));
OAI31X1 g4387(.A0 (ACK), .A1 (n_10), .A2 (state[1]), .B0 (n_86), .Y
     (n_89));
AOI33XL g4389(.A0 (n_20), .A1 (n_70), .A2 (n_83), .B0 (state[2]), .B1
     (n_14), .B2 (state[0]), .Y (n_88));
NAND2X1 g4390(.A (n_84), .B (state[1]), .Y (n_87));
AOI22X1 g4391(.A0 (n_6), .A1 (n_83), .B0 (n_11), .B1 (n_10), .Y
     (n_86));
OR4X1 g4392(.A (Port[1]), .B (n_28), .C (n_32), .D (n_82), .Y (n_85));
NOR4X1 g4393(.A (n_1), .B (n_70), .C (n_10), .D (n_81), .Y (n_84));
INVX1 g4394(.A (n_82), .Y (n_83));
NAND3BXL g4395(.AN (n_81), .B (state[2]), .C (state[1]), .Y (n_82));
NOR4X1 g4396(.A (timeoutCounter[6]), .B (timeoutCounter[7]), .C
     (timeoutCounter[8]), .D (n_80), .Y (n_81));
OR4X1 g4397(.A (timeoutCounter[3]), .B (timeoutCounter[4]), .C
     (timeoutCounter[29]), .D (n_79), .Y (n_80));
OR4X1 g4398(.A (timeoutCounter[0]), .B (timeoutCounter[25]), .C
     (timeoutCounter[26]), .D (n_78), .Y (n_79));
OR4X1 g4399(.A (timeoutCounter[27]), .B (timeoutCounter[28]), .C
     (timeoutCounter[31]), .D (n_77), .Y (n_78));
OR4X1 g4400(.A (timeoutCounter[13]), .B (timeoutCounter[14]), .C
     (timeoutCounter[30]), .D (n_76), .Y (n_77));
OR4X1 g4401(.A (timeoutCounter[15]), .B (timeoutCounter[16]), .C
     (timeoutCounter[9]), .D (n_75), .Y (n_76));
OR4X1 g4402(.A (timeoutCounter[10]), .B (timeoutCounter[11]), .C
     (timeoutCounter[12]), .D (n_74), .Y (n_75));
OR4X1 g4403(.A (timeoutCounter[17]), .B (timeoutCounter[18]), .C
     (timeoutCounter[19]), .D (n_73), .Y (n_74));
OR4X1 g4436(.A (timeoutCounter[20]), .B (timeoutCounter[23]), .C
     (timeoutCounter[24]), .D (n_35), .Y (n_73));
INVX1 g4437(.A (n_71), .Y (n_72));
AOI221X1 g4438(.A0 (n_33), .A1 (n_148), .B0 (reset), .B1
     (timeoutCounter[1]), .C0 (n_36), .Y (n_71));
NAND3BXL g4439(.AN (Port[6]), .B (Port[0]), .C (n_38), .Y (n_70));
INVX1 g4440(.A (n_68), .Y (n_69));
AOI221X1 g4441(.A0 (reset), .A1 (timeoutCounter[3]), .B0 (n_33), .B1
     (n_146), .C0 (n_36), .Y (n_68));
AO22XL g4442(.A0 (n_33), .A1 (n_132), .B0 (reset), .B1
     (timeoutCounter[17]), .Y (n_67));
AO22XL g4443(.A0 (n_33), .A1 (n_131), .B0 (reset), .B1
     (timeoutCounter[18]), .Y (n_66));
AO22XL g4444(.A0 (n_33), .A1 (n_130), .B0 (reset), .B1
     (timeoutCounter[19]), .Y (n_65));
AO22XL g4445(.A0 (n_33), .A1 (n_129), .B0 (reset), .B1
     (timeoutCounter[20]), .Y (n_64));
AO22XL g4446(.A0 (n_33), .A1 (n_128), .B0 (reset), .B1
     (timeoutCounter[21]), .Y (n_63));
AO22XL g4447(.A0 (n_33), .A1 (n_127), .B0 (reset), .B1
     (timeoutCounter[22]), .Y (n_62));
AO22XL g4448(.A0 (n_33), .A1 (n_126), .B0 (reset), .B1
     (timeoutCounter[23]), .Y (n_61));
AO22XL g4449(.A0 (n_33), .A1 (n_125), .B0 (reset), .B1
     (timeoutCounter[24]), .Y (n_60));
AO22XL g4450(.A0 (n_33), .A1 (n_124), .B0 (reset), .B1
     (timeoutCounter[25]), .Y (n_59));
AO22XL g4451(.A0 (n_33), .A1 (n_123), .B0 (reset), .B1
     (timeoutCounter[26]), .Y (n_58));
```

