

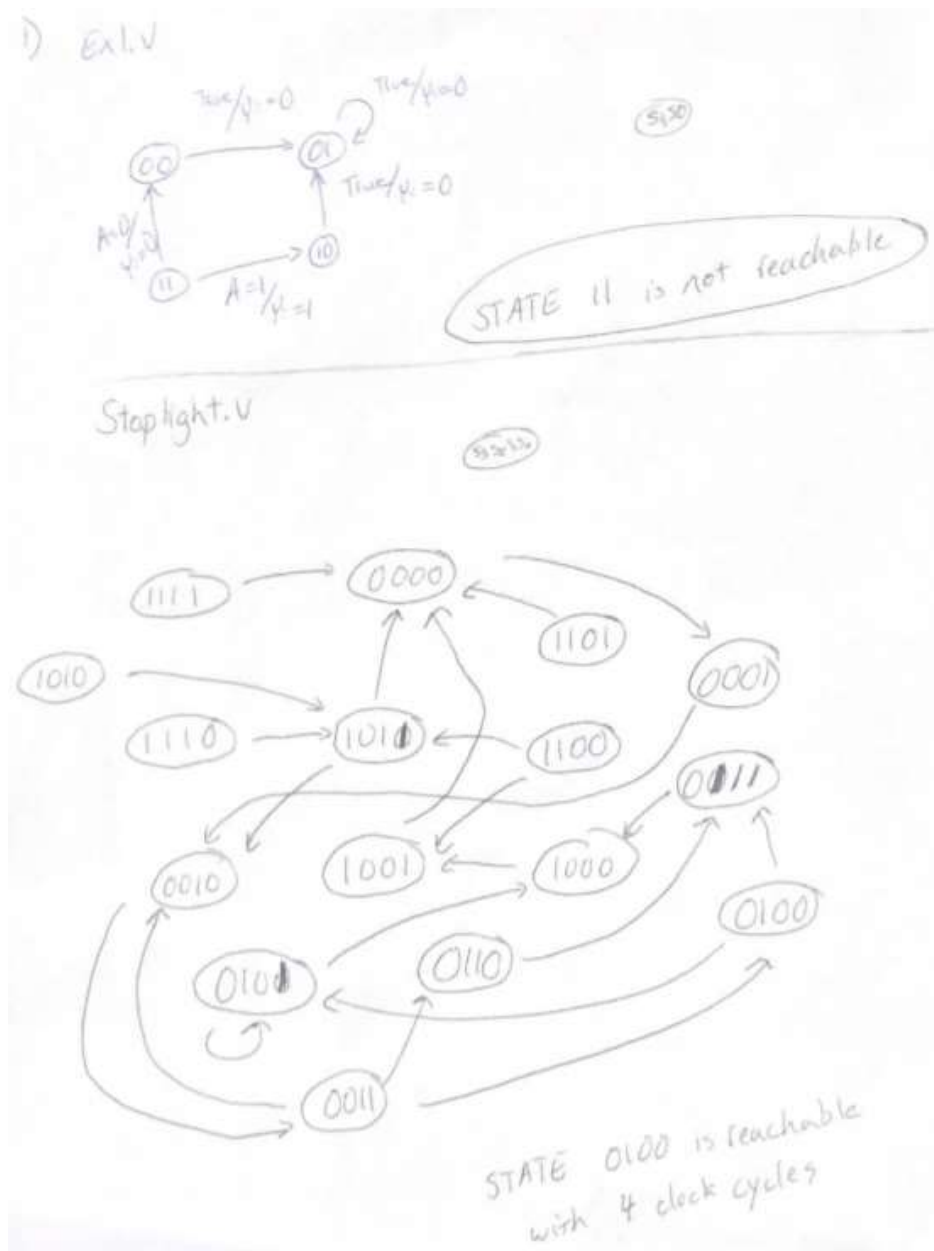
Alan Chen

ECE597 Lab 1

2/21/2020

## Part A

1. Draw legibly the state transition graph for ex1.v and stoplight1.v. Include all states, even if unreachable. You can use any method that you prefer for figuring out which state transitions are possible (pen and paper, simulation, etc), but you must include all transitions.



**2. Using these two state transition graphs, indicate whether the target states are reachable. If so, what is the minimum number of execution steps (clock cycles) needed to get from the initial state to the target state?**

For Ex1.v the target state 11 is not reachable

For Stoplight1.v the target state 0100 is reachable with 4 clock cycles

**3. If you are given an arbitrary transition relation that maps states to sets of states, describe in pseudo-code the most efficient algorithm you can think of to check whether a target state is reachable from initial state. Your algorithm must terminate upon (1) reaching the target, or (2) finding that the target can never be reached.**

Pseudo-Code

Given: State 1: {next state 1, ... }, State 3: {next state 1, ... }, ...

# General Depth First Search

arrVisited = []

myStack = Stack()

myStack.push (initial State)

while myStack not Empty:

    currentState = myStack.pop()

    arrVisited.append (currentState)

    for transitions in currentState:

        if transition == finalState:

            return True #reachable

        if transition not in arrVisited:

            myStack.push (transition)

return False    # can never be reached

4. For the benchmarks given (ex1, ex2, ex3, ex4, stoplight1, stoplight2), write a Verilog testbench to initialize the state to 0, apply random inputs in each cycle, and notify you if/when it reaches the target state. If the target state is not reached within 1M cycles, the execution can terminate, and the result can be reported as “timed out”. If the target state is reached, report how many cycles of random simulation were performed before reaching it. Include your testbench code with the report.

