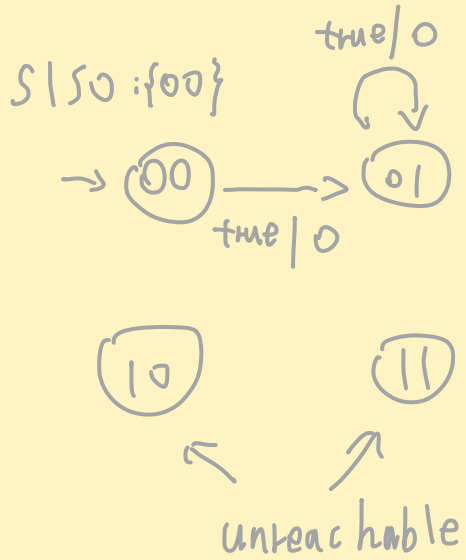
lab1. Wang Junfei 3300688796

Part A.

(1) For ex1.v

S0	S1	A	X1	Y	NS0	NS1
0	0	0	0	0	1	0
0	0	1	0	0	1	0
0	1	0	0	0	1	0
0	1	1	0	0	1	0
1	0	0	0	0	1	0
1	0	1	0	0	1	0
1	1	0	1	0	0	0
1	1	1	1	1	0	1



X1 = S0 ∧ S1

Y = X1 ∧ A

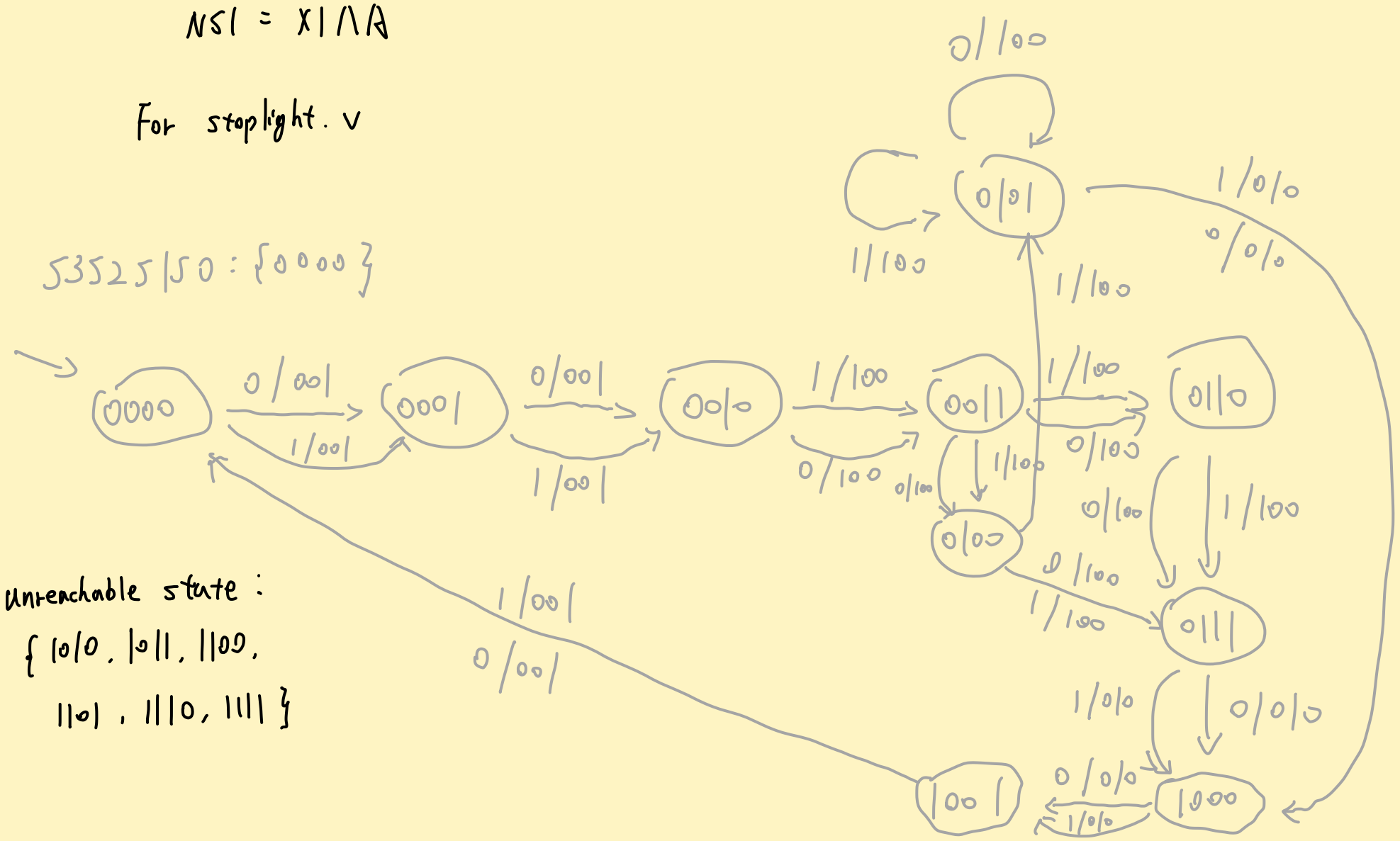
NS0 = ¬X1

NS1 = X1 ∧ A

For ex1.v, I draw state transition graph by writting the state transition table.