```
AO22XL g4452(.A0 (n_33), .A1 (n_122), .B0 (reset), .B1
     (timeoutCounter[27]), .Y (n_57));
AO22XL g4453(.A0 (n_33), .A1 (n_121), .B0 (reset), .B1
     (timeoutCounter[28]), .Y (n_56));
AO22XL g4454(.A0 (n_33), .A1 (n_120), .B0 (reset), .B1
     (timeoutCounter[29]), .Y (n_55));
AO22XL g4455(.A0 (n_33), .A1 (n_147), .B0 (reset), .B1
     (timeoutCounter[2]), .Y (n_54));
AO22XL g4456(.A0 (n_33), .A1 (n_119), .B0 (reset), .B1
     (timeoutCounter[30]), .Y (n_53));
AO22XL g4457(.A0 (n_33), .A1 (n_118), .B0 (reset), .B1
     (timeoutCounter[31]), .Y (n_52));
AO22XL g4458(.A0 (n_33), .A1 (n_145), .B0 (reset), .B1
     (timeoutCounter[4]), .Y (n_51));
AO22XL g4459(.A0 (n_33), .A1 (n_144), .B0 (reset), .B1
     (timeoutCounter[5]), .Y (n_50));
AO22XL g4460(.A0 (n_33), .A1 (n_143), .B0 (reset), .B1
     (timeoutCounter[6]), .Y (n_49));
AO22XL g4461(.A0 (n_33), .A1 (n_142), .B0 (reset), .B1
     (timeoutCounter[7]), .Y (n_48));
AO22XL g4462(.A0 (n_33), .A1 (n_141), .B0 (reset), .B1
     (timeoutCounter[8]), .Y (n_47));
AO22XL g4463(.A0 (n_33), .A1 (n_140), .B0 (reset), .B1
     (timeoutCounter[9]), .Y (n_46));
AO22XL g4464(.A0 (n_33), .A1 (n_139), .B0 (reset), .B1
     (timeoutCounter[10]), .Y (n_45));
AO22XL g4465(.A0 (n_33), .A1 (n_138), .B0 (reset), .B1
     (timeoutCounter[11]), .Y (n_44));
AO22XL g4466(.A0 (n_33), .A1 (n_137), .B0 (reset), .B1
     (timeoutCounter[12]), .Y (n_43));
AO22XL g4467(.A0 (n_33), .A1 (n_136), .B0 (reset), .B1
     (timeoutCounter[13]), .Y (n_42));
AO22XL g4468(.A0 (n_33), .A1 (n_135), .B0 (reset), .B1
     (timeoutCounter[14]), .Y (n_41));
AO22XL g4469(.A0 (n_33), .A1 (n_134), .B0 (reset), .B1
     (timeoutCounter[15]), .Y (n_40));
AO22XL g4470(.A0 (n_33), .A1 (n_133), .B0 (reset), .B1
     (timeoutCounter[16]), .Y (n_39));
NOR3XL g4471(.A (Port[1]), .B (n_17), .C (n_32), .Y (n_38));
OAI2BB1X1 g4472(.A0N (n_19), .A1N (n_23), .B0 (state[0]), .Y (n_37));
NOR2X1 g4473(.A (reset), .B (n_24), .Y (n_36));
NAND2BX1 g4474(.AN (timeoutCounter[21]), .B (n_25), .Y (n_35));
NOR2X1 g4475(.A (Buffer), .B (n_29), .Y (n_34));
AND2X1 g4476(.A (n_24), .B (n_7), .Y (n_33));
NAND4XL g4477(.A (Port[4]), .B (n_13), .C (n_16), .D (n_15), .Y
     (n_32));
NOR2BX1 g4478(.AN (n_11), .B (n_21), .Y (n_31));
AOI211XL g4479(.A0 (Buffer), .A1 (n_170), .B0 (n_8), .C0 (n_171), .Y
     (n_30));
NAND3X1 g4480(.A (n_7), .B (state[3]), .C (n_12), .Y (n_29));
AOI33XL g4481(.A0 (Port[6]), .A1 (n_4), .A2 (n_9), .B0 (Port[0]), .B1
     (Port[2]), .B2 (n_5), .Y (n_28));
MX2XL g4482(.A (n_3), .B (n_11), .S0 (SYN), .Y (n_27));
OAI31X1 g4483(.A0 (ACK), .A1 (state[2]), .A2 (n_3), .B0 (n_18), .Y
     (n_26));
NOR4X1 g4484(.A (timeoutCounter[1]), .B (timeoutCounter[2]), .C
     (timeoutCounter[5]), .D (timeoutCounter[22]), .Y (n_25));
NOR4X1 g4485(.A (state[3]), .B (state[0]), .C (n_3), .D (n_10), .Y
     (n_24));
OAI211X1 g4486(.A0 (Buffer), .A1 (FIN), .B0 (state[2]), .C0 (n_3), .Y
     (n_23));
NAND2X1 g4487(.A (Enable), .B (n_12), .Y (n_22));
NAND2X1 g4488(.A (n_14), .B (state[0]), .Y (n_21));
INVX1 g4489(.A (n_1), .Y (n_20));
NAND2X1 g4491(.A (SYN), .B (n_12), .Y (n_19));
NAND3BXL g4492(.AN (Mode), .B (Enable), .C (n_12), .Y (n_18));
XNOR2X1 g4493(.A (Port[2]), .B (Port[3]), .Y (n_17));
NOR4X1 g4494(.A (Port[10]), .B (Port[11]), .C (Port[12]), .D
```