Ex1.v (the target state is not reachable)

testbench.v

```

1 // Code your testbench here
2 // or browse Examples
3 timescale 1 ns / 1 ps
4 module test;
5   reg A,CLK;
6   ex1 x1(CLK,A,Y); //instantiate module under test
7   initial begin
8     x1.S1 = 0; //set initial state inside module
9     x1.S0 = 0;
10    CLK = 0;
11    A = $random;
12    #1000000 $finish; //how long to run before stopping
13  end
14
15  always #0.5 CLK=~CLK; //clock toggle every 0.5 ns
16  always #20 x1.S1 = 0;
17  always #20 x1.S0 = 0;
18  always #20 A = $random;
19
20  always @(posedge CLK) begin //each cycle, print state and set next input
21    if ((x1.S1,x1.S0) == 2'b11) begin //did we hit target state?
22      $display("%8d", $time, " state:", x1.S1, x1.S0, " Input:", x1.A, " (reached target)");
23    end
24  end
25 endmodule
26
27

```

SV/Verilog Testbench

design.v

```

1 // Code your design here
2 module ex1(clock,A,Y);
3   input clock;
4   input A;
5   output Y;
6   reg S0,S1;
7   wire X1,N50,NS1;
8   and g0(X1,S0,S1);
9   and g1(NS1,A,X1);
10  not g2(NS0,X1);
11  and g3(Y,A,X1);
12  always @(posedge clock) begin
13    S1<=NS1;
14    S0<=NS0;
15  end
16 endmodule
17 //State:11

```

SV/Verilog Design

Log

Share

[2020-02-14 23:36:18 EST] iverilog '-wall' '-g2012' design.v testbench.v && unbuffer vvp a.out

testbench.v:3: warning: Some modules have no timescale. This may cause

testbench.v:3: : confusing timing results. Affected modules are:

testbench.v:3: : -- module ex1 declared here: design.v:2

testbench.v:6: warning: implicit definition of wire 'Y'.

Done

Ex2.v (the target state is not reachable)

testbench.v

```

28 always #0.5 CLK=~CLK; //clock toggle every 0.5 ns
29 always #200 START = $random;
30 always #200 x1.S0 = 0;
31 always #200 x1.S1 = 0;
32 always #200 x1.S2 = 0;
33 always #200 x1.S3 = 0;
34 always #200 x1.S4 = 0;
35 always #200 x1.S5 = 0;
36 always #200 x1.S6 = 0;
37 always #200 x1.S7 = 0;
38 always #200 x1.S8 = 0;
39 always #200 x1.S9 = 0;
40 always #200 x1.S10 = 0;
41 always #200 x1.S11 = 0;
42 always #200 x1.S12 = 0;
43 always #200 x1.S13 = 0;
44 always #200 x1.S14 = 0;
45
46 always @(posedge CLK) begin //each cycle, print state and set next input
47   if
48   ((x1.S0,x1.S1,x1.S2,x1.S3,x1.S4,x1.S5,x1.S6,x1.S7,x1.S8,x1.S9,x1.S10,x1.S11,x1.S12,x1.S13,x1.S14) ==
49   15'b111111111111111) begin //did we hit target state?
50     $display("%8d", $time, " state:", x1.S1,x1.S0, " Input:", x1.START, " (reached target)");
51   end
52 end
53 endmodule
54
55

```

SV/Verilog Testbench

design.v

```

1 module
2 ex2(clock,START,B0,B1,B2,B3,A0,A1,A2,A3,CNTVC02,CNTVC02,REA
3 DY,P0,P1,P2,P3,P4,P5,P6,P7);
4 input clock;
5 input START,B0,B1,B2,B3,A0,A1,A2,A3;
6 output CNTVC02,CNTVC02,READY,P0,P1,P2,P3,P4,P5,P6,P7;
7 reg S0,S1,S2,S3,S4,S5,S6,S7,S8,S9,S10,S11,S12,S13,S14;
8
9 wire
10 n66,n67_1,n68,n69,n78,n81,n83,n84,n85,n86,n87_1,n88,n89,n90,
11 n92_1,n93,n94,n95,n96,n97_1,n98,n100,n101,n102_1,n104,n105,n
12 106,n107_1,n108,n109,n110,n111,n112_1,n113,n114,n115,n116,n
13 117,n118,n119,n120,n121,n122,n123,n124,n125,n126,n127,n128,n
14 129,n130,n131,n132,n133,n134,n135,n136,n137,n138,n139,n140,n
15 141,n142,n143,n144,n145,n146,n147,n148,n149,n151,n152,n153,n
16 154,n155,n156,n157,n158,n159,n160,n161,n162,n163,n164,n165,n
17 166,n167,n169,n170,n171,n172,n173,n174,n175,n176,n177,n178,n
18 179,n180,n181,n182,n183,n184,n185,n187,n188,n189,n190,n191,n
19 192,n193,n194,n195,n196,n197,n198,n199,n200,n201,n202,n203,n
20 205,n206,n207,n208,n209,n210,n211,n212,n213,n214,n215,n216,n
21 217,n219,n220,n221,n222,n223,n224,n225,n226,n227,n228,n230,n
22 231,n232,n233,n234,n235,n236,n237,n238,n239,n241,n242,n243,n
23 244,n245,n246,n247,n248,n249,n250,n252,n253,n254,n255,n256,n
24 258,n259,n260,n261,n262,n264,n265,n266,n267,n268,n270,n271,n
25 272,n273,n274,NS0,NS1,NS2,NS3,NS4,NS5,NS6,NS8,NS7,NS9,NS10,NS
26 11,NS12,NS13,NS14;
27 not g0(n66,START);
28 not g1(n67_1,S0);
29 not n2(n68,S1);

```

SV/Verilog Design

Log

Share

testbench.v:6: warning: implicit definition of wire 'CNTVC02'.

testbench.v:6: warning: implicit definition of wire 'READY'.

testbench.v:6: warning: implicit definition of wire 'P0'.

testbench.v:6: warning: implicit definition of wire 'P1'.

testbench.v:6: warning: implicit definition of wire 'P2'.

testbench.v:6: warning: implicit definition of wire 'P3'.

testbench.v:6: warning: implicit definition of wire 'P4'.

testbench.v:6: warning: implicit definition of wire 'P5'.

testbench.v:6: warning: implicit definition of wire 'P6'.

testbench.v:6: warning: implicit definition of wire 'P7'.

