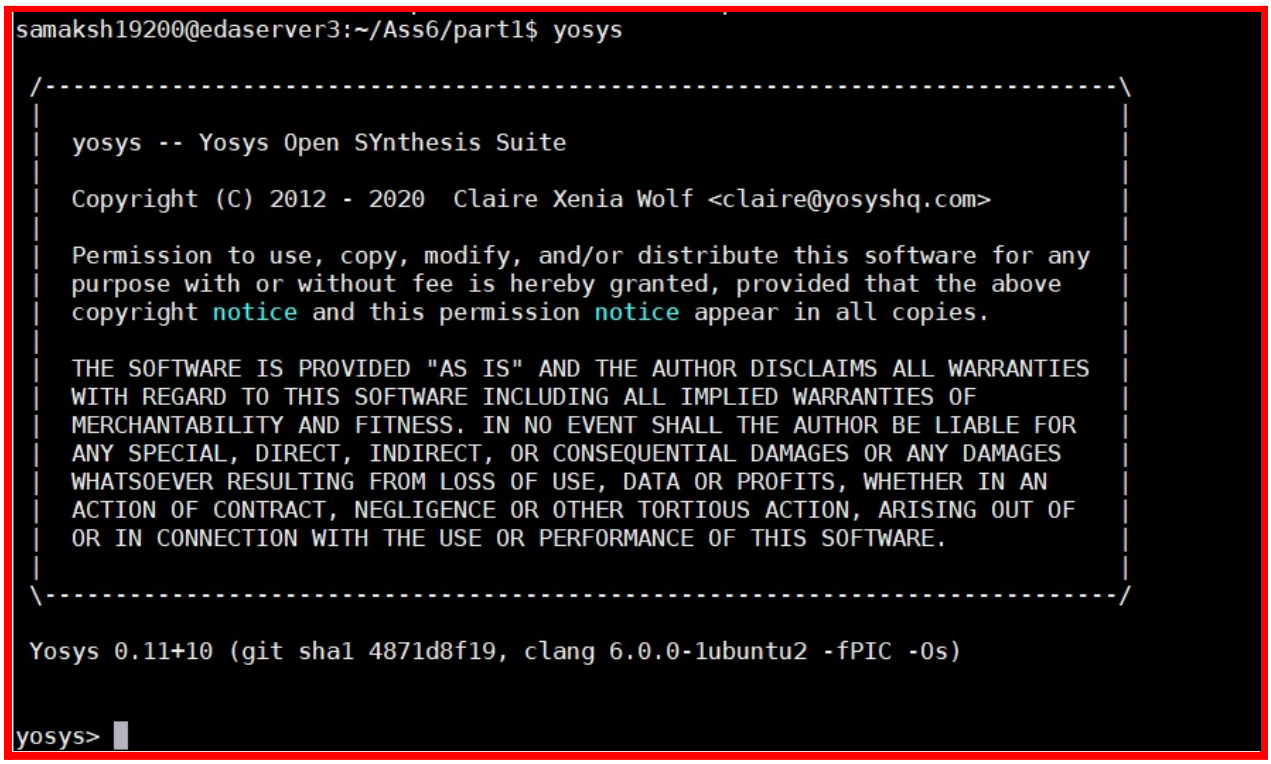
ASSIGNMENT 6

SAMAKSH GUPTA
2019200

PART 2

1. Running Yosys



2. Netlists for the two designs

Design 1

```
//`include "fast.lib"

module des1(input in, input clk, output out);


//CLKINVX1
//AND2X4
//XNOR2X1
//SDFFX1
//BUFX2

wire in11; wire in12; wire q1; wire q2; wire d; wire q3;

CLKINVX1 inv1(in, in11);
BUFX2 buf1(in, in12);

SDFFX1 ff1(.CK(clk), .D(in11), .Q(q1));
SDFFX1 ff2(.CK(clk), .D(in12), .Q(q2));

XNOR2X1 xnor1(q1,q2,d);
SDFFX1 ff3(.CK(clk), .D(d), .Q(q3));

CLKINVX1 inv2(q3, out);


endmodule
```