```
      (Port[13]), .Y (n_16));
  NOR4X1 g4495(.A (Port[5]), .B (Port[7]), .C (Port[8]), .D (Port[9]),
      .Y (n_15));
  NOR2X1 g4496(.A (reset), .B (state[3]), .Y (n_14));
  NOR2XL g4497(.A (Port[14]), .B (Port[15]), .Y (n_13));
  NOR2X1 g4498(.A (state[2]), .B (state[1]), .Y (n_12));
  NOR2BX1 g4499(.AN (ACK), .B (n_3), .Y (n_11));
  INVX1 g4502(.A (Port[0]), .Y (n_9));
  INVX1 g4504(.A (reset), .Y (n_7));
  INVX1 g4505(.A (Buffer), .Y (n_6));
  INVX1 g4506(.A (Port[6]), .Y (n_5));
  INVX1 g4507(.A (Port[2]), .Y (n_4));
  INVX1 drc_bufs4510(.A (n_88), .Y (n_2));
  NAND2BX1 g2(.AN (state[0]), .B (n_14), .Y (n_1));
  NAND2BX1 g4512(.AN (n_2), .B (n_93), .Y (n_0));
  DFFX1 \state_reg[2] (.CK (clk), .D (n_0), .Q (state[2]), .QN (n_10));
  DFFX1 \state_reg[1] (.CK (clk), .D (n_94), .Q (state[1]), .QN (n_3));
  DFFX1 \state_reg[0] (.CK (clk), .D (n_96), .Q (state[0]), .QN
      (n_170));
  DFFX1 \state_reg[3] (.CK (clk), .D (n_91), .Q (state[3]), .QN (n_8));
  TIELO tie_0_cell(.Y (Protocol[2]));
  TIEHI tie_1_cell(.Y (logic_1_1_net));
endmodule
```