Done

## Ex3.v (the target state is not reachable)

```
testbench.v
1 | timescale 1 ns / 1 ps
2 | module test;
3 |   reg
4 |   clock, Rdy1RtHs1, Rdy2RtHs1, Rdy1BmHs1, Rdy2BmHs1, InDoneHs1, RtTSHs1, TpArrayHs1, OutputHs1, Want
5 |   BmHs1, WantRtHs1, OutAvHs1, Full10Hs1, Full11Hs1, Prog_2, Prog_1, Prog_0;
6 |   ex3
7 |   x1(clock, Rdy1RtHs1, Rdy2RtHs1, Rdy1BmHs1, Rdy2BmHs1, InDoneHs1, RtTSHs1, TpArrayHs1, OutputHs1, W
8 |   antBmHs1, WantRtHs1, OutAvHs1, Full10Hs1, Full11Hs1, Prog_2, Prog_1, Prog_0, S26, S25, S22, S27, S14, S1
9 |   1, S17, S10, S13, S16, S6, S12, S19, S18, S15, S7, S9, S20, S28, S23, S8, S21, S24); //instantiate module
10 |   under test
11 |   initial begin
12 |     Rdy1RtHs1 = $random;
13 |     Rdy2RtHs1 = $random;
14 |     Rdy1BmHs1 = $random;
15 |     Rdy2BmHs1 = $random;
16 |     InDoneHs1 = $random;
17 |     RtTSHs1 = $random;
18 |     TpArrayHs1 = $random;
19 |     OutputHs1 = $random;
20 |     WantBmHs1 = $random;
21 |     WantRtHs1 = $random;
22 |     OutAvHs1 = $random;
23 |     Full10Hs1 = $random;
24 |     Full11Hs1 = $random;
25 |   end
26 | endmodule
```

```
design.v
1 // Code your design here
2 module
3 ex3(clock, Rdy1RtHs1, Rdy2RtHs1, Rdy1BmHs1, Rdy2BmHs1, InDoneHs1, RtTSHs1, TpAr
4 rayHs1, OutputHs1, WantBmHs1, WantRtHs1, OutAvHs1, Full10Hs1, Full11Hs1, Prog_2, P
5 rog_1, Prog_0, S26, S25, S22, S27, S14, S11, S17, S10, S13, S16, S6, S12, S19, S18, S15,
6 S7, S9, S20, S28, S23, S8, S21, S24);
7 input clock;
8 Rdy1RtHs1, Rdy2RtHs1, Rdy1BmHs1, Rdy2BmHs1, InDoneHs1, RtTSHs1, TpArrayHs1, Out
9 putHs1, WantBmHs1, WantRtHs1, OutAvHs1, Full10Hs1, Full11Hs1, Prog_2, Prog_1, Prog
10 _0;
11 output
12 S26, S25, S22, S27, S14, S11, S17, S10, S13, S16, S6, S12, S19, S18, S15, S7, S9, S20, S28
13 , S23, S8, S21, S24;
14 reg
15 S0, S1, S2, S3, S4, S5, S6, S7, S8, S9, S10, S11, S12, S13, S14, S15, S16, S17, S18, S19, S2
16 0, S21, S22, S23, S24, S25, S26, S27, S28;
17 wire
18 n127, n128, n129, n130_1, n131, n132, n133, n134_1, n135, n136, n137, n138_1, n139, n
19 140, n141, n142_1, n143, n144, n145, n146_1, n147, n148, n149, n150_1, n151, n152, n1
20 53, n154_1, n155, n156, n157, n158_1, n159, n160, n161, n163, n164, n165, n166_1, n16
21 7, n168, n169, n170_1, n171, n173, n174_1, n175, n176, n177, n178_1, n179, n180, n181
22 , n183_1, n182, n184, n185, n186_1, n187, n188, n189, n190, n191_1, n192, n193, n194, n195, n196_1, n197, n198, n199, n200, n201_1, n202, n203, n204, n205, n206_1, n207, n208, n209, n210, n211_1, n212, n213, n214, n215, n216_1, n217, n218, n219, n220, n221_1, n222, n223, n224, n225, n226_1, n227, n228, n229, n230, n231_1, n232, n233, n234, n235, n236, n237, n238, n239, n240, n241, n242, n243, n244, n245, n246, n247, n248, n249, n250, n251, n252, n253, n254, n255, n256, n257, n258, n259, n260, n261, n262, n263, n264, n265, n266, n267, n268, n269, n270, n271, n272, n273, n274, n275, n276, n277, n278, n279, n280, n281, n282, n283, n284, n285, n286, n287, n288, n289, n290, n291, n292, n293, n294, n295, n296, n297, n298, n299, n300, n301, n302, n303, n304, n305, n306, n307, n308, n309, n310, n311, n312, n313, n314, n315, n316, n317, n318, n319, n320, n321, n322, n323, n324, n325, n326, n327, n328, n329, n330, n331, n332, n333, n334, n335, n336, n337, n338, n339, n340, n341, n342, n343, n344, n345, n346, n347, n348, n349, n350, n351, n352, n353, n354, n355, n356, n357, n358, n359, n360, n361, n362, n363, n364, n365, n366, n367, n368, n369, n370, n371, n372, n373, n374, n375, n376, n377, n378, n379, n380, n381, n382, n383, n384, n385, n386, n387, n388, n389, n390, n391, n392, n393, n394, n395, n396, n397, n398, n399, n400, n401, n402, n403, n404, n405, n406, n407, n408, n409, n410, n411, n412, n413, n414, n415, n416, n417, n418, n419, n420, n421, n422, n423, n424, n425, n426, n427, n428, n429, n430, n431, n432, n433, n434, n435, n436, n437, n438, n439, n440, n441, n442, n443, n444, n445, n446, n447, n448, n449, n450, n451, n452, n453, n454, n455, n456, n457, n458, n459, n460, n461, n462, n463, n464, n465, n466, n467, n468, n469, n470, n471, n472, n473, n474, n475, n476, n477, n478, n479, n480, n481, n482, n483, n484, n485, n486, n487, n488, n489, n490, n491, n492, n493, n494, n495, n496, n497, n498, n499, n500, n501, n502, n503, n504, n505, n506, n507, n508, n509, n510, n511, n512, n513, n514, n515, n516, n517, n518, n519, n520, n521, n522, n523, n524, n525, n526, n527, n528, n529, n530, n531, n532, n533, n534, n535, n536, n537, n538, n539, n540, n541, n542, n543, n544, n545, n546, n547, n548, n549, n550, n551, n552, n553, n554, n555, n556, n557, n558, n559, n560, n561, n562, n563, n564, n565, n566, n567, n568, n569, n570, n571, n572, n573, n574, n575, n576, n577, n578, n579, n580, n581, n582, n583, n584, n585, n586, n587, n588, n589, n590, n591, n592, n593, n594, n595, n596, n597, n598, n599, n600, n601, n602, n603, n604, n605, n606, n607, n608, n609, n610, n611, n612, n613, n614, n615, n616, n617, n618, n619, n620, n621, n622, n623, n624, n625, n626, n627, n628, n629, n630, n631