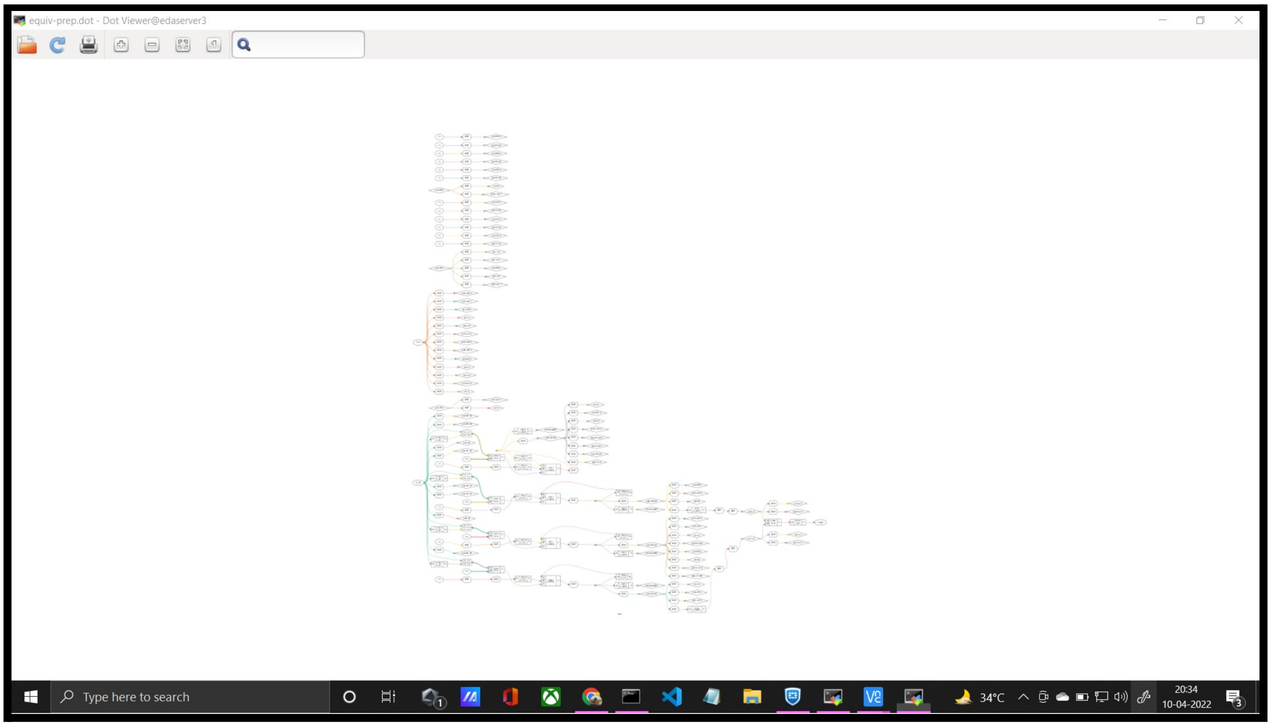
Design 2

```
//`include "fast.lib"

module des2(input in, input clk, output out);


//CLKINVX1
//AND2X4
//XNOR2X1
//SDFFX1
//BUFX2

wire in11; wire in12; wire q1; wire q2; wire d; wire q3;

CLKINVX1 inv1(in, in11);
BUFX2 buf2(in, in12);

SDFFX1 ff1(.CK(clk), .D(in11), .Q(q1));
SDFFX1 ff2(.CK(clk), .D(in12), .Q(q2));

AND2X4 and1(q1,q2,d);
SDFFX1 ff3(.CK(clk), .D(d), .Q(q3));

CLKINVX1 inv2(q3, out);


endmodule
~
```

CEC Script

```
#Source: Google Classroom Assignment-6 (Given)

#gold design
read_verilog des1.v
read_liberty -ignore_miss_func -ignore_miss_data_latch fast.lib
#If we don't use the above method we get miter design instantiation error.
design -save lib
prep -flatten -top des1
splitnets -ports;;
design -stash gold

################################################################################

#gate design
read_verilog des2.v
read_liberty -ignore_miss_func -ignore_miss_data_latch fast.lib
#If we don't use the above method we get miter design instantiation error.
prep -flatten -top des2
splitnets -ports;;
design -stash gate

################################################################################

design -copy-from gold -as gold des1
design -copy-from gate -as gate des2

#prove combinational equivalence checking
miter -equiv -flatten gold gate miter
prep -flatten -top miter

#design -save lib
dfflibmap -liberty fast.lib
abc -liberty fast.lib
techmap -map %lib
clk2fflogic

show -prefix equiv-prep -colors 1 -stretch
sat -all -verify -tempinduct -prove trigger 0 -set-at 1 in_in 0

################################################################################
```