For stoplight.v

For stoplight1.v, the drawing of stoplight1 is very complex. so I use testbench instead of state transition table. according to the transchipt in ModelSim. I draw the state transition graph.



unreachable state:

{ 1010, 1011, 1100, 1101, 1110, 1111 }

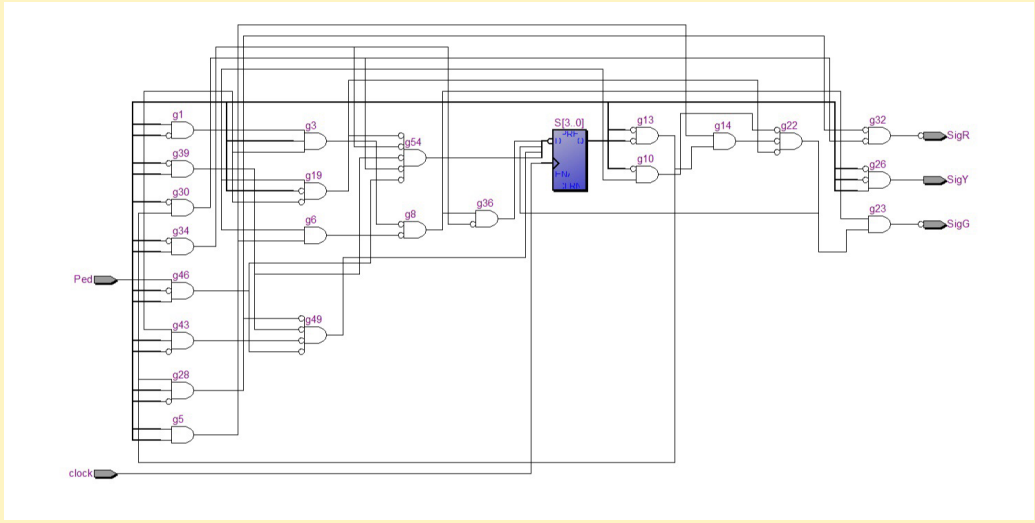
Transcript

File Edit View Bookmarks Window Help

Transcript

```
#    95010 Input: 0 Output: 100 state:0011
#    95011 Input: 1 Output: 100 state:0100  (reached target)
#    95012 Input: 0 Output: 100 state:0111
#    95013 Input: 1 Output: 010 state:1000
#    95014 Input: 0 Output: 010 state:1001
#    95015 Input: 0 Output: 001 state:0000
#    95016 Input: 0 Output: 001 state:0001
#    95017 Input: 0 Output: 001 state:0010
#    95018 Input: 0 Output: 100 state:0011
#    95019 Input: 0 Output: 100 state:0100  (reached target)
#    95020 Input: 1 Output: 100 state:0101
#    95021 Input: 1 Output: 010 state:1000
#    95022 Input: 1 Output: 010 state:1001
#    95023 Input: 0 Output: 001 state:0000
#    95024 Input: 0 Output: 001 state:0001
#    95025 Input: 0 Output: 001 state:0010
#    95026 Input: 1 Output: 100 state:0011
#    95027 Input: 1 Output: 100 state:0110
#    95028 Input: 1 Output: 100 state:0111
#    95029 Input: 0 Output: 010 state:1000
#    95030 Input: 0 Output: 010 state:1001
#    95031 Input: 1 Output: 001 state:0000
#    95032 Input: 1 Output: 001 state:0001
#    95033 Input: 1 Output: 001 state:0010
#    95034 Input: 1 Output: 100 state:0011
#    95035 Input: 1 Output: 100 state:0110
#    95036 Input: 1 Output: 100 state:0111
#    95037 Input: 1 Output: 010 state:1000
#    95038 Input: 0 Output: 010 state:1001
#    95039 Input: 1 Output: 001 state:0000
#    95040 Input: 0 Output: 001 state:0001
#    95041 Input: 1 Output: 001 state:0010
#    95042 Input: 0 Output: 100 state:0011
#    95043 Input: 0 Output: 100 state:0100  (reached target)
#    95044 Input: 1 Output: 100 state:0101
#    95045 Input: 0 Output: 010 state:1000
#    95046 Input: 0 Output: 010 state:1001
#    95047 Input: 0 Output: 001 state:0000
#    95048 Input: 0 Output: 001 state:0001
#    95049 Input: 0 Output: 001 state:0010
```

Transcript for stoplight1.v



RTL viewer for stoplight1.V

(2)

For ex1.v, target state is unreachable.

For stoplight1.v, target state is reachable, shortest path: {0000, 0001, 0010, 0011, 0100} 4 clock cycles just mentioned with transcript for stolight1.v in (1). I just see the transcript and find the shortest path.

(3).