, n632, n633, n634, n635, n636, n637, n638, n639, n640, n641, n642, n643, n644, n645, n646, n647, n648, n649, n650, n651, n652, n653, n654, n655, n656, n657, n658, n659, n660, n661, n662, n663, n664, n665, n666, n667, n668, n669, n670, n671, n672, n673, n674, n675, n676, n677, n678, n679, n680, n681, n682, n683, n684, n685, n686, n687, n688, n689, n690, n691, n692, n693, n694, n695, n696, n697, n698, n699, n700, n701, n702, n703, n704, n705, n706, n707, n708, n709, n710, n711, n712, n713, n714, n715, n716, n717, n718, n719, n720, n721, n722, n723, n724, n725, n726, n727, n728, n729, n730, n731, n732, n733, n734, n735, n736, n737, n738, n739, n740, n741, n742, n743, n744, n745, n746, n747, n748, n749, n750, n751, n752, n753, n754, n755, n756, n757, n758, n759, n760, n761, n762, n763, n764, n765, n766, n767, n768, n769, n770, n771, n772, n773, n774, n775, n776, n777, n778, n779, n780, n781, n782, n783, n784, n785, n786, n787, n788, n789, n790, n791, n792, n793, n794, n795, n796, n797, n798, n799, n800, n801, n802, n803, n804, n805, n806, n807, n808, n809, n810, n811, n812, n813, n814, n815, n816, n817, n818, n819, n820, n821, n822, n823, n824, n825, n826, n827, n828, n829, n830, n831, n832, n833, n834, n835, n836, n837, n838, n839, n840, n841, n842, n843, n844, n845, n846, n847, n848, n849, n850, n851, n852, n853, n854, n855, n856, n857, n858, n859, n860, n861, n862, n863, n864, n865, n866, n867, n868, n869, n870, n871, n872, n873, n874, n875, n876, n877, n878, n879, n880, n881, n882, n883, n884, n885, n886, n887, n888, n889, n890, n891, n892, n893, n894, n895, n896, n897, n898, n899, n900, n901, n902, n903, n904, n905, n906, n907, n908, n909, n910, n911, n912, n913, n914, n915, n916, n917, n918, n919, n920, n921, n922, n923, n924, n925, n926, n927, n928, n929, n930, n931, n932, n933, n934, n935, n936, n937, n938, n939, n940, n941, n942, n943, n944, n945, n946, n947, n948, n949, n950, n951, n952, n953, n954, n955, n956, n957, n958, n959, n960, n961, n962, n963, n964, n965, n966, n967, n968, n969, n970, n971, n972, n973, n974, n975, n976, n977, n978, n979, n980, n981, n982, n983, n984, n985, n986, n987, n988, n989, n990, n991, n992, n993, n994, n995, n996, n997, n998, n999, n1000, n1001, n1002, n1003, n1004, n1005, n1006, n1007, n1008, n1009, n1010, n1011, n1012, n1013, n1014, n1015, n1016, n1017, n1018, n1019, n1020, n1021, n1022, n1023, n1024, n1025, n1026, n1027, n1028, n1029, n1030, n1031, n1032, n1033, n1034, n1035, n1036, n1037, n1038, n1039, n1040, n1041, n1042, n1043, n1044, n1045, n1046, n1047, n1048, n1049, n1050, n1051, n1052, n1053, n1054, n1055, n1056, n1057, n1058, n1059, n1060, n1061, n1062, n1063, n1064, n1065, n1066, n1067, n1068, n1069, n1070, n1071, n1072, n1073, n1074, n1075, n1076, n1077, n1078, n1079, n1080, n1081, n1082, n1083, n1084, n1085, n1086, n1087, n1088, n1089, n1090, n1091, n1092, n1093, n1094, n1095, n1096, n1097, n1098, n1099, n1100, n1101, n1102, n1103, n1104, n1105, n1106, n1107, n1108, n1109, n1110, n1111, n1112, n1113, n1114, n1115, n1116, n1117, n1118, n1119, n1120, n1121, n1122, n1123, n1124, n1125, n1126, n1127, n1128, n1129, n1130, n1131, n1132, n1133, n1134, n1135, n1136, n1137, n1138, n1139, n1140, n1141, n1142, n1143, n1144, n1145, n1146, n1147, n1148, n1149, n1150, n1151, n1152, n1153, n1154, n1155, n1156, n1157, n1158, n1159, n1160, n1161, n1162, n1163, n1164, n1165, n1166, n1167, n1168, n1169, n1170, n1171, n1172, n1173, n1174, n1175, n1176, n1177, n1178, n1179, n1180, n1181, n1182, n1183, n1184, n1185, n1186, n1187, n1188, n1189, n1190, n1191, n1192, n1193, n1194, n1195, n1196, n1197, n1198, n1199, n1200, n1201, n1202, n1203, n1204, n1205, n1206, n1207, n1208, n1209, n1210, n1211, n1212, n1213, n1214, n1215, n1216, n1217, n1218, n1219, n1220, n1221, n1222, n1223, n1224, n1225, n1226, n1227, n1228, n1229, n1230, n1231, n1232, n1233, n1234, n1235, n1236, n1237, n1238, n1239, n1240, n1241, n1242, n1243, n1244, n1245, n1246, n1247, n1248, n1249, n1250, n1251, n1252, n1253, n1254, n1255, n1256, n1257, n1258, n1259, n1260, n1261, n1262, n1263, n1264, n1265, n1266, n1267, n1268, n1269, n1270, n1271, n1272, n1273, n1274, n1275, n1276, n1277, n1278, n1279, n1280, n1281, n1282, n1283, n1284, n1285, n1286, n1287, n1288, n1289, n1290, n1291, n1292, n1293, n1294, n1295, n1296, n1297, n1298, n1299, n1300, n1301, n1302, n1303, n1304, n1305, n1306, n1307, n1308, n1309, n1310, n1311, n1312, n1313, n1314, n1315, n1316, n1317, n1318, n1319, n1320, n1321, n1322, n1323, n1324, n1325, n1326, n1327, n1328, n1329, n1330, n1331, n1332, n1333, n1334, n1335, n1336, n1337, n1338, n1339, n1340, n1341, n1342, n1343, n1344, n1345, n1346, n1347, n1348, n1349, n1350, n1351, n1352, n1353, n1354, n1355, n1356, n1357, n1358, n1359, n1360, n1361, n1362, n1363, n1364, n1365, n1366, n1367, n1368, n1369, n1370, n1371, n1372, n1373, n1374, n1375, n1376, n1377, n1378, n1379, n1380, n1381, n1382, n1383, n1384, n1385, n1386, n1387, n1388, n1389, n1390, n1391, n1392, n1393, n1394, n1395, n1396, n1397, n1398