RESULT OF RUNNING THE SEC SCRIPT

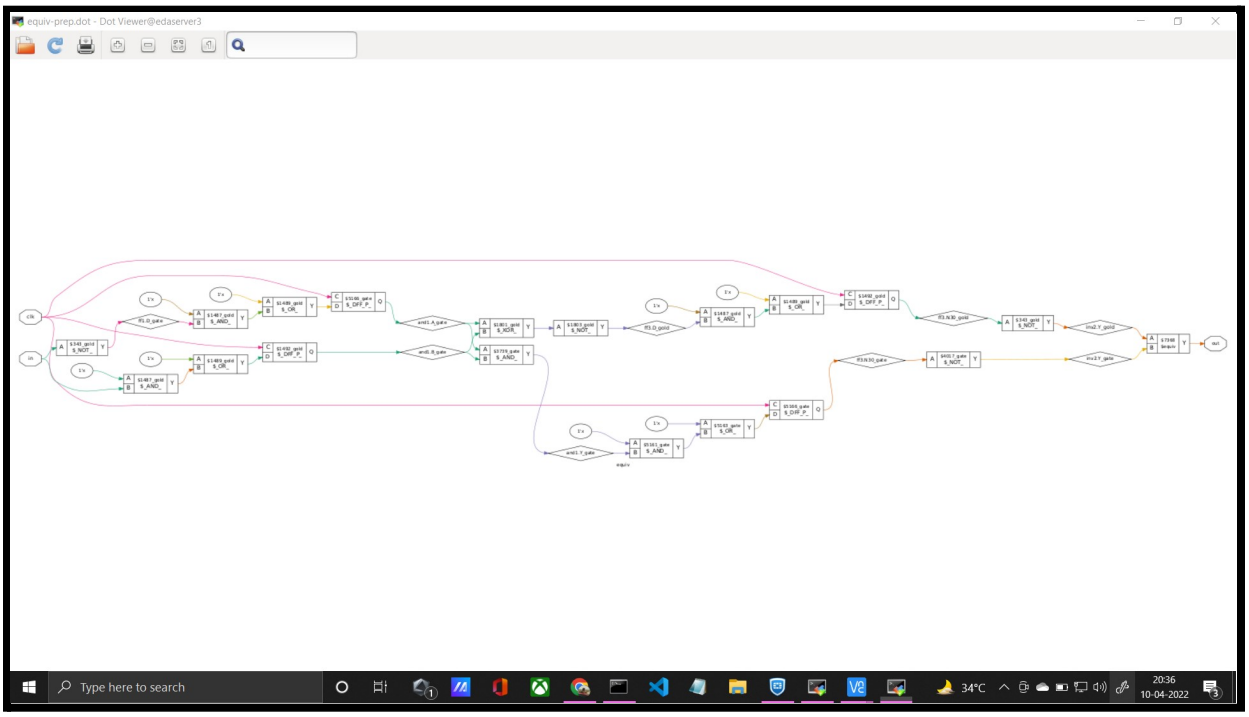



```
Final constraint equation: \in_in = 1'0
Imported 28 cells to SAT database.
Import set-constraint from init attribute: $auto$clk2fflogic.cc:185:execute$7391 = 1'1
Import set-constraint from init attribute: $auto$clk2fflogic.cc:185:execute$7381 = 1'1
Import set-constraint from init attribute: $auto$clk2fflogic.cc:185:execute$7371 = 1'1
Import set-constraint from init attribute: $auto$clk2fflogic.cc:185:execute$7361 = 1'1
Final init constraint equation: { $auto$clk2fflogic.cc:185:execute$7361 $auto$clk2fflogic.cc:185:execute$7371 $auto$clk2fflogic.cc:185:execute$7381 $auto$clk2fflogic.cc:1
85:execute$7391 } = 4'1111
Import proof-constraint: \trigger = 1'0
Final proof equation: \trigger = 1'0

[base case 1] Solving problem with 119 variables and 284 clauses..
SAT temporal induction proof finished - model found for base case: FAIL!

   ______                   ___       ___       _ _            _ _
  (_____ \                 / __)     / __)     (_) |          | | |
   _____) )___ ___   ___ _| |__    _| |__ _____ _| | _____  __| | |
  |  ____/ ___) _ \ / _ (_   __)  (_   __|____ | | || ___ |/ _  |_|
  | |   | |  | |_| | |_| || |       | |  / ___ | | || ____( (_| |_
  |_|   |_|   \___/ \___/ |_|       |_|  \_____|_|\_)_____)\____|_|


  Time Signal Name                                 Dec        Hex             Bin
  ---- -------------------------------------- ---------- --------- ------------
  init $auto$clk2fflogic.cc:173:execute$7359           0          0              0
  init $auto$clk2fflogic.cc:173:execute$7369           1          1              1
  init $auto$clk2fflogic.cc:173:execute$7379           1          1              1
  init $auto$clk2fflogic.cc:173:execute$7389           0          0              0
  init $auto$clk2fflogic.cc:185:execute$7361           1          1              1
  init $auto$clk2fflogic.cc:185:execute$7371           1          1              1
  init $auto$clk2fflogic.cc:185:execute$7381           1          1              1
  init $auto$clk2fflogic.cc:185:execute$7391           1          1              1
  init $auto$clk2fflogic.cc:205:execute$7365           0          0              0
  init $auto$clk2fflogic.cc:205:execute$7375           0          0              0
  init $auto$clk2fflogic.cc:205:execute$7385           0          0              0
  init $auto$clk2fflogic.cc:205:execute$7395           0          0              0
  ---- -------------------------------------- ---------- --------- ------------
     1 \in_clk                                         0          0              0
     1 \in_in                                          0          0              0
     1 \trigger                                        1          1              1

ERROR: Called with -verify and proof did fail!
```