For stoplight1.v :

```
run -all
      13 Input: 1 Output: 100 state: 0100  (reached target)
      13 clock cycle in first received:
```

For stoplight2.v :

```
un -all
   1358 Input: 1 Output: 010 state: 01000  (reached target)
   1358 clock cycle in first received:
```

For ex1.v :

```
run -all
    timed out
** Note: $finish    : D:/FPGA/Project/umass_labl/prj/../tb/stoplight_test.v(13)
   Time: 100 us  Iteration: 0  Instance: /stoplight_test
1
```

For ex2.v :

```
run -all
    timed out
** Note: $finish    : D:/FPGA/Project/umass_labl/prj/../tb/stoplight_test.v(35)
   Time: 100 us  Iteration: 0  Instance: /stoplight_test
```

For ex3.v :

```
run -all
    timed out
** Note: $finish    : D:/FPGA/Project/umass_labl/prj/../tb/stoplight_test.v(56)
   Time: 100 us  Iteration: 0  Instance: /stoplight_test
1
```

For ex4.v ;

```
#     timed out
# ** Note: $finish    : D:/FPGA/Project/umass_labl/prj/../tb/stoplight_test.v(79)
#    Time: 100 us  Iteration: 0  Instance: /stoplight_test
```

Testbench for stoplight2.v :

```
`timescale 1 ns / 1 ps
module stoplight_test;
   reg clock;
   reg Ped;
   wire SigG,SigY,SigR;
   stoplight X1(Ped,clock,SigG,SigY,SigR); //instantiate module under test
   initial begin
      //set initial state inside module
      X1.S0 = 0;
      X1.S1 = 0;
      X1.S2 = 0;
      X1.S3 = 0;
      X1.S4 = 0;
      clock = 0;
      Ped = 0;
      $dumpfile(" stoplight_test.vcd"); //print for waveform viewing
      $dumpvars(0, stoplight_test);
      #100_000 $finish;  //how long to run before stopping
   end

   always #0.5 clock=~clock; //clock toggle every 0.5 ns

   always @(posedge clock) begin //each cycle, print state and set next input
      if ({X1.S4,X1.S3,X1.S2,X1.S1,X1.S0} == {5'b01000}) begin //did we hit target state?
    $display("%8d",$time," Input: ",Ped, " Output: ",SigG,SigY,SigR, " state: ",X1.S4,X1.S3,X1.S2,X1.S1,X1.S0, "  (reached target)");
    $display("%8d clock cycle in first received ",$time);
    $finish;
    //$finish; //stop if reached target state
      end
      else if($time == 100_000) begin
    $display("    timed out");
      end
      Ped = $random;
   end
endmodule
```

(4) for stoplight1. we get the shortest path is 5 clock cycles. but in A.3 the simulation show the answer is 13 clock cycles. Simply because A.3 show the first reaching it clock cycles but A.2 show the path with the fewest transitions. In order to fix the mistake, we can display the all states transition relation in transchipts and observe the shortest path with initial state being 00000 and targe state being 01000.

Part B

(1)

```
skyline@skyline-VirtualBox: /media/sf_sharedir/lab1
skyline@skyline-VirtualBox:/media/sf_sharedir/lab1$ python3.7 solveReachability.py data/ex1.v -c 2 11

 1  UNROLL RESULTS:
s UNSATISFIABLE
runtime:  0.011683940887451172 seconds

 2  UNROLL RESULTS:
s UNSATISFIABLE
runtime:  0.02252817153930664 seconds
skyline@skyline-VirtualBox:/media/sf_sharedir/lab1$ python3.7 solveReachability.py data/ex1.v -c 2 01

 1  UNROLL RESULTS:
s SATISFIABLE
v -1 -2 -3 -4 5 6 -7 0
runtime:  0.01692056655883789 seconds

 2  UNROLL RESULTS:
s SATISFIABLE
v -1 -2 3 -4 -5 -6 -7 8 -9 10 11 12 -13 -14 0
runtime:  0.025813579559326172 seconds
skyline@skyline-VirtualBox:/media/sf_sharedir/lab1$ python3.7 solveReachability.py data/ex1.v -c 2 10

 1  UNROLL RESULTS:
s UNSATISFIABLE
runtime:  0.01301121711730957 seconds

 2  UNROLL RESULTS:
s UNSATISFIABLE
runtime:  0.022869586944580078 seconds
skyline@skyline-VirtualBox:/media/sf_sharedir/lab1$ python3.7 solveReachability.py data/ex1.v -c 2 00

 1  UNROLL RESULTS:
s UNSATISFIABLE
runtime:  0.012392997741699219 seconds

 2  UNROLL RESULTS:
s UNSATISFIABLE
runtime:  0.019263029098510742 seconds
skyline@skyline-VirtualBox:/media/sf_sharedir/lab1$
```

obviously, my finding consistent with the state transition graph from question A.1. only state 01 is reachable from initial state 00 as 1st and 2nd state.