2 Visual Waveforms of State Transitions from Mapped Verilog

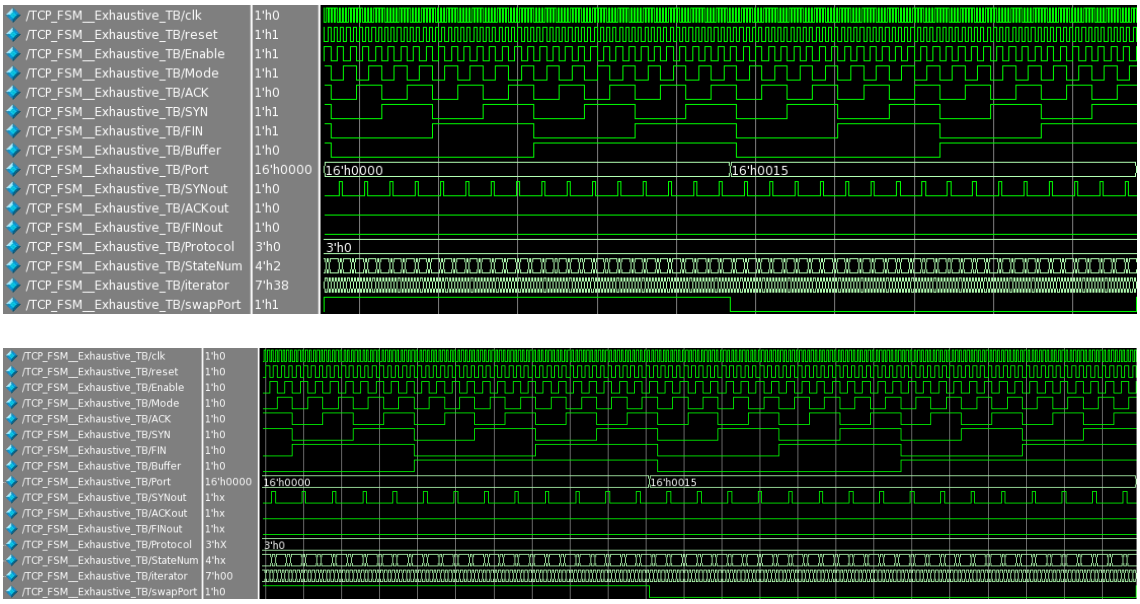

Figure 1: Synthesized (top) and Unsynthesized (bottom) Exhaustive Testbench

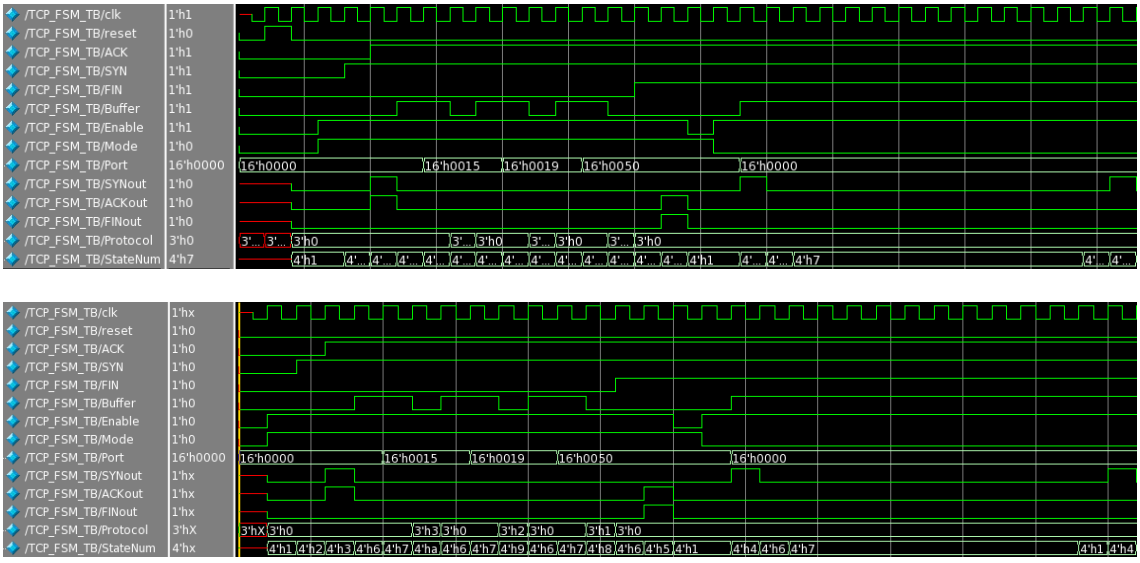

Figure 2: Synthesized (top) and Unsynthesized (bottom) User Path Testbench

Note: I had to modify the testbench for the simulation of the User Path testbench with the synthesized verilog. Observe there is an added toggle of the reset bit in the beginning of the testbench for the synthesized waveform.

In the unsynthesized version, the reset bit was set to zero while the clock still had a value of `X` and this did not pose a problem. However, with the synthesized code, this reset bit needed to be set to zero synchronously with the clock, otherwise the outputs remained undriven. See section 4.

3 RTL Compiler Report

The following are the contents of the `/elec402/Cadence_21518139/synth/files_out/TCP_FSM_gates.rpt` file. My finite state machine uses **211 cells**, a large number of these are in service of the timeout counter for the Read state.