, n1399, n1400, n1401, n1402, n1403, n1404, n1405, n1406, n1407, n1408, n1409, n1410, n1411, n1412, n1413, n1414, n1415, n1416, n1417, n1418, n1419, n1420, n1421, n1422, n1423, n1424, n1425, n1426, n1427, n1428, n1429, n1430, n1431, n1432, n1433, n1434, n1435, n1436, n1437, n1438, n1439, n1440, n1441, n1442, n1443, n1444, n1445, n1446, n1447, n1448, n1449, n1450, n1451, n1452, n1453, n1454, n1455, n1456, n1457, n1458, n1459, n1460, n1461, n1462, n1463, n1464, n1465, n1466, n1467, n1468, n1469, n1470, n1471, n1472, n1473, n1474, n1475, n1476, n1477, n1478, n1479, n1480, n1481, n1482, n1483, n1484, n1485, n1486, n1487, n1488, n1489, n1490, n1491, n1492, n1493, n1494, n1495, n1496, n1497, n1498, n1499, n1500, n1501, n1502, n1503, n1504, n1505, n1506, n1507, n1508, n1509, n1510, n1511, n1512, n1513, n1514, n1515, n1516, n1517, n1518, n1519, n1520, n1521, n1522, n1523, n1524, n1525, n1526, n1527, n1528, n1529, n1530, n1531, n1532, n1533, n1534, n1535, n1536, n1537, n1538, n1539, n1540, n1541, n1542, n1543, n1544, n1545, n1546, n1547, n1548, n1549, n1550, n1551, n1552, n1553, n1554, n1555, n1556, n1557, n1558, n1559, n1560, n1561, n1562, n1563, n1564, n1565, n1566, n1567, n1568, n1569, n1570, n1571, n1572, n1573, n1574, n1575, n1576, n1577, n1578, n1579, n1580, n1581, n1582, n1583, n1584, n1585, n1586, n1587, n1588, n1589, n1590, n1591, n1592, n1593, n1594, n1595, n1596, n1597, n1598, n1599, n1600, n1601, n1602, n1603, n1604, n1605, n1606, n1607, n1608, n1609, n1610, n1611, n1612, n1613, n1614, n1615, n1616, n1617, n1618, n1619, n1620, n1621, n1622, n1623, n1624, n1625, n1626, n1627, n1628, n1629, n1630, n1631, n1632, n1633, n1634, n1635, n1636, n1637, n1638, n1639, n1640, n1641, n1642, n1643, n1644, n1645, n1646, n1647, n1648, n1649, n1650, n1651, n1652, n1653, n1654, n1655, n1656, n1657, n1658, n1659, n1660, n1661, n1662, n1663, n1664, n1665, n1666, n1667, n1668, n1669, n1670, n1671, n1672, n1673, n1674, n1675, n1676, n1677, n1678, n1679, n1680, n1681, n1682, n1683, n1684, n1685, n1686, n1687, n1688, n1689, n1690, n1691, n1692, n1693, n1694, n1695, n1696, n1697, n1698, n1699, n1700, n1701, n1702, n1703, n1704, n1705, n1706, n1707, n1708, n1709, n1710, n1711, n1712, n1713, n1714, n1715, n1716, n1717, n1718, n1719, n1720, n1721, n1722, n1723, n1724, n1725, n1726, n1727, n1728, n1729, n1730, n1731, n1732, n1733, n1734, n1735, n1736, n1737, n1738, n1739, n1740, n1741, n1742, n1743, n1744, n1745, n1746, n1747, n1748, n1749, n1750, n1751, n1752, n1753, n1754, n1755, n1756, n1757, n1758, n1759, n1760, n1761, n1762, n1763, n1764, n1765, n1766, n1767, n1768, n1769, n1770, n1771, n1772, n1773, n1774, n1775, n1776, n1777, n1778, n1779, n1780, n1781, n1782, n1783, n1784, n1785, n1786, n1787, n1788, n1789, n1790, n1791, n1792, n1793, n1794, n1795, n1796, n1797, n1798, n1799, n1800, n1801, n1802, n1803, n1804, n1805, n1806, n1807, n1808, n1809, n1810, n1811, n1812, n1813, n1814, n1815, n1816, n1817, n1818, n1819, n1820, n1821, n1822, n1823, n1824, n1825, n1826, n1827, n1828, n1829, n1830, n1831, n1832, n1833, n1834, n1835, n1836, n1837, n1838, n1839, n1840, n1841, n1842, n1843, n1844, n1845, n1846, n1847, n1848, n1849, n1850, n1851, n1852, n1853, n1854, n1855, n1856, n1857, n1858, n1859, n1860, n1861, n1862, n1863, n1864, n1865, n1866, n1867, n1868, n1869, n1870, n1871, n1872, n1873, n1874, n1875, n1876, n1877, n1878, n1879, n1880, n1881, n1882, n1883, n1884, n1885, n1886, n1887, n1888, n1889, n1890, n1891, n1892, n1893, n1894, n1895, n1896, n1897, n1898, n1899, n1900, n1901, n1902, n1903, n1904, n1905, n1906, n1907, n1908, n1909, n1910, n1911, n1912, n1913, n1914, n1915, n1916, n1917, n1918, n1919, n1920, n1921, n1922, n1923, n1924, n1925, n1926, n1927, n1928, n1929, n1930, n1931, n1932, n1933, n1934, n1935, n1936, n1937, n1938, n1939, n1940, n1941, n1942, n1943, n1944, n1945, n1946, n1947, n1948, n1949, n1950, n1951, n1952, n1953, n1954, n1955, n1956, n1957, n1958, n1959, n1960, n1961, n1962, n1963, n1964, n1965, n1966, n1967, n1968, n1969, n1970, n1971, n1972, n1973, n1974, n1975, n1976, n1977, n1978, n1979, n1980, n1981, n1982, n1983, n1984, n1985, n1986, n1987, n1988, n1989, n1990, n1991, n1992, n1993, n1994, n1995, n1996, n1997, n1998, n1999, n2000, n2001, n2002, n2003, n2004, n2005, n2006, n2007, n2008, n2009, n2010, n2011, n2012, n2013, n2014, n2015, n2016, n2017, n2018, n2019, n2020, n2021, n2022, n2023, n2024, n2025, n2026, n2027, n2028, n2029, n2030, n2031, n2032, n2033, n2034, n2035, n2036, n2037, n2038, n2039, n2040, n2041, n2042, n2043, n2044, n2045, n2046, n2047, n2048, n2049, n2050, n2051, n2052, n2053, n2054, n2055, n2056, n2057, n2058, n2059, n2060, n2061, n2062, n2063, n2064, n2065, n2066, n2067, n2068, n2069, n2070, n2071, n2072, n2073, n2074, n2075, n2076, n2077, n2078, n2079, n2080, n2081, n2082, n2083, n2084, n2085, n2086, n2087, n2088, n2089, n2090, n2091, n2092, n2093, n2094, n2095, n2096, n2097, n2098, n2099, n2100, n2101, n2102, n2103, n2104, n2105, n2106, n2107, n2108, n2109, n2110, n2111, n2
```