(2)

```
skyline@skyline-VirtualBox:/media/sf_sharedir/lab1$ python3.7 solveReachability.py data/ex1.v 10 11

 10  UNROLL RESULTS:
s UNSATISFIABLE
untime:  0.013088464736938477 seconds
skyline@skyline-VirtualBox:/media/sf_sharedir/lab1$ python3.7 solveReachability.py data/ex2.v 10 111111111111111

 10  UNROLL RESULTS:
s UNSATISFIABLE
untime:  0.1803908348083496 seconds
skyline@skyline-VirtualBox:/media/sf_sharedir/lab1$ python3.7 solveReachability.py data/ex3.v 10 00000000010110110000011000001

 10  UNROLL RESULTS:
s SATISFIABLE
v -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 -11 -12 -13 -14 -15 -16 -17 -18 -19 -20 -21
v -22 -23 -24 -25 -26 -27 -28 -29 -30 -31 32 33 -34 35 -36 -37 -38 39 -40 -41
v -42 -43 -44 45 -46 -47 48 49 -50 -51 -52 -53 -54 -55 -56 57 -58 -59 -60 61
v -62 -63 -64 -65 -66 -67 -68 -69 -70 -71 -72 -73 -74 -75 -76 -77 -78 -79 -80
v -81 -82 -83 -84 -85 -86 -87 -88 -89 90 -91 -92 93 -94 95 -96 -97 98 -99 -100
v -101 -102 -103 104 -105 106 107 -108 -109 -110 -111 -112 -113 -114 115 -116
v -117 -118 119 -120 121 122 -123 124 -125 -126 127 -128 -129 -130 -131 -132
v 133 -134 135 136 -137 -138 -139 -140 -141 -142 -143 -144 -145 -146 -147 148
v 149 150 -151 -152 153 -154 -155 156 -157 -158 -159 -160 -161 162 -163 164 165
v -166 -167 -168 -169 -170 -171 -172 -173 -174 -175 -176 177 178 -179 -180 -181
v 182 -183 -184 185 -186 -187 -188 -189 -190 191 -192 -193 -194 -195 -196 -197
v -198 -199 -200 -201 -202 -203 -204 -205 -206 207 -208 -209 -210 211 -212 -213
v 214 -215 -216 -217 -218 -219 -220 -221 222 223 224 -225 -226 227 -228 -229
v -230 -231 -232 233 -234 -235 -236 -237 -238 -239 -240 -241 -242 -243 244 -245
v -246 -247 -248 -249 -250 251 252 -253 -254 -255 -256 -257 -258 -259 -260 -261
v 262 263 -264 -265 -266 -267 -268 -269 270 271 -272 -273 -274 -275 -276 -277
v -278 -279 280 -281 282 -283 -284 -285 -286 -287 -288 -289 -290 291 -292 293
v -294 295 -296 297 -298 299 -300 301 -302 303 -304 305 -306 -307 308 309 -310
v 311 -312 313 -314 315 -316 317 -318 319 -320 321 -322 323 -324 325 -326 327
v -328 329 -330 -331 332 333 -334 335 -336 337 -338 339 -340 341 -342 343 -344
v 345 -346 -347 348 349 -350 -351 352 353 -354 355 -356 357 -358 359 -360 361
v -362 363 -364 365 -366 367 -368 -369 370 371 -372 373 -374 375 -376 377 -378
v 379 -380 -381 382 383 -384 385 -386 387 -388 -389 390 391 -392 393 -394 395
v -396 397 -398 399 -400 401 -402 403 -404 405 -406 407 -408 409 -410 411 -412
v 413 -414 415 -416 417 -418 419 -420 421 -422 423 -424 425 -426 -427 428 429
v -430 431 -432 433 -434 435 -436 437 -438 439 -440 441 -442 443 -444 445 -446
v -447 448 449 -450 -451 452 -453 454 -455 456 -457 458 -459 460 -461 462 -463
v 464 -465 466 -467 468 469 -470 471 -472 473 -474 475 -476 477 -478 479 -480
v 481 -482 483 -484 485 -486 487 -488 489 -490 491 -492 -493 494 -495 496 -497
v 498 -499 500 -501 502 -503 504 -505 506 507 -508 509 -510 511 -512 513 -514
v 515 -516 517 -518 519 -520 521 -522 523 -524 525 -526 -527 528 529 -530 531
v -532 533 -534 535 -536 537 -538 539 -540 541 -542 543 -544 -545 546 547 -548
v -549 550 551 -552 553 -554 555 -556 557 -558 559 -560 561 -562 563 -564 565
v -566 -567 568 569 -570 571 572 573 574 575 576 577 578 -579 -580 581 582 583
v 584 585 586 587 588 589 -590 591 -592 -593 -594 -595 -596 -597 598 599 600
v 601 -602 603 604 605 -606 -607 -608 609 610 611 612 613 614 -615 -616 617 618
v 619 620 621 -622 623 -624 -625 626 627 628 629 630 -631 -632 -633 -634 -635
v -636 -637 -638 639 640 -641 -642 -643 -644 -645 -646 -647 -648 649 650 651
v -652 -653 -654 -655 -656 -657 -658 659 660 -661 662 663 664 665 666 667 668
v -669 -670 -671 672 673 674 675 676 677 678 -679 -680 -681 -682 -683 -684 -685