```
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.13 - v14.10-s027_1
  Generated on:           Oct 09 2016  08:45:21 pm
  Module:                 TCP_FSM
  Technology library:     slow_vdd1v0 1.0
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

   Gate    Instances   Area       Library
-----------------------------------------------
AND2X1             1    1.368    slow_vdd1v0
AND2XL             1    1.368    slow_vdd1v0
AO22XL            29   79.344    slow_vdd1v0
AOI211XL           1    2.052    slow_vdd1v0
AOI21X1            1    1.710    slow_vdd1v0
AOI221X1           3    7.182    slow_vdd1v0
AOI22X1            2    4.104    slow_vdd1v0
AOI33XL            2    4.788    slow_vdd1v0
DFFHQX1           31  169.632    slow_vdd1v0
DFFX1              4   28.728    slow_vdd1v0
INVX1             11    7.524    slow_vdd1v0
MX2XL              2    4.788    slow_vdd1v0
NAND2BX1           3    4.104    slow_vdd1v0
NAND2BXL           2    2.736    slow_vdd1v0
NAND2X1            5    5.130    slow_vdd1v0
NAND2XL            1    1.026    slow_vdd1v0
NAND3BXL           4    6.840    slow_vdd1v0
NAND3X1            1    1.710    slow_vdd1v0
NAND4XL            1    1.710    slow_vdd1v0
NOR2BX1            3    4.104    slow_vdd1v0
NOR2X1             4    4.104    slow_vdd1v0
NOR2XL             3    3.078    slow_vdd1v0
NOR3XL             1    1.710    slow_vdd1v0
NOR4X1             6   10.260    slow_vdd1v0
OA21X1             1    2.052    slow_vdd1v0
OAI211X1           3    5.130    slow_vdd1v0
OAI21XL            1    1.710    slow_vdd1v0
OAI22XL            1    2.052    slow_vdd1v0
OAI2BB1X1         33   56.430    slow_vdd1v0
OAI31X1            4    8.208    slow_vdd1v0
OR2X1              1    1.368    slow_vdd1v0
OR2XL             30   41.040    slow_vdd1v0
OR4X1              9   18.468    slow_vdd1v0
SDFFQX1            1    7.524    slow_vdd1v0
TIEHI              1    1.026    slow_vdd1v0
TIELO              2    2.052    slow_vdd1v0
XNOR2X1            2    4.788    slow_vdd1v0
-----------------------------------------------
total            211  510.948