## Stoplight1.v (the target state is reachable with about 400 cycles of simulation)

The screenshot shows a Verilog simulation environment with two panels: **testbench.v** and **design.v**.

**testbench.v** (SV/Verilog Testbench):

```
8 initial begin
9     x1.S3 = 0;
10    x1.S2 = 0;
11    x1.S1 = 0; //set initial state inside module
12    x1.S0 = 0;
13    CLK = 0;
14    Ped = $random;
15    #1000000 $finish; //how long to run before stopping
16 end
17
18 always #0.5 CLK=~CLK; //clock toggle every 0.5 ns
19 always #200 x1.S3 = 0;
20 always #200 x1.S2 = 0;
21 always #200 x1.S1 = 0;
22 always #200 x1.S0 = 0;
23 always #200 Ped = $random;
24
25 always @(posedge CLK) begin //each cycle, print state and set next input
26     if ((x1.S3,x1.S2,x1.S1,x1.S0) == 4'b0100) begin //did we hit target state?
27         $display("%8d",$time," state:",x1.S3,x1.S2,x1.S1,x1.S0," SigGVR:",x1.SigG,x1.SigY,x1.SigR,"
28         (reached target)");
29     end
30 endmodule
31
```

**design.v** (SV/Verilog Design):

```
1 module stoplight(Ped,clock,SigG,SigY,SigR);
2 input Ped,clock;
3 output SigG,SigY,SigR;
4 reg S3,S2,S1,S0;
5 wire
6     n13,n14,n15,n16,n17,n18,n19,n20,n21,n22,n23,n24,n25,n26,n27,
7     n28,n29,n30,n31,n32,n33,n34,n35,n36,n38,n39,n40,n42,n41,n43,
8     n44,n45,n47,n48,n49,n50,n52,n53,n54,n55,n56,n57,n58,n59,n60,
9     n61,n62,n64,n65,n66,n67,NS0,NS1,NS2,NS3;
10 not g0(n13,S3);
11 and g1(n14,S2,n13);
12 and g2(n15,Ped,S0);
13 and g3(n16,n15,n14);
14 not g4(n17,n16);
15 and g5(n18,S0,S1);
16 and g6(n19,n18,n14);
17 not g7(n20,n19);
18 and g8(n21,n20,n17);
19 not g9(n22,S0);
20 and g10(n23,n14,n22);
21 not g11(n24,n23);
22 not g12(n25,S2);
23 and g13(n26,n25,n13);
24 and g14(n27,n26,n18);
25 not n15(n28,n27);

```

**Log** (Share):

```
[2020-02-15 12:04:36 EST] iverilog '-wall' '-g2012' design.v testbench.v && unbuffer vvp a.out
testbench.v:4: warning: Some modules have no timescale. This may cause
testbench.v:4:      : confusing timing results. Affected modules are:
testbench.v:4:      : -- module stoplight declared here: design.v:1
testbench.v:7: warning: implicit definition of wire 'SigG'.
testbench.v:7: warning: implicit definition of wire 'SigY'.
testbench.v:7: warning: implicit definition of wire 'SigR'.
5 state:0100 SigGVR:100 (reached target)
1405 state:0100 SigGVR:100 (reached target)
2005 state:0100 SigGVR:100 (reached target)
2605 state:0100 SigGVR:100 (reached target)
3005 state:0100 SigGVR:100 (reached target)
3405 state:0100 SigGVR:100 (reached target)

```

## Stoplight2.v (the target state is not reachable)

The screenshot shows a Verilog simulation environment with two panels: **testbench.v** and **design.v**.