v -686 -687 -688 -689 -690 -691 -692 -693 -694 -695 -696 -697 -698 -699 -700
v 701 702 703 704 705 706 707 708 709 710 -711 -712 -713 -714 -715 -716 -717
v -718 719 -720 -721 -722 -723 -724 -725 -726 -727 -728 -729 -730 -731 -732
v -733 -734 -735 -736 -737 -738 -739 -740 741 742 743 744 745 746 747 748 749
v 750 751 752 753 754 755 756 757 758 759 760 -761 -762 -763 -764 -765 -766
v -767 -768 769 770 -771 -772 -773 -774 -775 -776 -777 -778 -779 780 -781 -782
v -783 -784 -785 -786 -787 -788 -789 790 791 792 793 794 795 796 797 798 799
v -800 801 802 803 804 805 806 807 -808 -809 -810 -811 -812 -813 -814 -815 -816
v -817 818 819 820 -821 -822 -823 824 -825 826 -827 828 -829 830 -831 832 -833
v 834 835 836 -837 -838 -839 -840 841 842 843 844 845 846 847 -848 849 850 851
v 852 853 854 855 856 857 858 -859 -860 -861 -862 -863 -864 -865 866 867 868
v -869 -870 -871 -872 -873 -874 -875 876 877 878 -879 -880 -881 -882 -883 -884
v -885 -886 887 888 -889 -890 -891 -892 -893 -894 -895 -896 -897 898 -899 -900
v -901 -902 -903 -904 -905 -906 -907 908 -909 -910 911 912 913 914 915 916 917
v -918 919 920 921 922 923 924 925 926 927 -928 929 -930 931 932 933 934 935
v 936 937 -938 939 -940 941 -942 943 -944 -945 -946 947 948 949 950 -951 952
v -953 954 955 956 -957 -958 -959 -960 -961 -962 -963 -964 965 -966 -967 -968
v -969 -970 971 972 973 974 -975 976 977 978 979 980 -981 982 -983 984 -985 986
v -987 -988 -989 -990 -991 -992 -993 -994 -995 -996 -997 -998 -999 -1000 -1001
v -1002 -1003 -1004 -1005 -1006 -1007 -1008 -1009 -1010 1011 1012 1013 1014
v 1015 1016 1017 1018 1019 1020 -1021 -1022 -1023 -1024 -1025 -1026 -1027 -1028
v 1029 -1030 1031 1032 1033 1034 1035 1036 1037 1038 -1039 1040 -1041 -1042
v -1043 -1044 -1045 -1046 -1047 -1048 1049 -1050 1051 1052 1053 1054 1055 1056
v 1057 1058 -1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 -1069 1070 1071
v -1072 -1073 -1074 -1075 -1076 -1077 -1078 -1079 -1080 -1081 1082 1083 1084
```

```
runtime:  0.8805508613586426 seconds
skyline@skyline-VirtualBox:/media/sf_sharedir/lab1$ python3.7 solveReachability.py data/ex4.v 10 0000000000000000000000010000000

 10  UNROLL RESULTS:
s UNSATISFIABLE
runtime:  0.6546969413757324 seconds
skyline@skyline-VirtualBox:/media/sf_sharedir/lab1$ python3.7 solveReachability.py data/stoplight1.v 10 0100

 10  UNROLL RESULTS:
s UNSATISFIABLE
runtime:  0.061269521713256836 seconds
skyline@skyline-VirtualBox:/media/sf_sharedir/lab1$ python3.7 solveReachability.py data/stoplight2.v 10 01000

 10  UNROLL RESULTS:
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 -67
v -68 -69 -70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 -87 -88 -89 -90
v -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 -142
v 143 144 145 -146 -147 148 149 -150 151 152 -153 -154 -155 -156 -157 158 -159
v -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
v -192 -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 -225 226 227
v 228 229 230 231 232 233 234 -235 -236 237 238 239 240 241 242 243 244 -245
v -246 -247 -248 -249 -250 -251 -252 -253 254 -255 256 -257 -258 259 260 -261
v -262 -263 264 -265 -266 -267 -268 -269 -270 -271 -272 -273 274 -275 -276 -277
v -278 -279 -280 281 282 283 -284 285 286 287 288 289 290 291 292 293 294 -295
v -296 297 298 299 300 -301 -302 -303 -304 -305 306 -307 -308 -309 310 -311
v -312 -313 -314 -315 -316 -317 -318 -319 320 321 322 323 324 325 326 327 328
v 329 -330 331 332 333 -334 335 336 337 338 339 -340 -341 -342 -343 -344 345
v 346 -347 -348 -349 -350 -351 -352 -353 -354 -355 356 -357 -358 -359 -360 361
v 362 363 364 365 -366 367 368 369 370 -371 -372 -373 -374 -375 -376 377 378