SEC Script

```
##Source: Google Classroom Assignment-6 (Given)


#gold design
read_verilog des1.v
read_liberty -lib fast.lib
read_liberty -ignore_miss_func -ignore_miss_data_latch fast.lib
prep -flatten -top des1
splitnets -ports;;
design -stash gold

################################################################################

#gate design
read_verilog des2.v
read_liberty -lib fast.lib
read_liberty -ignore_miss_func -ignore_miss_data_latch fast.lib
prep -flatten -top des2
splitnets -ports;;
design -stash gate

################################################################################

design -copy-from gold -as gold des1
design -copy-from gate -as gate des2

################################################################################

#prove sequential equivalence checking
equiv_make gold gate equiv
prep -flatten -top equiv
opt_clean -purge

show -prefix equiv-prep -colors 1 -stretch
equiv_induct -seq 5
equiv_status -assert

################################################################################
~
```

RESULT OF RUNNING THE SEC SCRIPT

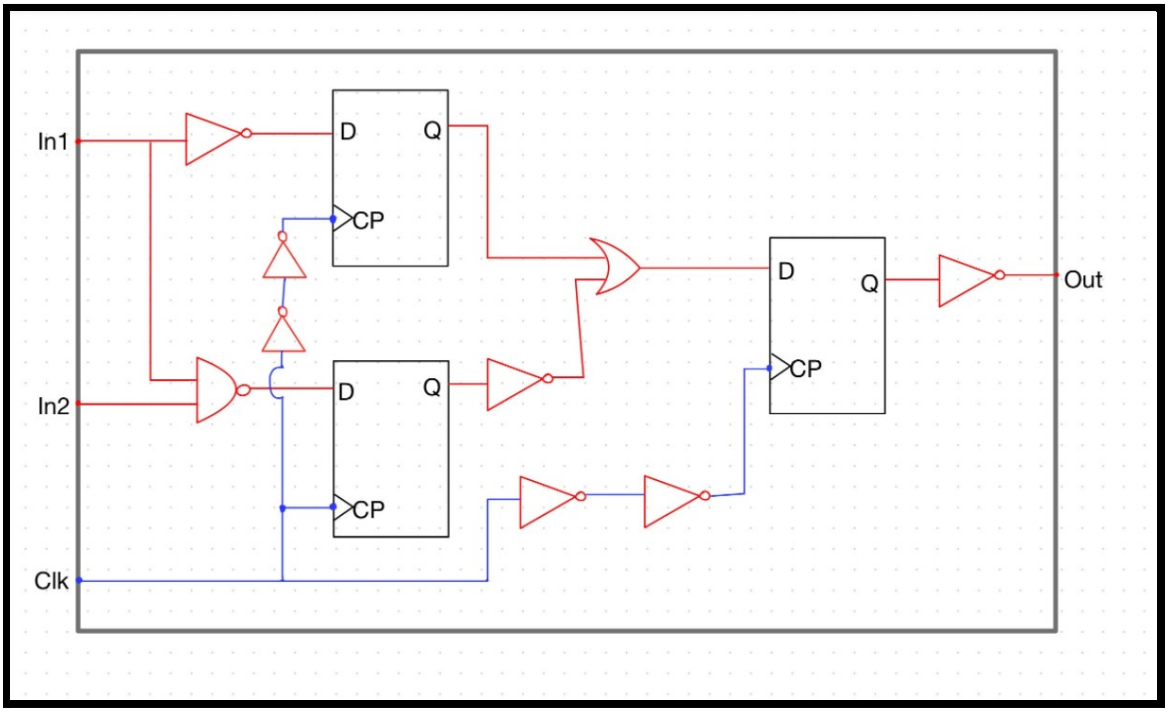



```
   Number of wires:                 90
   Number of wire bits:             90
   Number of public wires:          81
   Number of public wire bits:      81
   Number of memories:               0
   Number of memory bits:            0
   Number of processes:              0
   Number of cells:                 19
     $_AND_                          5
     $_DFF_P_                        4
     $_NOT_                          4
     $_OR_                           4
     $_XOR_                          1
     $equiv                          1

12.13. Executing CHECK pass (checking for obvious problems).
Checking module equiv...
Found and reported 0 problems.

13. Executing OPT_CLEAN pass (remove unused cells and wires).
Finding unused cells or wires in module \equiv..
Removed 0 unused cells and 69 unused wires.
<suppressed ~69 debug messages>

14. Generating Graphviz representation of design.
Writing dot description to `equiv-prep.dot'.
Dumping module equiv to page 1.
Exec: { test -f 'equiv-prep.dot.pid' && fuser -s 'equiv-prep.dot.pid' 2> /dev/null; } || ( echo $$ >&3; exec xdot 'equiv-prep.dot'; ) 3> 'equiv-prep.dot.pid' &

15. Executing EQUIV_INDUCT pass.
Found 1 unproven $equiv cells in module equiv:
  Proving existence of base case for step 1. (186 clauses over 78 variables)
  Proving induction step 1. (409 clauses over 166 variables)
  Proof for induction step holds. Entire workset of 1 cells proven!
Proved 1 previously unproven $equiv cells.

16. Executing EQUIV_STATUS pass.
Found 1 $equiv cells in equiv:
  Of those cells 1 are proven and 0 are unproven.
  Equivalence successfully proven!

yosys>
```