**testbench.v** (SV/Verilog Testbench):

```
1 // Code your testbench here
2 // or browse Examples
3 timescale 1 ns / 1 ps
4 module test;
5     reg Ped,CLK;
6     stoplight2 x1(Ped,CLK,SigG,SigY,SigR); //instantiate module under test
7     initial begin
8         x1.S4 = 0;
9         x1.S3 = 0;
10        x1.S2 = 0;
11        x1.S1 = 0; //set initial state inside module
12        x1.S0 = 0;
13        CLK = 0;
14        Ped = $random;
15        #1000000 $finish; //how long to run before stopping
16    end
17
18    always #0.5 CLK=~CLK; //clock toggle every 0.5 ns
19    always #200 x1.S4 = 0;
20    always #200 x1.S3 = 0;
21    always #200 x1.S2 = 0;
22    always #200 x1.S1 = 0;
23    always #200 x1.S0 = 0;
24    always #200 Ped = $random;
25
```

**design.v** (SV/Verilog Design):

```
1 module stoplight2(Ped,clock,SigG,SigY,SigR);
2 input Ped,clock;
3 reg S4,S3,S2,S1,S0;
4 output SigG,SigY,SigR;
5 wire
6     n15,n16,n17,n18,n19,n20,n21,n22,n23,n24,n25,n26,n27,n28,n29,n30,n31,n32,n33,n34,n
7     35,n36,n37,n38,n39,n40,n41,n42,n43,n44,n45,n46,n47,n48,n49,n50,n51,n52,n53,n54,n5
8     5,n57,n58,n60,n61,n62,n63,n64,n65,n67,n68,n69,n70,n71,n72,n73,n74,n75,n77,n78,n79
9     ,n80,n81,n82,n83,n84,n85,n86,n88,n89,n90,n91,n92,n93,n94,n95,n96,n97,n99,n100,n10
10    1,n102,n103,n104,n105,n106,n107,n108,n109,n110,n111,n112,n113,n114,n116,n117,n118
11    ,n119,n120,NS4,NS3,NS2,NS1,NS0;
12 not g0(n15,S0);
13 not g1(n16,S4);
14 not g2(n17,S3);
15 and g3(n18,n17,n16);
16 and g4(n19,n18,S2);
17 and g5(n20,n19,n15);
18 not g6(n21,n20);
19 not g7(n22,S1);
20 not g8(n23,Ped);
21 and g9(n24,n23,n22);
22 and g10(n25,n24,n19);
23 not g11(n26,n25);
24 and g12(n27,Ped,S0);
25 and n13(n28,n27,n19);

```

**Log** (Share):

```
[2020-02-15 12:12:06 EST] iverilog '-wall' '-g2012' design.v testbench.v && unbuffer vvp a.out
testbench.v:4: warning: Some modules have no timescale. This may cause
testbench.v:4:      : confusing timing results. Affected modules are:
testbench.v:4:      : -- module stoplight2 declared here: design.v:1
testbench.v:7: warning: implicit definition of wire 'SigG'.
testbench.v:7: warning: implicit definition of wire 'SigY'.
testbench.v:7: warning: implicit definition of wire 'SigR'.
Done

```



**5. Are your results from part A.4 consistent with your results from A.2? Describe this. If the results are consistent, give a precise explanation of what results would show them to be incompatible with each other.**

The results are consistent. An example of an inconsistent result would be if the testbench of Ex.1 saying that state 11 is reachable.

## Part B

**1. Use your program to generate the CNF files for ex1 with the transition relation unrolled twice. Be sure to add the initial state of 0 and be sure to enforce that the next state of each transition relation is equal to the starting state of the next transition relation. Use the SAT solver to check whether state 11 is reachable as the 2nd state after the initial one. Is your finding consistent with the state transition graph from question A.1? Explain your answer. Now that you've checked whether "11" is reachable in 2 cycles, modify the example and repeat, to check whether the other 3 states (00,01,10) are reachable in 2 cycles, and again explain whether your findings are consistent with question A.1. Be sure to fix any bugs here before trying the larger examples that follow!**

```
alanchen@DESKTOP-TM62I39:/mnt/c/Users/Alan Chen/Desktop/ECE597/Lab 1/PythonFiles/Part2$ python3.6 main.py verilogFiles/ex1.v 2
THIS IS THE 1 UNROLL RESULTS:
FINAL STATE: 11
s UNSATISFIABLE

THIS IS THE 2 UNROLL RESULTS:
FINAL STATE: 11
s UNSATISFIABLE
```

```
alanchen@DESKTOP-TM62I39:/mnt/c/Users/Alan Chen/Desktop/ECE597/Lab 1/PythonFiles/Part2$ python3.6 main.py verilogFiles/ex1.v 2
THIS IS THE 1 UNROLL RESULTS:
FINAL STATE: 00
s UNSATISFIABLE

THIS IS THE 2 UNROLL RESULTS:
FINAL STATE: 00
s UNSATISFIABLE
```

```
alanchen@DESKTOP-TM62I39:/mnt/c/Users/Alan Chen/Desktop/ECE597/Lab 1/PythonFiles/Part2$ python3.6 main.py verilogFiles/ex1.v 2
THIS IS THE 1 UNROLL RESULTS:
FINAL STATE: 01
s SATISFIABLE
v -1 -2 -3 -4 5 6 -7 0

THIS IS THE 2 UNROLL RESULTS:
FINAL STATE: 01
s SATISFIABLE
v -1 -2 3 -4 -5 -6 -7 8 -9 10 11 12 -13 -14 0
```

```
alanchen@DESKTOP-TM62I39:/mnt/c/Users/Alan Chen/Desktop/ECE597/Lab 1/PythonFiles/Part2$ python3.6 main.py verilogFiles/ex1.v 2
THIS IS THE 1 UNROLL RESULTS:
FINAL STATE: 10
s UNSATISFIABLE

THIS IS THE 2 UNROLL RESULTS:
FINAL STATE: 10
s UNSATISFIABLE
```



```

a1anchen@DESKTOP-TM62I39:/mnt/c/Users/Alan Chen/Desktop/ECE597/Lab 1/PythonFiles/Part2$ python3.6 main.py verilogFiles/stoplight1.v 10
THIS IS THE 10 UNROLL RESULTS:
TESTING FOR FINAL STATE: 0100
s UNSATISFIABLE
My program took 0.030384302139282227 seconds to run

a1anchen@DESKTOP-TM62I39:/mnt/c/Users/Alan Chen/Desktop/ECE597/Lab 1/PythonFiles/Part2$ python3.6 main.py verilogFiles/stoplight2.v 10
THIS IS THE 10 UNROLL RESULTS:
TESTING FOR FINAL STATE: 01000
s UNSATISFIABLE
My program took 0.048485755920410156 seconds to run

```

**3. Compare your results using SAT-based reachability to the results from random simulation. Explain any differences in results and what causes them. Are there any limitations in SAT-based reachability due to only checking whether a state is reachable in the last unrolling and not intermediate ones? How might you improve the symbolic search to avoid this limitation?**

There are some cases where the SAT-based reachability results differ from the Verilog testbench results like in ex3 Verilog file. This is because the SAT is an exhaustive test whereas my Verilog testbenches use random inputs. There may be times when we get unlucky with the random generator especially if it requires lot of random input (looking for a specific input out of all combinations). The limitation of a SAT solver depends on how you write the your DIMACS file. If there is a problem with only checking whether a state is reachable in the last unrolling and not intermediate ones, then you can write your DIMACS file to be NS0\_1 or NS0\_2 or NS0\_3 depending on what you are search for. Another way is to check your state at unrolling 1, unrolling 2, ..., up to the last unrolling.