v 379 380 381 -382 -383 -384 -385 -386 387 -388 -389 -390 -391 -392 -393 -394
v -395 -396 397 -398 -399 -400 401 402 403 404 405 406 -407 408 409 410 -411
v -412 -413 -414 -415 -416 417 418 419 420 421 422 423 424 425 426 -427 -428
v -429 -430 431 432 433 434 435 436 -437 -438 -439 -440 441 442 443 444 445
v -446 -447 -448 -449 -450 451 452 453 -454 455 -456 -457 -458 -459 -460 461
v 462 463 -464 -465 -466 -467 -468 -469 -470 -471 -472 -473 474 475 476 477 478
v 479 480 -481 -482 -483 -484 -485 -486 -487 -488 -489 -490 491 492 -493 -494
v -495 -496 497 498 -499 -500 -501 -502 -503 -504 -505 -506 -507 -508 -509 -510
v -511 512 -513 -514 -515 -516 -517 518 -519 -520 -521 522 -523 -524 -525 -526
v -527 -528 -529 -530 531 -532 533 534 535 536 537 538 539 540 541 -542 543
v -544 -545 -546 -547 -548 -549 -550 -551 552 -553 554 555 556 -557 558 -559 560
v -561 -562 -563 564 565 566 567 568 569 570 571 572 573 -574 -575 -576 -577
v -578 -579 -580 -581 -582 -583 -584 -585 -586 587 -588 589 -590 -591 -592 -593
v -594 -595 -596 597 -598 599 -600 601 602 603 604 605 606 -607 608 -609 610
v -611 -612 -613 -614 -615 -616 -617 618 -619 620 -621 -622 -623 -624 -625 -626
v -627 628 -629 630 631 632 633 634 635 636 637 -638 639 -640 641 642 643 644
v 645 646 -647 -648 -649 -650 651 652 653 654 655 -656 -657 -658 -659 -660 661
v 662 663 664 665 -666 -667 -668 -669 -670 -671 -672 -673 -674 -675 676 677 678
v 679 680 -681 -682 -683 -684 685 686 -687 -688 -689 -690 -691 -692 -693 -694
v -695 -696 -697 -698 -699 -700 701 702 703 704 705 706 707 708 709 710 -711
v -712 -713 -714 -715 -716 -717 -718 -719 -720 721 722 723 724 725 726 727 728
v 729 730 -731 -732 -733 -734 -735 -736 -737 -738 -739 -740 -741 -742 -743 -744
v -745 -746 -747 -748 -749 -750 751 752 753 754 755 756 757 758 759 760 761 762
v 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 -781
v -782 -783 -784 -785 -786 -787 -788 -789 -790 -791 -792 -793 -794 -795 -796
v -797 -798 -799 -800 -801 802 -803 -804 -805 806 -807 808 -809 810 -811 -812
v -813 -814 -815 -816 -817 -818 -819 -820 821 822 823 824 825 826 827 828 829
v 830 -831 -832 -833 -834 835 -836 -837 -838 -839 -840 -841 -842 -843 -844 -845
v -846 -847 -848 -849 -850 851 852 853 854 855 856 857 858 859 860 861 862 863
v 864 865 866 867 868 869 870 871 872 873 874 -875 876 877 878 879 880 881 882
v 883 -884 -885 886 887 888 889 -890 -891 -892 -893 894 895 -896 -897 -898 -899
v 900 -901 -902 -903 -904 -905 -906 -907 908 -909 -910 911 912 913 914 915 916
v 917 -918 919 920 -921 -922 -923 -924 925 926 -927 -928 -929 -930 -931 -932
v -933 -934 -935 -936 -937 -938 -939 -940 941 942 943 944 945 946 947 948 949
v 950 951 952 953 954 955 956 957 -958 959 960 -961 -962 963 -964 965 -966 -967
v -968 969 -970 -971 -972 -973 -974 -975 -976 -977 -978 979 -980 981 982 983
v 984 985 986 987 988 -989 990 -991 -992 993 994 995 996 -997 -998 -999 -1000
v -1001 -1002 1003 -1004 1005 -1006 -1007 -1008 -1009 -1010 1011 1012 -1013
v 1014 -1015 1016 1017 1018 1019 1020 1021 1022 -1023 1024 -1025 1026 1027 1028
v -1029 1030 1031 1032 -1033 1034 -1035 1036 1037 -1038 -1039 1040 1041 -1042
v 1043 -1044 1045 1046 1047 1048 1049 1050 1051 -1052 -1053 -1054 -1055 1056
v 1057 -1058 -1059 1060 -1061 1062 1063 1064 1065 -1066 -1067 1068 1069 -1070
v 1071 1072 1073 1074 1075 1076 -1077 1078 -1079 1080 1081 1082 -1083 1084
v -1085 1086 1087 1088 1089 1090 1091 1092 -1093 1094 -1095 1096 -1097 1098
v -1099 1100 -1101 1102 -1103 1104 1105 1106 1107 1108 1109 1110 -1111 1112
v -1113 1114 -1115 1116 -1117 1118 -1119 1120 1121 -1122 1123 -1124 1125 -1126
v 1127 -1128 1129 -1130 0
runtime:  0.11345291137695312 seconds
skyline@skyline-VirtualBox:/media/sf_sharedir/lab1$
```