   Type    Instances   Area  Area %
-----------------------------------
sequential        36 205.884   40.3
inverter          11   7.524    1.5
logic            164 297.540   58.2
-----------------------------------
total            211 510.948  100.0
```

4 Verilog Code Changes

```
module TCP_FSM_TB;
// inputs of FSM
logic clk;
logic reset;
logic ACK;
logic SYN;
logic FIN;
logic Buffer;
logic Enable;
logic Mode;
logic [15:0] Port;
// outputs of FSM
logic SYNout;
logic ACKout;
logic FINout;
logic [2:0] Protocol;
logic [3:0] StateNum;

// connecting logit to FSM module
TCP_FSM U0(
    .clk(clk),
    .reset(reset),
    .ACK(ACK),
    .SYN(SYN),
    .FIN(FIN),
    .Buffer(Buffer),
    .Enable(Enable),
    .Mode(Mode),
    .Port(Port),
    .SYNout(SYNout),
    .ACKout(ACKout),
    .FINout(FINout),
    .Protocol(Protocol),
    .StateNum(StateNum)
);

// This is a fixed sequence of inputs.
// This process is an example of a typical user path
// that a TPC router may walk through, this particular
// example will send us through every single state in the FSM
 initial begin
    // initialize inputs other than reset to zero
    reset=1; Enable=0; Mode=0; SYN=0; ACK=0; Port=0; Buffer=0; FIN=0;

    // toggle the reset
    reset=0;          #200;
                           //                                        NOTE:
    reset=1;          #200; // <----------------------- these two lines of code were added --------------- //
    reset=0;          #200; // <----------------------- to the testbench -------------------------------- //

                           // enable the router in mode 1 (listen mode) -- go to Listen state
    Enable=1; Mode=1; #200;

    // provide a SYN input to request connection -- go to Open state
    SYN=1;            #200;

    // provide an ACK response to the SYNACK -- go to Await state
    ACK=1;            #200;

    // provide a Buffer signal to indicate queued packet -- go to Read state
    Buffer=1;         #200;

    // provide a Port number 21 to indidate FTP protocol -- go to FTP state
    Port=21;          #200;
```

```
        // disable Buffer signal to indicate packet read -- go back to Await state
        Buffer=0;         #200;

        // reenable Buffer to indidate another packet -- go to Read state
        Buffer=1;         #200;

         // provide port Number 25 to indicate SMTP protocol -- go to SMTP state
        Port=25;          #200;

        // clear buffer -- go back to Await state
        Buffer=0;         #200;

        // reenable Buffer to indidate another packet -- go to Read state
        Buffer=1;         #200;

         // provide port Number 80 to indicate HTTP protocol -- go to HTTP state
        Port=80;          #200;

                              // clear buffer -- back to Await state
                              Buffer=0;         #200;

        // provide FIN request to close connection -- go to Finish state
        FIN=1;            #200;

        // provide ACK to confirm close -- go to Closed state
        ACK=1;            #200;

        // disable Enable to stay in Closed state
        Enable=0;         #200;

        // re-enable in Mode 0 (send mode) -- go to Send state
        Enable=1; Mode=0; #200;

        // provide SYNACK to accept connection, this will send us to Await
        // but a high buffer will send us straight through to the Read state
        // however, Port=0 is an unrecognized port number -- stay in Read state
        SYN=1; ACK=1; Buffer=1; Port=0; #200;

        // wait for Read state to timeout and send us back to Closed state
        #5000;

        // reset the router and turn off enable
        reset=1; Enable=0; #200;
    end


    // Defining clock to cycle
    always
        begin
            #100 clk = 0;
            #100 clk = 1;
        end

endmodule
```