**4. Submit the source code of your program as a separate file with your report and include instructions for how to run it for testing.**

Should be attached, if not :

[https://github.com/achen173/Modeling\\_Embedding\\_Systems](https://github.com/achen173/Modeling_Embedding_Systems)



**5. For the benchmark stoplight1.v, check whether the target state is reachable as the  $i$ th state after the initial state for  $i=1,2,\dots,32$ . This should be done by invoking your program (and the SAT solver) 32 times while changing the input argument that specifies the number of unrollings. You can do this using a script. Report the values of  $i$  for which the target state is reachable. How does this compare with your results from A.2?**

```
alanchen@DESKTOP-TM62I39:/mnt/c/Users/Alan Chen/Desktop/ECE597/Lab 1/PythonFiles/Part2$ python3.6 main.py -c verilogFiles/stoplight1.v 32

THIS IS THE 1 UNROLL RESULTS:
TESTING FOR FINAL STATE: 0100
s UNSATISFIABLE

THIS IS THE 2 UNROLL RESULTS:
TESTING FOR FINAL STATE: 0100
s UNSATISFIABLE

THIS IS THE 3 UNROLL RESULTS:
TESTING FOR FINAL STATE: 0100
s UNSATISFIABLE

THIS IS THE 4 UNROLL RESULTS:
TESTING FOR FINAL STATE: 0100
s SATISFIABLE
v -1 -2 -3 -4 -5 -6 -7 8 -9 -10 11 -12 -13 -14 15 16 17 18 19 20 -21 -22 -23
v -24 -25 26 -27 -28 -29 30 -31 -32 -33 -34 -35 -36 37 38 39 40 -41 -42 -43 44
v -45 -46 -47 -48 49 50 51 52 53 54 55 56 57 -58 59 -60 -61 -62 -63 -64 65 66
v 67 68 69 70 71 72 73 74 75 76 -77 -78 -79 80 81 82 83 -84 85 86 -87 -88 -89
v 90 -91 92 -93 94 -95 -96 -97 -98 -99 -100 101 102 103 104 105 106 107 -108
v 109 110 111 -112 113 114 115 -116 -117 -118 -119 120 -121 -122 -123 -124 -125
v -126 -127 -128 -129 130 -131 -132 -133 134 -135 -136 137 -138 139 140 141
v -142 143 -144 -145 146 -147 148 -149 -150 -151 152 153 154 155 -156 -157 -158
v -159 -160 161 162 163 164 165 166 167 168 -169 -170 -171 -172 -173 -174 -175
v 176 -177 -178 179 -180 181 182 -183 184 185 -186 -187 188 -189 -190 191 192
v -193 -194 195 -196 197 198 -199 200 -201 -202 -203 -204 -205 -206 -207 -208
v 209 210 211 212 213 214 -215 216 217 -218 -219 220 -221 222 223 -224 225 226
v -227 228 229 230 -231 232 -233 234 -235 236 -237 238 -239 240 241 -242 243
v -244 0

THIS IS THE 5 UNROLL RESULTS:
TESTING FOR FINAL STATE: 0100
s UNSATISFIABLE

THIS IS THE 6 UNROLL RESULTS:
TESTING FOR FINAL STATE: 0100
s UNSATISFIABLE

THIS IS THE 7 UNROLL RESULTS:
TESTING FOR FINAL STATE: 0100
s UNSATISFIABLE
```

i-Values = 4, 12 – 32

My answer is consistent with my results from A.2. It is reachable at exactly 4 clock cycles (minimum). However, it is also reachable with anything above 12 clock cycle.

**6. Repeat previous question using benchmark stoplight2.v. How does this compare to your finding from random execution of stoplight2.v in A.4?**

```
alanchen@DESKTOP-TM62I39:/mnt/c/Users/Alan Chen/Desktop/ECE597/Lab 1/PythonFiles/Part2$ python3.6 main.py -c verilogFiles/stoplight2.v 32
THIS IS THE 1 UNROLL RESULTS:
TESTING FOR FINAL STATE: 01000
s UNSATISFIABLE

THIS IS THE 2 UNROLL RESULTS:
TESTING FOR FINAL STATE: 01000
s UNSATISFIABLE

THIS IS THE 3 UNROLL RESULTS:
TESTING FOR FINAL STATE: 01000
s UNSATISFIABLE

THIS IS THE 4 UNROLL RESULTS:
TESTING FOR FINAL STATE: 01000
s UNSATISFIABLE

THIS IS THE 5 UNROLL RESULTS:
TESTING FOR FINAL STATE: 01000
s UNSATISFIABLE

THIS IS THE 6 UNROLL RESULTS:
TESTING FOR FINAL STATE: 01000
s UNSATISFIABLE

THIS IS THE 7 UNROLL RESULTS:
TESTING FOR FINAL STATE: 01000
s UNSATISFIABLE

THIS IS THE 8 UNROLL RESULTS:
TESTING FOR FINAL STATE: 01000
s UNSATISFIABLE

THIS IS THE 9 UNROLL RESULTS:
TESTING FOR FINAL STATE: 01000
s UNSATISFIABLE
```

i-Values = 15 – 32

My results are inconsistent with the random execution of stoplight2.v from A.4 because the testbench use random input. With the # of unrolling to succeed (15-32), the odds for a correct random input decrease. Therefore, it is expected to be inconsistent.

**7. Given the number of state bits in the stoplight designs, is it correct to conclude that any states not reachable within 32 unrollings will remain unreachable with more unrollings? Justify your answer.**

Yes, because there are in stoplight has a maximum of 5-bit state, so  $2^5 = 32$  possibilities. As a result, the 32 unrolling's covers all possibilities.