For this question. I just unroll 10 times for the transition relation.

For each time, I assign the value by different loop.

The data structure I use is dictionary. list. and class.

Dictionary map the string to digit.

class contain many gate. reggister. write function which show the CNF formula.

(3). There should be some difference between SAT-based reachability and random simulation. simply because symbolic Reachability use exhaust test while random simulation use random input. When too many inputs required, maybe you can get a 'timed out' although some satisfied state exist.

In order to recover the limitation in SAT-based rechability due to only checking the last unrolling for final state. We can traversal every unrolling for the final state.

```
skyline@skyline-VirtualBox:/media/sf_sharedir/lab1$ python3.7 solveReachability.py data/stoplight1.v -c 32 0100

 1  UNROLL RESULTS:
s UNSATISFIABLE
runtime:  0.021446943283081055 seconds

 2  UNROLL RESULTS:
s UNSATISFIABLE
runtime:  0.035504102706909 18 seconds

 3  UNROLL RESULTS:
s UNSATISFIABLE
runtime:  0.05167007446289 0625 seconds

 4  UNROLL RESULTS:
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
runtime:  0.070375204086303 71 seconds

 5  UNROLL RESULTS:
s UNSATISFIABLE
runtime:  0.090251684188842 77 seconds

 6  UNROLL RESULTS:
s UNSATISFIABLE
runtime:  0.120880842208862 3 seconds

 7  UNROLL RESULTS:
s UNSATISFIABLE
runtime:  0.146992683410644 53 seconds

 8  UNROLL RESULTS:
s UNSATISFIABLE
runtime:  0.176416873931884 77 seconds

 9  UNROLL RESULTS:
s UNSATISFIABLE
runtime:  0.207785367965698 24 seconds

 10  UNROLL RESULTS:
s UNSATISFIABLE
runtime:  0.262479066848754 9 seconds

 11  UNROLL RESULTS:
s UNSATISFIABLE
runtime:  0.304505825804272 46 seconds

 12  UNROLL RESULTS:
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 24
v 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 -67
v -68 -69 -70 -71 -72 -73 74 -75 -76 -77 78 79 80 -81 82 -83 -84 -85 86 -87 -88
v -89 90 -91 -92 -93 94 -95 -96 -97 -98 -99 -100 -101 -102 -103 -104 -105 -106
v -107 -108 109 110 111 112 113 114 115 116 117 118 119 120 -121 -122 -123 124
v -125 126 -127 -128 -129 -130 -131 132 -133 -134 -135 -136 -137 138 -139 -140
v -141 -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 -176
v 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 -209 -210
v 211 212 213 214 215 216 217 218 219 220 -221 -222 -223 -224 225 226 227 228
v -229 -230 -231 232 -233 -234 -235 -236 -237 -238 -239 240 241 242 243 -244
v 245 246 247 248 249 250 251 -252 253 254 -255 -256 -257 -258 259 260 261 262
v -263 -264 -265 266 -267 -268 -269 270 -271 272 -273 274 -275 276 -277 278
v -279 -280 -281 -282 -283 284 -285 286 -287 -288 -289 -290 -291 -292 -293 -294
v -295 -296 -297 -298 -299 -300 301 302 303 304 305 306 307 308 309 310 311 312
v 313 314 315 -316 317 318 319 320 321 322 323 -324 325 326 327 -328 -329 330
v 331 332 333 334 335 -336 337 338 339 -340 -341 -342 343 344 345 346 347 -348
v -349 -350 -351 352 353 354 -355 -356 -357 -358 -359 360 -361 -362 -363 -364
v -365 -366 367 368 -369 -370 -371 -372 -373 -374 -375 -376 -377 -378 379 380
v -381 -382 -383 -384 -385 386 -387 -388 -389 -390 -391 392 -393 394 -395 -396
v -397 398 -399 -400 -401 -402 -403 -404 -405 406 -407 -408 409 -410 411 412
v 413 414 415 416 417 -418 419 420 421 -422 423 -424 -425 -426 -427 -428 429
v -430 431 -432 -433 434 -435 436 437 438 439 440 -441 442 -443 444 -445 -446
v -447 448 449 450 451 452 -453 -454 -455 456 457 458 459 -460 -461 -462 -463
```

the value i should be 4 or 12 ~ 32

In A.2 the shortest path for stoplight1 is 4, which is the same with the value i. but SAT solver can detect more reachable clock by unrolling.