Equivalence Successfully Proved

PART 1

Design



Netlist

top.v

```
module top(input clk, input in1, input in2, output out);

wire clk1; wire clk2; wire clk3; wire clk4;
wire d1; wire d2; wire d3;
wire q1; wire q2; wire q3; wire q4;



CLKINVX1 Inv1(clk, clk1);
CLKINVX1 Inv2(clk1, clk2);
CLKINVX1 Inv3(clk, clk3);
CLKINVX1 Inv4(clk3, clk4);

CLKINVX1 Inv5(in1, d1);
NAND2X1 Nand1(in1, in2, d2);

SDFFX1 Ff1(.CK(clk2), .D(d1), .Q(q1));
SDFFX1 Ff2(.CK(clk), .D(d2), .Q(q2));

CLKINVX1 Inv6(q2, q3);
NOR2X1 Nor1(q1,q3,d3);

SDFFX1 Ff3(.CK(clk4), .D(d3), .Q(q4));
CLKINVX1 Inv7(q4, out);

endmodule
```

```
    RC-Corner PostRoute Cap Factor        : 1
    RC-Corner PostRoute XCap Factor       : 1
Current (total cpu=0:00:08.8, real=0:00:09.0, peak res=856.6M, current mem=816.7M)
INFO (CTE): Constraints read successfully.
Ending "Constraint file reading stats" (total cpu=0:00:00.0, real=0:00:00.0, peak res=835.9M, cur
rent mem=835.9M)
Current (total cpu=0:00:08.9, real=0:00:09.0, peak res=856.6M, current mem=835.9M)
**WARN: (IMPESI-3468):  User needs to specify -equivalent_waveform_model propagation first before
 specifying -waveform_compression_mode accurate|clock_detailed .
AAE DB initialization (MEM=1094.76 CPU=0:00:00.0 REAL=0:00:00.0)
################################################################################
# Design Name: rtl_module
# Design Mode: 65nm
# Analysis Mode: MMMC OCV
# Parasitics Mode: No SPEF/RCDB
# Signoff Settings: SI Off
################################################################################
Start delay calculation (fullDC) (1 T). (MEM=950.77)
End delay calculation. (MEM=987.062 CPU=0:00:00.0 REAL=0:00:00.0)
End delay calculation (fullDC). (MEM=979.062 CPU=0:00:00.1 REAL=0:00:00.0)
INFO: Path Based Analysis (PBA) performed on total '30' paths
INFO: Path Based Analysis (PBA) performed on total '30' paths
INFO: Path Based Analysis (PBA) performed on total '30' paths
tempus 1>
```

Unable to get the license … :(