(6).

```
skyline@skyline-VirtualBox:/media/sf_sharedir/lab1$ python3.7 solveReachability.py data/stoplight2.v -c 32 01000

 1  UNROLL RESULTS:
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 24 25
v -26 -27 -28 29 30 -31 -32 33 34 -35 -36 37 -38 39 -40 41 -42 43 44 45 46 47
v -48 -49 50 -51 -52 -53 54 55 -56 -57 58 -59 -60 61 -62 -63 64 65 66 67 -68
v -69 -70 71 -72 73 -74 -75 76 77 78 -79 -80 -81 -82 83 -84 -85 86 87 88 89 -90
v -91 92 -93 -94 95 96 -97 -98 99 -100 -101 102 103 104 105 106 -107 108 109
v 110 -111 -112 113 0
runtime:  0.013433218002319336 seconds

 2  UNROLL RESULTS:
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 24
v 25 26 27 -28 29 -30 31 32 -33 -34 35 36 -37 -38 -39 -40 41 42 43 44 45 46 47
v 48 49 50 -51 -52 -53 -54 -55 -56 57 58 59 60 -61 -62 -63 -64 65 66 67 68 -69
v -70 -71 -72 73 74 -75 -76 77 -78 -79 -80 81 82 -83 -84 85 86 87 88 89 90 91
v 92 93 94 -95 -96 -97 -98 -99 100 -101 -102 -103 104 -105 -106 -107 -108 109
v -110 111 112 -113 114 115 116 -117 -118 -119 -120 121 122 -123 -124 -125 -126
v 127 128 129 130 131 132 133 134 -135 -136 -137 -138 -139 140 141 142 -143
v -144 145 146 -147 -148 -149 -150 151 152 153 154 155 156 -157 -158 -159 -160
v -161 162 -163 -164 165 166 -167 -168 -169 -170 171 172 173 174 175 176 177
v 178 -179 -180 -181 -182 183 184 -185 -186 -187 -188 189 190 191 192 -193 -194
v -195 -196 197 198 -199 -200 -201 -202 203 204 205 206 207 208 209 -210 211
v -212 -213 214 215 216 217 218 219 220 -221 222 -223 224 225 -226 0
runtime:  0.033943414688110 35 seconds

 3  UNROLL RESULTS:
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 24
v 25 26 27 -28 -29 -30 -31 -32 -33 34 35 36 37 38 -39 40 -41 42 -43 44 -45 46
v 47 -48 -49 -50 -51 52 53 54 -55 -56 -57 -58 -59 -60 61 62 63 64 65 66 67 68
v 69 70 71 72 73 74 75 -76 -77 -78 -79 -80 -81 -82 -83 -84 85 86 87 88 89 90
v -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 -142 -143
v -144 -145 -146 -147 148 149 -150 -151 -152 -153 -154 155 -156 -157 158 -159
v 160 -161 162 163 -164 165 -166 167 -168 -169 -170 -171 172 173 174 -175 -176
v -177 -178 -179 -180 181 182 183 -184 -185 -186 -187 -188 -189 190 191 192 193
v 194 195 196 197 198 199 200 201 -202 -203 -204 -205 -206 -207 -208 -209 -210
v 211 212 213 -214 -215 -216 217 218 219 -220 -221 -222 -223 -224 -225 226 227
v 228 229 230 231 232 233 234 -235 -236 -237 -238 -239 -240 -241 242 -243 -244
v -245 -246 247 248 249 -250 -251 -252 -253 -254 -255 256 257 258 259 260 261
v 262 263 264 265 266 267 -268 -269 -270 -271 -272 -273 274 275 276 -277 -278
v -279 -280 -281 -282 283 284 285 286 287 288 -289 -290 291 -292 -293 -294 295
v 296 297 -298 -299 300 -301 -302 303 304 305 -306 307 308 -309 310 311 -312
v 313 -314 315 316 -317 -318 -319 320 321 322 323 324 325 326 -327 328 329 -330
v -331 332 -333 -334 335 -336 337 -338 339 0
runtime:  0.0612637996673584 seconds

 4  UNROLL RESULTS:
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 45 46
v 47 48 49 50 -51 -52 53 -54 55 -56 57 -58 59 -60 61 62 -63 -64 -65 -66 -67 -68
v 69 70 71 72 -73 -74 -75 -76 -77 -78 -79 -80 81 82 83 84 85 86 87 88 89 90 91
v 92 93 94 95 96 97 98 99 100 -101 -102 -103 104 -105 -106 -107 -108 -109 -110
v -111 -112 113 114 115 116 117 118 119 120 -121 -122 -123 124 -125 -126 -127
v 128 129 130 131 -132 133 134 135 -136 -137 -138 -139 -140 -141 -142 -143 -144
v 145 146 147 148 -149 -150 -151 -152 153 -154 -155 -156 -157 -158 -159 -160
v 161 162 163 164 -165 -166 -167 -168 169 170 171 172 173 174 175 176 177 178
v 179 180 181 182 183 -184 185 186 187 -188 -189 -190 -191 192 -193 -194 -195
v -196 197 198 -199 -200 -201 -202 -203 -204 -205 206 -207 -208 -209 210 -211
v -212 213 -214 215 216 217 -218 219 -220 -221 222 -223 224 -225 -226 -227 228
v 229 230 231 -232 -233 -234 -235 -236 -237 -238 -239 -240 241 242 243 244 -245
v -246 -247 -248 -249 -250 -251 -252 253 254 255 256 257 258 259 260 261 262
v 263 264 265 266 267 268 -269 -270 -271 -272 -273 -274 -275 -276 -277 -278
v -279 -280 281 282 283 284 -285 -286 -287 -288 289 290 291 292 -293 -294 -295
v -296 -297 -298 -299 -300 301 302 303 304 305 306 307 308 309 310 311 312 -313
v -314 -315 -316 -317 -318 -319 -320 -321 322 -323 324 -325 -326 -327 -328 329
v 330 331 332 -333 -334 -335 -336 -337 -338 -339 -340 341 342 343 344 345 346
v 347 348 349 350 351 352 353 354 355 -356 -357 -358 -359 360 -361 -362 -363
v -364 365 366 367 368 -369 -370 -371 -372 -373 -374 -375 -376 377 378 379 380
v 381 382 383 384 -385 -386 387 -388 -389 -390 -391 -392 393 394 395 396 -397
v -398 399 400 -401 -402 403 -404 405 406 -407 408 409 410 -411 412 413 414
v -415 416 417 -418 419 420 421 -422 -423 424 -425 426 427 -428 429 430 431 432
v 433 434 -435 436 437 438 -439 440 -441 442 -443 444 -445 446 -447 448 449
v -450 451 -452 0
runtime:  0.09693574905395508 seconds

 5  UNROLL RESULTS:
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 45 -46 -47
v -48 -49 50 -51 -52 -53 -54 55 56 57 58 59 -60 61 62 -63 -64 -65 66 -67 68 -69
v 70 -71 -72 73 74 -75 76 77 -78 -79 -80 -81 -82 -83 -84 -85 86 87 88 89 90 -91
v -92 -93 94 -95 -96 -97 -98 -99 -100 101 102 103 104 105 106 107 108 109 110
v 111 112 113 114 -115 116 117 118 119 -120 121 122 123 124 -125 -126 -127 -128
v 129 -130 -131 -132 -133 134 -135 -136 -137 -138 139 -140 141 142 143 -144 145
```

the satisfied value is 1~32, which show with the increase of input combinations. the exhaust test is easier to find reachability than random input.

So overall. SAT-based Reachability is better than testbench random execution, especially in large input combinations.

In A.3. the answer is 1358 clock cycle, which is not a good answer simply because of large input combinations.

(7) the system has $2^5 = 32$ states for stoplight2.v. I just type the command line argument 32 times and change the final state parameter one by one. The answer is all states(32) can be reached as exactly the 17th state from the initial state.